Journal of Reliable Intelligent Environments (2023) 9:263–281
https://doi.org/10.1007/s40860-023-00209-5

ORIGINAL ARTICLE

An extension of the ADVISE Meta modeling framework and its application for an early-stage security analysis of a public transport supervision system

Francesco Mariotti[1] · Andrea Bondavalli[1] · Paolo Lollini[1] · Leonardo Montecchi[2] · Simone Nardi[3]

Received: 10 January 2023 / Accepted: 9 June 2023 / Published online: 23 June 2023

Abstract
Early-stage security analysis can be used for a preliminary assessment of the security level of a system, thus providing useful insights to guide the whole system's development. In this paper, we focus on a specific meta-level modeling framework for security analysis, ADVISE Meta, which allows representing a system using generic built-in blocks and relationships constituting the ontology of the framework, and to automatically derive complex low-level stochastic models representing attack steps and adversaries. In this paper, we extend the ADVISE Meta ontology to enlarge the variety of the possible attack paths and adversaries that can be represented in the framework, by modeling (i) attack patterns available in the CAPEC database, a comprehensive dictionary of known patterns of attack, and (ii) the adversaries' profiles defined in the Threat Agent Library (TAL), a reference library which describes the characteristics of threat agents. The paper provides a detailed description of the whole process for extending the ADVISE Meta ontology, and the application of the extended modeling framework for an early-stage security analysis of a public transport supervision system. The framework enables a variety of security-oriented analyses, in particular to assess the probability that a given adversary can successfully reach a specific goal, to analyze the most probable attack path that adversaries can follow to reach a goal, to perform sensitivity analysis at varying of attack patterns and adversaries' profiles, to compare different architectural solutions, and to identify the system's components that can be more probably attacked by adversaries.

Keywords Adversary profile · Attack pattern · CAPEC · Cybersecurity assessment · Modeling · TAL

✉ Francesco Mariotti
francesco.mariotti@unifi.it

Andrea Bondavalli
andrea.bondavalli@unifi.it

Paolo Lollini
paolo.lollini@unifi.it

Leonardo Montecchi
leonardo.montecchi@ntnu.no

Simone Nardi
simone.nardi@mermec-engineering.com

[1] Dipartimento di Matematica e Informatica 'U. Dini', University of Firenze, Viale Morgagni 65, 50134 Firenze, Italy

[2] Department of Computer Science, Norwegian University of Science and Technology, Sem Sælands vei 7–9, 7034 Trondheim, Norway

[3] Computational Science Department, MERMEC Engineering Srl, Via Livornese 1019, San Piero a Grado, 56122 Pisa, Italy

1 Introduction

Models for early-stage security analysis can be used for a preliminary assessment of the most critical architectural components of a system, allowing to identify those that should be more protected. Such analysis is performed having a very preliminary knowledge of the system, without knowing which are the vulnerabilities of the components, which are the possible involved attacks, which are the adversaries' profiles that could potentially perform such attacks, and the consequences of such attacks.

In the literature, several formalisms have been proposed for helping with such challenging activity [3, 13]. However, for complex systems, the application of such formalisms is often a time-consuming and error-prone activity, and information from experts in the application domain is required for properly capturing key elements (e.g., attack patterns or adversaries), which are rarely formalized. Model-driven engineering frameworks [24] can thus play a fundamental

role and, starting from an high-level architectural description of the system, they can be used to derive complex low-level analyzable models.

In our previous work [15], we proposed a preliminary version of a methodology for the extension of the security-oriented modeling framework called ADVISE Meta [10]. The methodology aims at integrating into the ontology: (i) attack patterns from the CAPEC (Common Attack Pattern Enumerations and Classifications) database [18], which is a publicly available catalog of common attack patterns; and (ii) adversaries' profiles from the Threat Agent Library (TAL) [4] by Intel, that provides a description of the human agents that can threaten IT systems and other information assets.

In this work, we refine the methodology, we discuss its application for extending the ADVISE Meta ontology with all the TAL adversaries and some representative CAPEC attacks, and we make use of the extended framework for an early-stage security analysis of a public transport supervision system called Smart Passenger Center (SPaCe) [14]. ADVISE Meta allows computing the probability that a given adversary can successfully reach a specific goal, and analyzing the most probable attack path that the adversary will follow to reach the goal. With the proposed extension, we enable additional analysis scenarios, ranging from broad security analysis at varying of TAL adversaries, where CAPEC attacks are involved, to the identification of the components of the system that are most exposed to threats. Such kind of analysis can be used to guide the system development process.

The rest of this paper is organized as follows. In Sect. 2, we provide some background information about CAPEC, TAL, ADVISE, and ADVISE Meta, for a better understanding of the rest of the paper. The methodology to map CAPEC and TAL elements into the ontology of ADVISE Meta is presented in Sect. 3, and it is applied in Sect. 4 for the extension of the ADVISE Meta ontology with TAL adversaries' profiles and CAPEC attack patterns. In Sect. 5, we introduce the SPaCe system and its high-level architecture, which is then modeled in Sect. 6, and analyzed in Sect. 7 considering different analysis scenarios. Related works are discussed in Sect. 8, while conclusions are finally drawn in Sect. 9.

2 Background

In the following, we give a brief description of the core elements used in the rest of paper, namely CAPEC, TAL, ADVISE, and ADVISE Meta.

2.1 Common attack pattern enumeration and classification

Throughout the history of IT security, it has become increasingly necessary to have reference lists of possible threats to IT systems, like the "OWASP Top 10" for web applications [20], or the MITRE Common Weakness Enumeration (CWE) [19].

The MITRE Common Attack Pattern Enumeration and Classification (CAPEC) [18] is a large online catalog containing more than 500 entries of common *attack patterns*. An attack pattern is a description of the common attributes and approaches used by adversaries to exploit known weaknesses in IT systems.

Each entry in the database describes a particular attack pattern and contains, among others, the following sections: a general *Description* of the attack; the *Prerequisites* that are needed in order to carry out the attack; *Resources Required*, providing information on devices, tools, and other resources needed to perform the attack; *Skills Required*, which indicates the skills that an adversary must possess to carry out the attack; and possible *Consequences* of the attack (its scope and its possible impact).

2.2 Threat agent library

When considering the possible threats that can threaten a system, it is also fundamental to identify which are the adversaries' profiles that can attempt the attack. Naming an adversary with generic terms like "hacker" or "spy" can be misleading. In order to have a detailed description of the adversaries that might be involved in attacks, Intel developed the Threat Agent Library (TAL) [4], a standardized reference that provides a description of the human agents that can threaten IT systems and other information assets.

TAL relies on a common set of characteristics, or attributes, to define each adversary, or "threat agent", in a unique way. In particular, the following eight attributes are defined: *Access*, *Intent*, *Limits*, *Objective*, *Outcome*, *Resources*, *Skill Level*, and *Visibility*. Each of these attributes can have different predefined values. For instance, the *Access* can be "internal" or "external", while the *Skill Level* can be "none", "minimal", "operational" or "adept". In TAL, the combination of these attributes' values results in a total of 21 different adversaries (e.g., Vandal, Employee Disgruntled, and Terrorist).

2.3 ADVISE

ADVISE (ADversary VIew Security Evaluation) [6, 13] is a security modeling framework that allows modeling adversaries and attack steps, aiming to analyze the probability that an adversary can achieve a certain goal, and the required effort. A model developed with ADVISE consists of two parts: the *Attack Execution Graph* (AEG) and the *Adversary Profile*.

The AEG describes the actions that an adversary has to follow to reach a certain goal, and it consists of items of five basic types. The *skill*, *knowledge*, and *access* items play the role of requirements (or prerequisites) for executing an attack, i.e., items that must be held by an adversary for enabling the execution of a specific attack. These items can also be gained as a result of a successful execution of an attack step. An *attack step* item represents a single step of an attack that can have different outcomes (e.g., success or failure), while a *goal* is an objective that the adversary wants to reach (like the logical access to a server).

In the Adversary Profile, we can define the profile of an adversary through the specification of a set of attributes, whose values determine if a particular adversary can reach a specific goal. Those attributes are: *name*, *decision parameters* (planning horizon, attack preference weights), *skills*, *initial access*, *initial knowledge*, and *goals*.

ADVISE is implemented as an atomic formalism in the Möbius framework [5, 22], which integrates the ADVISE execution algorithm [13] for simulating the adversary's behavior. The algorithm is based on Markov Decision Processes (MDPs), and consists of two steps repeated cyclically: (i) selection of the optimal attack step to be attempted next, and (ii) simulation of its outcome.

In order to reach a specific goal, in the first step, the algorithm selects the next attack step for an adversary among those that may lead to the goal and that can be actually executed (i.e., the attack steps for which the adversary has the prerequisites). The most attractive attack step also depends on adversary parameters, in particular on the *Planning Horizon*, which determines the number of steps in the future the adversary considers when making an attack decision; on the *Payoff* (gain of an adversary in case a goal is successfully reached), and on the *Cost of Detection*.

2.4 ADVISE Meta ontology

ADVISE Meta [10, 21] is a meta-level modeling framework that has been proposed to mitigate the complexity of building ADVISE models by hand, which can be a very difficult and time-consuming task. In the meta-level framework the model is built at an higher abstraction level (i.e., at meta-level), and the low-level ADVISE models are derived automatically. The system is described using generic built-in blocks and relationships that constitute the ontology of the framework, and that embed information on possible attacks in their definition. The elements of the ontology are briefly described in the following: the ontology consists of the following elements: those marked with a star (*) are also present in the plain ADVISE formalism.

- **Component.** It defines a base category of elements that can be part of a system. Some examples: *Device*, *OperatingSystem*, *Network*.
- **Relationship.** It defines a kind of relation that may exist between two components. Note that a relationship only applies to the specific kinds of components for which it is defined. Some examples: *onNetwork*, *storageDevice*, *canDamage*.
- **Attribute.** It represents a characteristic of a component, and can be used as parameter of the attack steps attached to the component. Some examples: *dataIntegrityControl*, *mediaPortEnabled*, *userAuthenticationType*.
- **Access*.** It defines an access that an adversary may have at the beginning of an attack, or which may be gained during the attack. Some examples: *InsiderAccess*, *LogicalAccess*, *PhysicalAccess*.
- **Skill*.** It defines a skill that an adversary may have in varying degrees of proficiency. Some examples: *BasicCyberOffense*, *Cryptanalysis*, *NetworkPenetration*.
- **Knowledge*.** It defines something that the adversaries may know beforehand, or that it may be acquired during the attack. The ADVISE Meta ontology from [10] includes only one knowledge: *FirewallRulesetKnowledge*.
- **Other State Variable.** It can be used to define state variables related to system components, which are also typically used to define adversary attack goals. Some examples: *Damaged*, *Disabled*, *MalwareInstalledOn*.
- **Attack Step*.** It defines a step of an attack that can be performed by an adversary. Some examples: *PhysicalDisable*, *GainUserCredentials*, *ModifyDataLocally*.
- **Adversary*.** It defines an adversary's profile with several characteristics. Some examples of built-in adversary templates: *ForeignGovernment*, *HackerGroup*, *OrganizedCrime*.
- **Metric.** Only one metric (*goalAchieved*) was defined in the base ontology from [10], but other metrics can be added.

The modeling process using ADVISE Meta consists of: (i) adding the components that are part of the system into the System Instance Diagram (SID) and setting the corresponding attributes; and (ii) connecting the components through the available relationships, based on the system architecture.

Once the model has been defined, the low-level model (i.e., ADVISE model) is generated, based on the definition

of the particular configuration of the system to be analyzed, i.e., one system diagram, one adversary, and a subset of available metrics. Other modeling elements are also automatically generated, including performance variable reward models, set studies, and the simulator. The attacks generated into the ADVISE model depend on the components (and their attributes) and the relationships between them, so if a particular component is not present in the SID model, the related attacks will not be derived.

The ADVISE Meta framework (and the generated ADVISE models) relies on some properties that are embedded in the definition of each attack step and in each adversary' profile. Each attack step has the following properties:

- *Attack Cost*, which determines the cost of the attack (e.g., expressed in dollars);
- *Attack Execution Time*, which expresses the time needed to perform the attack;
- *Precondition Expression*, which determines the prerequisites needed by the adversary to perform the attack (a few examples are given in Sect. 4.2);
- *Success Probability* (and *Failure Probability*), which defines the probability that an attack step succeeds (or not);
- *Detection Probability*, which provides the probability that the adversary will be detected during the attack.

When one defines the SID model, several attributes are associated with each Component, inherited from the ADVISE Meta ontology. Components of type "Device" have a total of sixteen attributes (e.g., *componentAnomalyDetectionStrength*, *resistanceToPhysicalDisable* and *userCyberSecurityAwareness*), while "Network" components have a total of thirteen attributes (e.g., *limitedIncomingProtocols*, *networkEncryptionStrength* and *networkWhiteList*). Some of these attributes are common to both types of components, e.g., *strengthOfUserAuthentication*.

The values assigned to the properties of each attack step, to the properties defined in the adversary's profile (i.e., *Planning Horizon*, *Payoff* and *Cost of Detection*, as explained at the end of Sect. 2.3), and to the attributes associated to each Component can have an impact on the adversary's behavior and on the probability of successfully reaching a goal. In Sect. 7.2, we will provide a concrete example at varying of one of these parameters.

3 Methodology: linking CAPEC, TAL and ADVISE Meta elements

In this work, we are interested in preliminary security evaluation conducted at an early stage of development. By definition, this type of analysis is subject to significant uncertainty about the effective attacks likely to be seen in operation. Thus, having a wide selection of attacks and adversaries would allow to conduct broad security analyses. After observing the ontology provided by the ADVISE Meta framework, two main aspects can be noted:

1. There are only a few attack steps in the ontology, categorized into just five attack types (i.e., damage and disable, malware, gain access, compromise data integrity, and compromise data confidentiality). The CAPEC database, on the other hand, provides more than 500 attack patterns.
2. The adversaries' templates provided by the ontology are still too generic. For example, considering the Hacker Group, any individual with enough IT skills could be potentially classified as "hacker". Conversely, the adversaries' profiles proposed by TAL are more specialized (e.g., Government Cyberwarrior, Thief, Civil Activist).

In this section, we present the refined methodology, initially proposed in [15], which is based on the definition of relationships between the properties identified in the CAPEC, TAL, and ADVISE Meta domains. The identified relationships can be used to extend the ADVISE Meta framework with CAPEC attacks and TAL adversaries (see Sect. 4), and thus use them to build ADVISE models for security analysis.

3.1 From CAPEC sections to TAL attributes

We have first identified the relationships between CAPEC and TAL, to understand how the information found in the CAPEC sections are linked to TAL attributes. This mapping allows CAPEC attack patterns to be described in terms of attributes of TAL adversaries. We have identified the following relationships between CAPEC and TAL, which are also summarized in the left part of Fig. 1:

- According to the TAL's *Intent*, which defines the adversary's intention to cause harm, an adversary can be "Hostile" or "Non-Hostile". The "Description" section of a CAPEC entry usually provides some inferable information related to this TAL attribute.
- In TAL, the *Access* attribute can have two values, "Internal" or "External", denoting the extent of the adversary's access to the system's assets. A list of prerequisites that an adversary must satisfy in order to execute the attack (including the access to assets) is usually given in the "Prerequisites" section of CAPEC. If this section is not detailed enough, additional information can be derived from the "Description" section.
- The TAL's *Limits* attribute defines the ethical and legal limits of an adversary, and how much the adversary is prepared to break the law. Four different values are allowed ("Code of Conduct", "Legal", "Extra-Legal

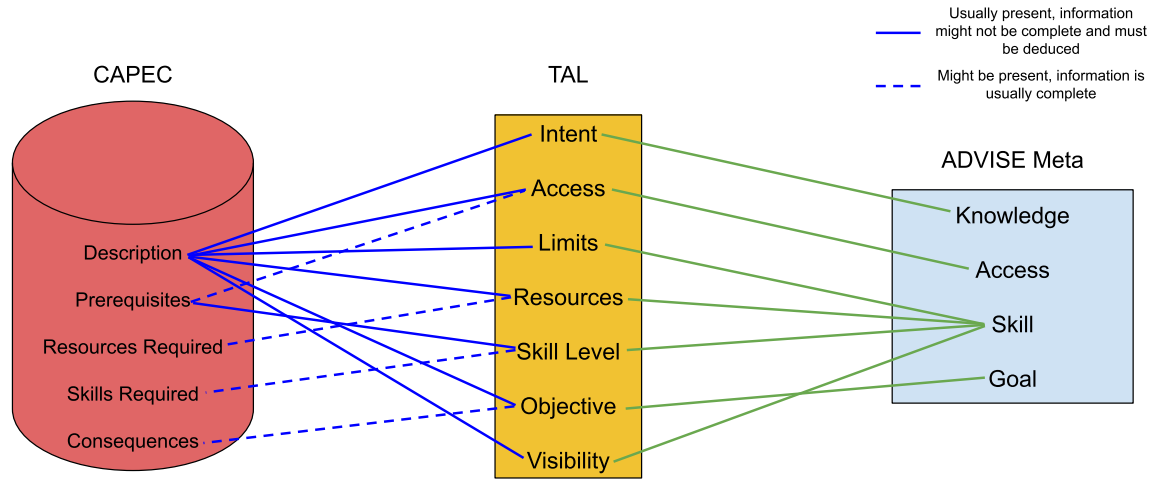

Fig. 1 Relationships between CAPEC sections, TAL attributes, and ADVISE Meta elements

Minor" and "Extra-Legal Major"). No specific section containing information about ethical and legal limits exists in CAPEC, so one must infer such information from the "Description" section.

- The type of organization and the amount of resources owned by the adversary to run attacks are defined by the TAL's *Resource* attribute, which can have six different values ("Individual", "Club", "Contest", "Team", "Organization", and "Government"). The "Resources Required" CAPEC section provides a little information about specific equipment, software, and other kinds of resources needed in order to perform the attack. If this section is not present, this information must be inferred from the "Prerequisites" section.
- The expertise of an adversary is determined by the TAL's *Skill Level*. Four different values are available ("None", "Minimal", "Operational" and "Adept"). The "Skills Required" CAPEC section describes the skill level needed to execute the attack. Additional information can be found in the "Prerequisites" section.
- In TAL, the *Objective* attribute defines the goal that an adversary wants to achieve. Five different values are allowed ("Copy", "Destroy", "Injure", "Take" and "Don't Care"). This attribute can be associated to the "Consequences" CAPEC section. Here the scope (e.g., confidentiality, integrity, or availability), and the impact of the attack are described. If this section is not present, the information must be inferred from the "Description" section.
- The extent to which the adversary intends to hide/reveal her or his identity is described by the *Visibility* attribute, which can have one of the following values: "Overt", "Covert", "Clandestine" and "Don't Care". No dedicated section is available in CAPEC, so the "Description" section should be checked.

Note that these relationships do not depend on the underlying modeling framework. On the other hand, the relationships are not always easy to identify, as the CAPEC database sometimes does not provide all the necessary information. For example, in CAPEC entries, while the "Description" and "Prerequisites" sections can always be found, other useful sections like "Resources Required" and "Skills Required" might be missing. In such cases, the missing information must be inferred and interpreted from the available sections, when possible.

3.2 From TAL attributes to ADVISE Meta elements

As a second step, we have identified the relationships between TAL and ADVISE Meta elements, to represent TAL adversaries' profiles in the ADVISE Meta framework. In the right part of Fig. 1, we can see how the TAL attributes are related to the ADVISE Meta elements:

- The TAL's *Intent* attribute can be associated to the *Knowledge* concept of ADVISE Meta, as the adversary knows if she/he has an hostile intent or not. To execute a malicious attack, a malicious intent is usually necessary, so we have added a new Knowledge element named *Intent* to the ontology, to be associated to malicious adversaries.
- The *Access* attribute from TAL can be intuitively related to the Access element (ADVISE concept) already present in the ontology, called *InsiderAccess*.
- The TAL's *Limits* attribute represents the ethical and legal limits of the adversary. This concept has been associated to a Skill in ADVISE Meta, which can be interpreted as the adversary's ability to act at different levels of legality. We have, therefore, added a new Skill element called *Limits* to the ontology, to model such attribute in ADVISE Meta.

- Also the *Resources* TAL's attribute can be associated to a Skill in ADVISE Meta, because it can be seen as the adversary's ability to gain the appropriate resources required for attacks. A new Skill element named *Resources* has been added to the ontology.
- The TAL's *Skill Level* represents the expertise level of an adversary. We have related this attribute to its counterpart in ADVISE Meta, i.e., *Skill*. A new Skill element called *SkillLevel* has been added to the ontology of the framework.
- No ADVISE Meta element can be easily associated to TAL's *Objective* attribute. *Goal* is the closest concept in the ADVISE formalism, which is however associated to attack steps. Therefore, one should not add this element to the ontology when modeling adversaries' profiles, but when modeling attack steps.
- The degree of importance for the adversary to remain hidden is represented by TAL's *Visibility* attribute. Conceptually it is something that the adversary knows, but because it has more than two possible values, it cannot be associated to a Knowledge element, which is instead a Boolean property. Therefore, a new Skill element named *Visibility* has been added to the ontology, which can be interpreted as the extent to which the adversary intends to hide/reveal her or his identity.

In ADVISE, a Skill element is defined as an integer which can have integer values between 0 and 1000. The numerical thresholds used to represent the different TAL attributes' values are shown in Table 1. This should only be interpreted as a translation of the TAL attributes' values from a qualitative to a quantitative point of view, needed to practically represent these attributes in the framework. As a default setting, we have equally distributed the qualitative values of each TAL attribute in a quantitative range between 0 and 1000. This setting can be anyway modified according to the modeler's need and could be made parametric using global variables.

Table 1 Assignment of numerical values to TAL attributes. Table adapted from [15]

TAL attribute	TAL attribute value	Numerical value
Intent	Not hostile	0
	Hostile	1
InsiderAccess	Outsider	0
	Insider	1
Limits	Code of Conduct	250
	Legal	500
	Extra-legal minor	750
	Extra-legal major	1000
Resources	Individual	0
	Club	200
	Contest	400
	Team	600
	Organization	800
	Government	1000
SkillLevel	None	0
	Minimal	250
	Operative	750
	Adept	1000
Visibility	Overt	1000
	Covert	500
	Clandestine	250
	Do not care	0

3.3 From CAPEC sections to ADVISE Meta elements

The extension of the ADVISE Meta ontology with CAPEC attacks may require further information that can be found in CAPEC sections. In particular, when a new attack step is added to the ontology, the target of the attack (i.e., the involved system's component) should be specified. In ADVISE Meta, each attack step is associated with a single component, but the outcome of the execution of an attack step can have an impact on other components according to the relationships involved (the "Relationship" element is defined in the ontology, see Sect. 2.4).

Such kind of information can be derived from the CAPEC "Description" section. Moreover, the "Description" and "Precondition" sections can be used to derive additional information on attacks, like preconditions related to the existence of particular architectural components.

4 Extension of ADVISE Meta

The methodology presented in Sect. 3 has been applied to extend the ontology of the ADVISE Meta framework, to include some representative CAPEC attack patterns and all the adversaries' profiles provided by TAL.

4.1 Extension of ADVISE Meta with TAL adversaries' profiles

Following the TAL/ADVISE Meta relationships and the attributes' values specified in the TAL library, we translated the TAL adversaries' profiles to ADVISE Meta adversaries.

To exemplify the application of the methodology, we discuss how to extend the ontology with the *Terrorist* profile (another illustrative example can be found in [15]). In TAL this adversary has "External" Access, "Hostile" Intent, "Adept" Skill Level, "Extra-Legal Major" Limits, "Organization" Resources and "Covert" Visibility. To add this profile

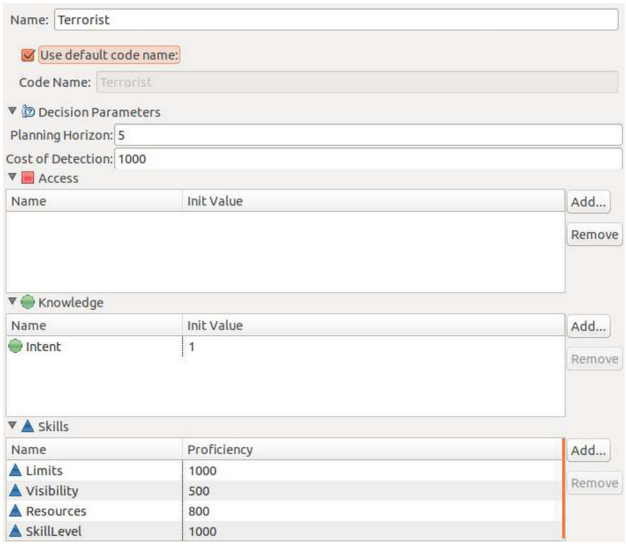

Fig. 2 TAL "Terrorist" adversary's profile visualized in the ADVISE Meta framework

to the ontology, we have created a new ADVISE Meta adversary called "Terrorist".

According to Table 1, the Terrorist profile has the following attributes values (Fig. 2):

- *InsiderAccess* with value 0 ("External"), i.e., the Access is not added to the profile;
- *Intent* with value 1 ("Hostile");
- *Limits* with value 1000 ("Extra-Legal Major");
- *Visibility* with value 500 ("Covert");
- *Resources* with value 800 ("Organization");
- *SkillLevel* with value 1000 ("Adept").

4.2 Extension of ADVISE Meta with CAPEC attacks

To show how it is possible to extend the ontology of the framework with new attacks by using the proposed methodology, we have chosen some representative attack patterns involving the core components of the CIA triad, i.e., Confidentiality, Integrity and Availability [2]. We have focused on the following CAPEC attacks (the affected security properties are specified in brackets):

- CAPEC-94: Adversary in the Middle (Confidentiality and Integrity);
- CAPEC-125: Flooding (Availability);
- CAPEC-153: Input Data Manipulation (Integrity);
- CAPEC-248: Command Injection (Confidentiality, Integrity, and Availability);
- CAPEC-549: Local Execution Of Code (Confidentiality, Integrity and Availability).

Although we have considered only a few representative CAPEC attacks, the methodology illustrated in Sect. 3 is applicable to all the attack patterns defined in the CAPEC database. However, in handling the largeness of the CAPEC database (more than 500 attacks), two main problems arise: (i) adding manually all the CAPEC attacks to the ontology would be an error-prone and very time-consuming activity, and (ii) the complexity of the derived ADVISE models will become unmanageable due to an exponential growth in the number of states to explore.

The identification of the relationships between CAPEC, TAL, and ADVISE Meta (described in Sect. 3) and the subsequent extension of the framework ontology have been done entirely manually. For a complete integration of the CAPEC database in the ADVISE Meta ontology, an automatic way to import the CAPEC attacks would be essential. CAPEC is also available in XML format, therefore, it would be a good candidate for automatic processing through the employment of model transformation languages and tools [9]. As previously explained, there are some information that are hidden or not always present in CAPEC sections, so further investigation would be necessary to implement the automatic process. This could be an interesting future work.

By adding more attack patterns to the ontology, the generated ADVISE models will become more complex and the number of states to explore will grow exponentially. However, as already explained in Sect. 2.4, the automatic generation of attacks depends on some modeling constraints that can (at least partially) mitigate the problem. In particular, an attack will not be generated if the related component is not present in the SID model. Moreover, to limit the number of generated attack steps, it could be helpful to disable one or more attacks on one or more specific components' instances in the SID model. For example, the modeler may believe that a specific workstation cannot be the target of flooding attacks, because of how the system is deployed. In this case, one could add some "flag" attributes to specific component elements in the ontology, and then use them as prerequisites of the attacks, so that some specific attacks can be disabled.

4.2.1 Flooding

Here, we describe the application of the methodology to the Flooding attack. Flooding is a Denial of Service attack, in which the adversary wants to deplete the resources of a target system to deny the access to users. In CAPEC, this attack is classified under the "Software" and "Communication" domains, and under the "Abuse Existing Functionality" attack mechanism.

This attack has been modeled by adding a new attack step to the ontology (Fig. 3). The targets of this attack are those system components that are classified as Device (according to the "Description" section of the CAPEC entry). Precondi-

Fig. 3 "Flooding" attack step added to the ontology of ADVISE Meta

tions for this attacks are at least "Extra-Legal Minor" Limits, and the adversary's intent to cause damage to the system (from the "Description" section of the CAPEC entry). Furthermore, at least "Club" Resources are required, because a script or a network able to generate an high number of requests is needed (from "Resources Required" section of the CAPEC entry).

Considering the default setting for the values of the TAL attributes presented in Table 1, the above preconditions are met if the value of the Limits TAL attribute is greater or equal than 750 (value of the "Extra-Legal Minor" Limits), and the value of the Resources TAL attribute is greater or equal than 200 (value of the "Club" Resources). Another precondition is the access to a network (still from the "Description" CAPEC section). Thus, the *Precondition Expression* of the Flooding attack step, i.e., the Boolean expression that has to be satisfied to perform the attack, is the following:

```
return((${Limits}>=750)
&& (${Intent})
&& (${Resources}>=200)
&& (${AccessToNetwork}));
```

To better model this attack, the *handlingRate* attribute has been added to the ontology for the Device element, denoting the rate at which the device is able to handle requests. This attribute can have values in the range between 0 and 10: low (0–3), medium (4–6), and high (7–10). In the following, we show the expression of the *Success Probability* for the Flooding attack:

```
if (${handlingRate}>=7) return 0.2;
else if (${handlingRate}>=4) return 0.5;
else return 0.9;
```

The *handlingRate* attribute has an impact on the probability that an adversary can successfully perform an attack: the lower its value, the higher the probability to perform a successful attack. When creating the SID, the modeler can adjust the value of the attribute, according to the characteristics of the device to be represented in the model.

4.2.2 Adversary in the middle

In the CAPEC-94 Adversary in the Middle, also known as Man in the Middle, the adversary takes position between two network nodes, to retrieve or modify the messages exchanged between the two victims before forwarding them to the other node. In the CAPEC entry, the "Description" section reports that the possible targets of this kind of attack are components classified as Device.

To model this attack in ADVISE Meta framework, we have added three different attack steps to the ontology, following the "Execution Flow" section of the CAPEC entry. In the first step (*MITMDetermineCommunicationMechanism*), the adversary identifies the mechanism used for the communication between the two nodes. Preconditions for this step are "Extra-Legal Minor" Limits (i.e., with value of at least 750), "Hostile" Intent (i.e., the Intent element is present), and "Team" Resources (i.e., with value of at least 600). We retrieved this information from the "Description" section of the CAPEC entry. Moreover, according to the "Prerequisites" section of CAPEC, two components communicating through a network must be present in the system model (in this case, a client and a server). The *Preconditions Expression* of this attack step is the following:

```
return((${Limits}>=750)
&& (${Intent})
&& (${commAccessAlready}==0)
&& (${Resources}>=600)
&& (${ClientExists})
&& (${ServerExists}));
```

After successfully performing the attack, the *CommunicationAccess* Access is gained by the adversary. In the subsequent attack step (*MITMPositionBetweenTargets*), the adversary takes position inside the network to intercept the messages exchanged between the two victims. The preconditions of this attack are the same as the previous ones, with the addition of the *CommunicationAccess* Access. If this attack is successfully completed, the adversary gains the *MonitoredNetworkAccess* Access.

In the last attack step (*MITMMonitoringNetworkAccess*), the adversary attempts to read or modify the intercepted data.

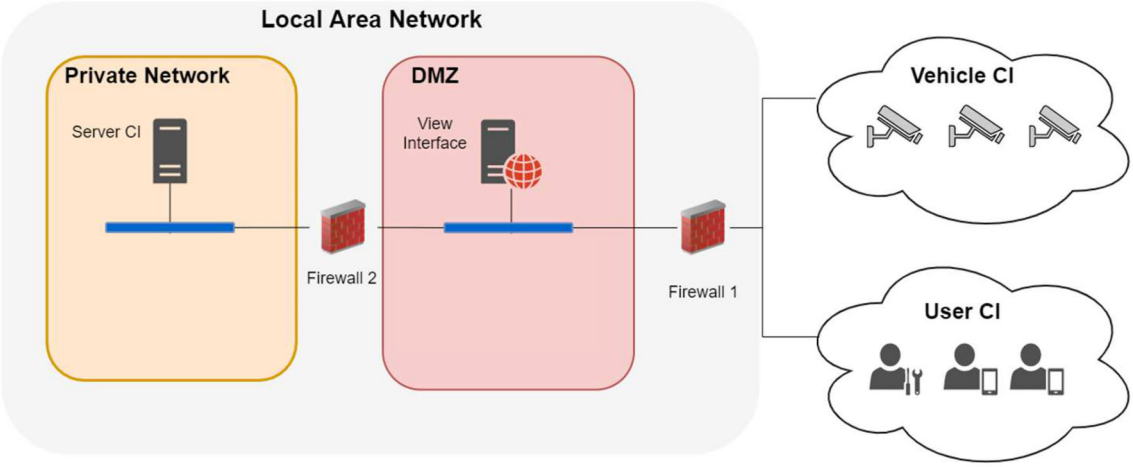

Fig. 4 DMZ architecture for the SPaCe system

For the previous attack steps, no major law breaches were required, while a major illegal action is required to carry out this final step. Thus, additional preconditions for this attack step are "Extra-Legal Major" Limits, along with the *MonitoredNetworkAccess* Access obtained in the previous step. If the attack step is successful, the attacker finally obtains the *UseInterceptedData* Access.

5 Case study: SPaCe system

The Smart Passenger Center (SPaCe) system [14] is designed for the orchestration and supervision of the mobility of public transport, improving the passenger experience, trying to prevent security breaches and facilitating investigations after a system violation.

The SPaCe system receives the data collected by sensors (mainly video cameras) installed on board of vehicles (trains and buses), and processes them to infer the status of vehicles in real time. Through the analysis of the collected data, the system is able to obtain information on the occupancy level of vehicles, informing users and administrators and thus enabling the optimization of passengers flow. The system can also identify possible dangerous situations on board, such as the presence of suspicious objects or damaged equipment.

The network architecture adopted for the SPaCe system is based on the DMZ (DeMilitarized Zone) model, which generally aims to protect a private network from external threats. A DMZ is a subnetwork that exposes services to external networks that are not considered secure (e.g., the Internet), and it is located between the internal network and the external one, ensuring that if a machine inside the DMZ is attacked, it does not directly affect the private network. A typical DMZ architecture consists of a private LAN network, a first firewall that acts as a filter between it and the DMZ network, and a second firewall with the same purpose between the DMZ network and the external network. DMZ hosts have the dual task of (i) receiving information and requests from the external network and forwarding them to the private network hosts, and (ii) receiving messages from the private network hosts and forwarding them to the external network.

Figure 4 shows the system-level architecture of the SPaCe system and its components. The main components are the following:

- User Configuration Item (CI), located on the external network, as it is installed on the devices (e.g., PCs or smartphones) owned by users of the system, who can use them to retrieve information about on-board conditions (e.g., the occupancy level of the vehicles).
- Vehicle Configuration Item (CI), positioned on the external network, as it is installed on board of the vehicles. It supplies the data coming from vehicle sensors.
- Server Configuration Item (CI), located within the private network. Its objective is to provide the main functionalities of the system and to protect the stored data. A sub-component, called View Interface is located in the DMZ and it is responsible for retrieving, organizing and delivering information to be displayed to final users.

6 Security analysis of the SPaCe system

In this section, we describe the security analysis performed on the SPaCe system, using the methodology and the framework extension described in Sects. 3 and 4, respectively.

6.1 Objectives of the analysis

Thanks to the extensions to the ADVISE Meta modeling framework, we are able to perform broader security analyses with respect to the base ontology of the framework. In particular, the extended framework allows us to: (i) compute the probability that an adversary can successfully reach a goal in a given time window; (ii) derive the attack path (i.e., the sequence of attack steps) that allows the adversary to reach a goal; (iii) perform sensitivity analysis at varying of attack patterns and adversaries' profiles; (iv) compare different implementation solutions; and v) identify the system's components that can be more probably attacked by the adversaries.

These analyses are carried out during an initial phase of system development, in which just the main architectural aspects of the system are known and the exact security mechanisms that will be adopted are not yet defined.

Given the early-stage nature of the analyses, besides offering preliminary quantitative indications on the targeted security aspects, the derived results can provide qualitative indications to guide the design process of the system. For example, analysis results can be used to support architectural decisions, to identify the most vulnerable components with respect to different CAPEC attacks and TAL adversaries, and to compare the impact of adopting different implementations and security mechanisms.

6.2 Modeling process

In this section, we provide the steps required for the security analysis of the SPaCe system using the extended ADVISE Meta framework. We first show the architectural model of SPaCe, then we provide instruction on how to define goals, adversaries and metrics in ADVISE Meta framework. Finally, we give some details about the security models generation and simulation.

6.2.1 Definition of the architectural model

The first step is the creation of the architectural model, also called System Instance Diagram (SID), of the SPaCe system by using ADVISE Meta framework. This model, shown in Fig. 5, has been created following the architecture described in Sect. 5. The architectural components have been added to the model using the elements from the ADVISE Meta ontology, and the components have been connected to each other through relationships. The main relationship involved in the model is the *onNetwork* relationship, which connects components of type Device (e.g., the ServerCI) to components of type Network (e.g., the LAN), to indicate that the device is on the network. Another relationship that appears in the model is the *readData* relationship, which connects a Data element to a Sensor component, meaning that the sensor collects those data from the environment.

The main architectural components introduced in Sect. 5 are represented in the model as it follows:

- User CI is represented in the model by a Device of type Workstation. In fact, the Workstation type defines a host supporting the interaction of a human user with application functions, thus including PCs, tablets and smartphones.
- Vehicle CI is represented by a Device of type Sensor.
- Server CI is represented by two different Devices of type Server, called ViewInterface and ServerCI.

Several attributes, inherited from the ontology of the framework, are associated to each component of the SID.

Referring to the analyses shown in the following sections, we focus on the *strengthOfUserAuthentication* attribute: it determines the level of security measures used to authenticate users for that particular component. The attribute can have the following values: 0 (no authentication or authentication with short or poorly validated passwords); 4 (long and strongly validated passwords, e.g., using Bloom Filters); 6 (use of a Primary Key Infrastructure, PKI); 7 (two-factor authentication); 9 (biometrics). This attribute is associated to each component of type Device and Network (i.e., every component of the SID model in Fig. 5, with the exception of InputData).

6.2.2 Definition of the goals

Once the architectural model of the system is defined, the objectives (Goals) of the adversaries must be specified. The modeler must associate each goal with an element related to a specific component in the SID (e.g., the components' accesses, like Logical Access). For each component of the system, one or more representative goals involving that component have been defined. The added goals have been chosen among the CAPEC attacks added to the ontology. In particular:

- For each component of Network type, two objectives have been defined: one regarding access to the network (*NetworkAccess*), and another regarding the use of intercepted data in a potential Man-In-The-Middle attack (*UseInterceptedData*).

- For each component of Device type (in this case, the Firewall and Server type components), three objectives have been defined: logical access to the device (*LogicalAccess*), installation of a ransomware software

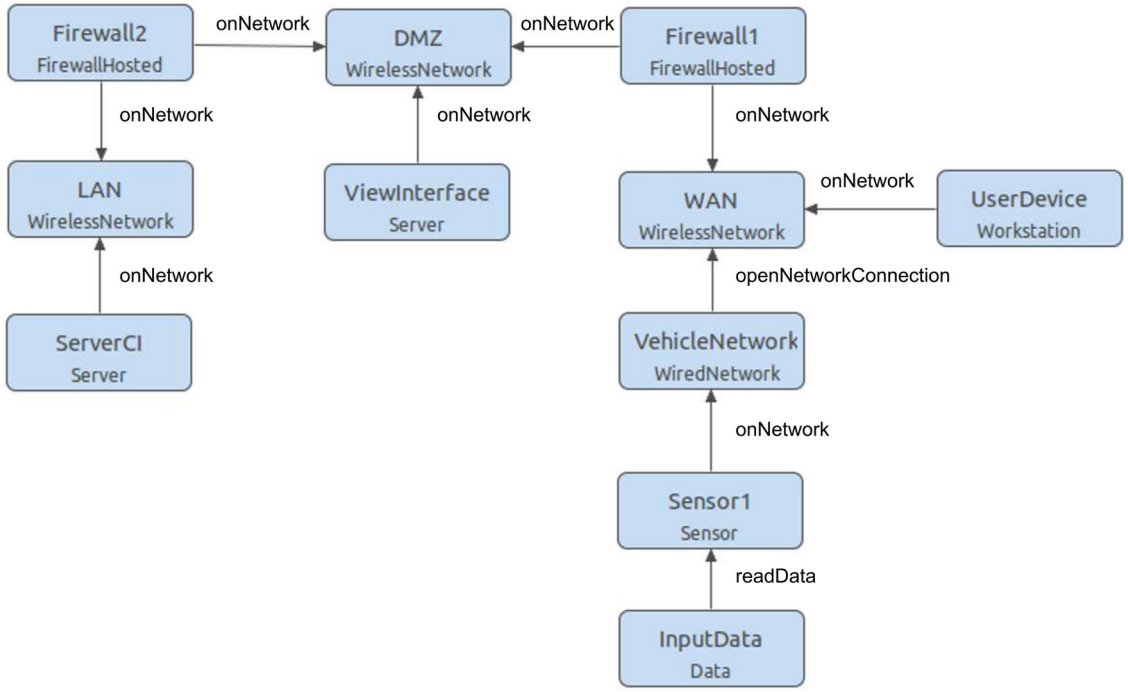

Fig. 5 Architectural model (SID) of SPaCe system created with ADVISE Meta framework

(*RansomwareInstalled*) and unavailability of the device (*UnableToService*).

- For components of Sensor type, the goal *ManipulatedInputData* has been defined. Achievement of this goal can be obtained, for example, with attacks that attempt to deceive the images classification algorithms by exposing an *ad hoc* crafted sign (Adversarial Patch) to a video camera. In the analyzed scenario, we have only one sensor (Sensor1).

In summary, considering all the components of the architectural model (Fig. 5), the following 21 different goals have been defined (the names are derived from the goal names and the names of the involved element in the SID):

- DMZNetworkAccess
- DMZUseInterceptedData
- Firewall1LogicalAccess
- Firewall1RansomwareInstalled
- Firewall1UnableToService
- Firewall2LogicalAccess
- Firewall2RansomwareInstalled
- Firewall2UnableToService
- LANNetworkAccess
- LANUseInterceptedData
- Sensor1ManipulatedInputData
- ServerCILogicalAccess
- ServerCIRansomwareInstalled
- ServerCIUnableToService
- VehicleNetworkNetworkAccess
- VehicleNetworkUseInterceptedData
- ViewInterfaceLogicalAccess
- ViewInterfaceRansomwareInstalled
- ViewInterfaceUnableToService
- WANNetworkAccess
- WANUseInterceptedData

6.2.3 Definition of the adversaries

The next step is to define the adversaries' profiles to be used in the analysis. To have the broadest possible view on the threats represented by the adversaries, all 21 adversaries' profiles of the TAL library added to the ontology during the extension of the framework have been used. All 21 previously defined goals have been associated with each adversary. This implies that each adversary can potentially aim to achieve all 21 defined goals, allowing for a broad comparison against various attacks and adversaries' profiles. Concerning the initial access to the system, it has been assumed that the adversary: owns a device (UserDevice) connected to the WAN, and has physical access to the sensor (Sensor1) located on the vehicle.

6.2.4 Definition of the metrics

In order to evaluate the probability of achieving goals by the adversaries, a *goalAchieved* metric has been associated with each goal. By running the simulation, this set of metrics

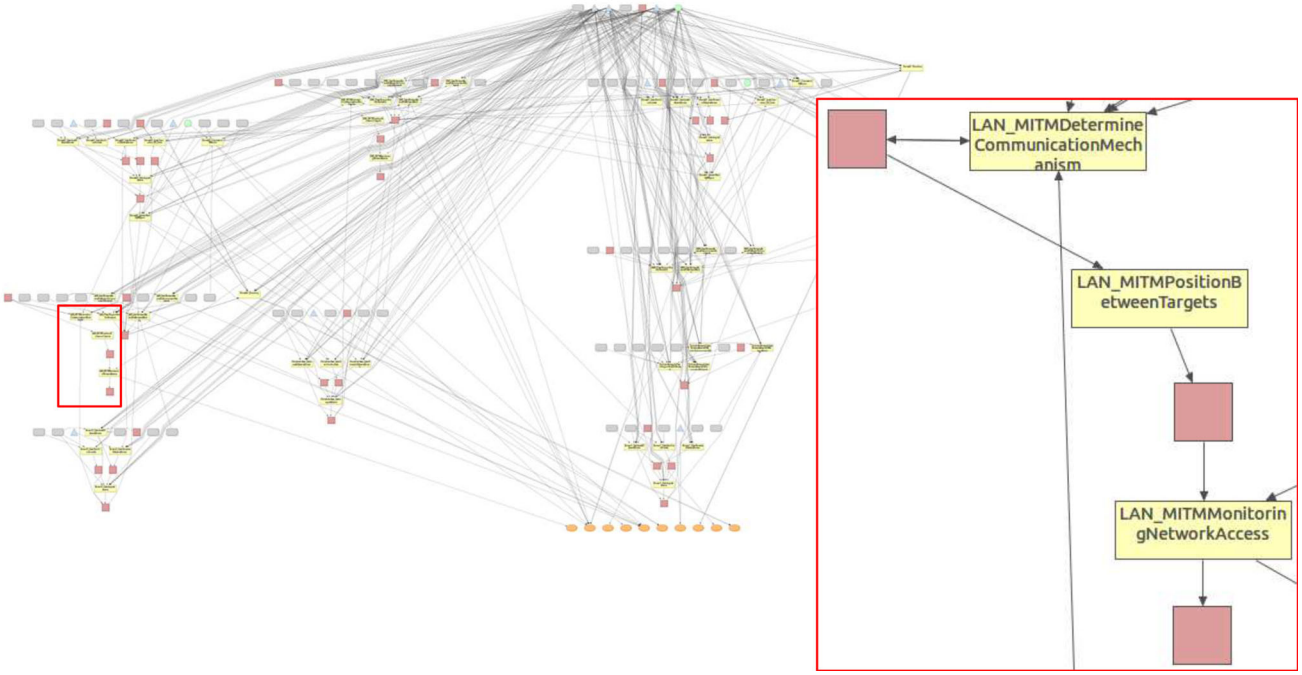

Fig. 6 ADVISE model generated by the tool for the Terrorist adversary. In the red box, details of the Man in the Middle attack steps on the LAN are highlighted

provides the probability that, as time varies, the adversary reaches each of the goals.

To define such metrics, we must select the goal to observe, the first and the last observation time-point and the length of time between the observations. One or more metrics can be assigned to each adversary. Through the model generator, ADVISE Meta will automatically create a reward model associated to each metric. In particular, the metrics are computed by an instant of time rate reward variable that returns the number of tokens in the goal element at the different observation times.

6.2.5 Generation and simulation of the models

The ADVISE models associated with each adversary's profile have been derived using the model generator (the framework generates a different ADVISE model for each adversary). Creating such models manually would have been a time-consuming and error-prone task. This complexity is clearly visible in Fig. 6 (left part), which shows the generated ADVISE model for the Terrorist adversary. A detail of the model is highlighted in the red box of Fig. 6 (right part), showing the attack steps related to Man In the Middle attack on the LAN component. This is one of the CAPEC attack patterns that we added to the ontology.

Each attack step has some prerequisites that the adversary must fulfill in order to successfully complete it, i.e., elements like Skills, Accesses, Knowledge and State variables. For example, the *LAN_MITMPositionBetweenTargets* attack step requires the *LAN_CommunicationAccess*, which is obtained after successfully completing the *LAN_MITMDetermine CommunicationMechanism* attack step. Indeed, the completion of each attack step can lead to the adversary gaining one or more elements (e.g., Accesses). Then, the attack path followed by the adversary can lead to one goal. As an example, a goal can be associated with the *UseInterceptedData* Access obtained after completing the *LAN_MITMMonitoringNetworkAccess* attack step. The names of these attack steps have been derived from the "Execution Flow" section of the CAPEC–94 entry.

Once the ADVISE models are derived, it is possible to simulate them using the simulator integrated in the Möbius tool, and to observe the values associated with the previously defined metrics. The tool automatically generates one simulator for each ADVISE model.

7 Analysis scenarios and results

The analyses we carried out are described in this section. Unless differently specified, we computed the probability of achieving the different goals within 96 time units (hours). The adopted interval of time allow us to observe significant variations in the probability of completing the attacks and successfully reaching the goal. We ran each simulation with, respectively, 1000 and 10,000 minimum and maximum

number of batches (simulation runs), converging within 95% probability in a 0.1 relative interval.

As part of the extension of the framework, we assigned numerical values to the properties of attack steps and adversaries' profiles. For the attack steps, the values have been derived from CAPEC, inferring them from the entries of the database (as an example, we have already discussed the Precondition Expression for the Flooding attack in Sect. 4.2.1). Regarding the adversaries, we have used the following setting: *Planning Horizon*=5, *Payoff* = 500,000 and *Cost of Detection*=1000.

If not differently specified, the properties of the model (e.g., the components' attributes in the SID model) have been set to the default values as they are defined in the original ADVISE Meta framework. Specific sensitivity analyses should be performed at varying of the values of these parameters to evaluate their impact on the targeted metrics (see Sect. 7.2 for a concrete example).

7.1 Analysis at varying of adversaries' profiles

Table 2 shows the results related to the probability of achieving the goals defined in Sect. 6.2.2, at varying of some adversaries' profiles. The adversaries have been chosen among some of the most representative for the SPaCe system: Vandal, Employee Untrained, Employee Disgruntled and Terrorist. For this analysis, the system authentication level has been configured as low. In particular, the *strengthOfUserAuthentication* parameter defined in each element of the SID model has been set to 0. This means that authentication is either absent or of an extremely low level (e.g., short passwords are used, with no constraints on characters).

The first observation that can be made is that, for the Employee Untrained adversary, the probability of achieving each objective is always 0. This happens because all the involved attacks require the adversary's hostility, i.e., the adversary's profile must have the Intent attribute of type Knowledge with value equal to 1. The Intent attribute of the Employee Untrained adversary has value 0, therefore, he can never reach any goal.

The Employee Disgruntled adversary is instead able to achieve the sensor data manipulation goal (*Sensor1 ManipulatedInputData*), but not any other goal. The values of the attributes associated with this adversary are in fact inadequate to reach the other goals. In particular, the Resources attribute of this adversary is of type Individual, therefore, the adversary's profile in ADVISE Meta has the Skill Resources with value 0. Most of the involved attacks instead require that the adversary has at least some club-level resources (e.g., the Flooding attack, see the Precondition Expression in Sect. 4.2.1), meaning that in ADVISE Meta the Skill Resources must have value greater or equal to 200 (see Table 1).

Comparing the results obtained for the Vandal and Terrorist adversaries, we can see that the probabilities that the Terrorist can achieve each goal is usually higher than those of the Vandal (in some cases for the Vandal it is even equal to 0). However, on some goals (e.g., *Firewall2UnableToService*), the probability of success obtained by the Vandal is instead higher than what is achieved by the Terrorist. In fact, some goals (e.g., *Firewall2UnableToService*) are actually much more attractive for a Vandal adversary than for a Terrorist, since the latter will have other exploitable attack paths that are more attractive and more convenient to attempt.

7.2 Analysis at varying of the attack step success probability

As mentioned in Sect. 2.4, there are several parameters embedded in the ADVISE Meta framework (and in the generated ADVISE model) whose setting can have an impact on the adversary's behavior and on the probability of successfully reaching a goal.

In this section, we perform a sensitivity analysis considering one of these parameters, the *Success Probability* property defined for each attack step.

The analysis has been performed considering the Man In the Middle attack pattern (already discussed in Fig. 6) and the Terrorist profile. For successfully completing the Man In the Middle attack and reaching the *UseInterceptedData* goal, the adversary must successfully execute three sequential attack steps: *LAN_MITMDetermineCommunicationAccess*, *LAN_MITMPositionBetweenTargets* and *LAN_MITM MonitoringNetworkAccess*.

The three plots in Fig. 7 show the probabilities of successfully completing the three attack steps at varying of the *Success Probability* of the *LAN_MITMMonitoringNetworkAccess* attack step (the last step before achieving the goal). It is interesting to note that when increasing this probability within the interval [0.1;0.7], the computed metrics remain almost constant: this means that in the identified interval this parameter is actually not affecting the adversary's behavior. On the contrary, when the *Success Probability* becomes greater than 0.7, we can note a steepening of all the three curves: not only the one directly describing the *LAN_MITMMonitoringNetworkAccess* attack step (the bottom one, in gray), but also the ones of the two preceding attack steps. In fact, for values greater than 0.7, the Man In the Middle attack becomes more appealing for the adversary, thus increasing the attack attempts and, in turn, increasing the probability of successfully completing the attack in the considered time interval (96 h).

Table 2 Probability of successfully achieving goals as the adversaries' profiles vary

Goal	Vandal	Employee untrained	Employee disgruntled	Terrorist
DMZNetworkAccess	1	0	0	1
DMZNetworkAccess	1	0	0	1
DMZUseInterceptedData	0	0	0	0.8191
Firewall1LogicalAccess	0.9667	0	0	0.9999
Firewall1RansomwareInstalled	0	0	0	1
Firewall1UnableToService	0.9834	0	0	0.9548
Firewall2LogicalAccess	1	0	0	1
Firewall2RansomwareInstalled	0	0	0	1
Firewall2UnableToService	0.8264	0	0	0.0551
LANNetworkAccess	0.9647	0	0	0.8058
LANUseInterceptedData	0	0	0	0.3617
Sensor1ManipulatedInputData	1	0	1	1
ServerCILogicalAccess	0.8279	0	0	0.7756
ServerCIRansomwareInstalled	0	0	0	0.8978
ServerCIUnableToService	0.522	0	0	0.0615
VehicleNetworkNetworkAccess	1	0	0	1
VehicleNetworkUseInterceptedData	0	0	0	0
ViewInterfaceLogicalAccess	0.9995	0	0	0.3462
ViewInterfaceRansomwareInstalled	0	0	0	1
ViewInterfaceUnableToService	0.9698	0	0	0.9788
WANNetworkAccess	1	0	0	1
WANUseInterceptedData	0	0	0	0.3657

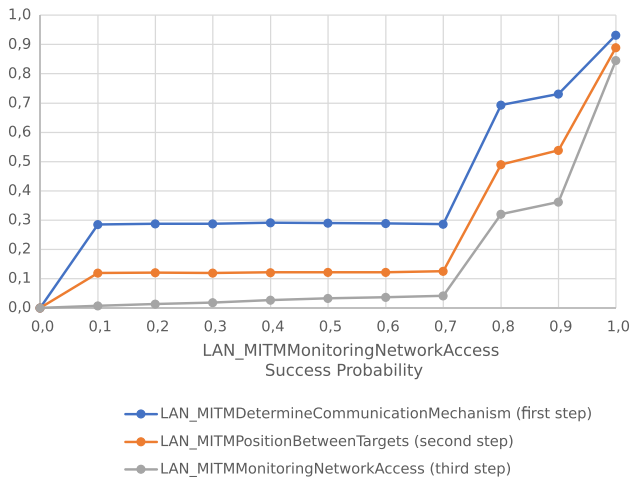

Fig. 7 Probability of successfully achieving the *LANUseIntercepted-Data* goal for the Terrorist adversary, as the *Success Probability* parameter of the *LAN_MITMMonitoringNetworkAccess* attack step varies. The probabilities of successfully completing each of the three attack steps composing the MITM attack are shown

7.3 Analysis at varying of implementation characteristics of the system

Figure 8 shows the results of a simulation in which two different system configurations are compared. The first uses a low authentication level (as in the previous scenario), while the second was configured with a high system's authentication level, i.e., more severe restrictions are placed on passwords. In the architectural model, the second configuration is represented by setting the *strengthOfUserAuthentication* parameter to 4, on each component of the system.

The results shown in the plot refer to a subset of the goals considered in Sect. 6.2.2, evaluated only for the Terrorist adversary, on the two configurations. As it can be immediately noted, for most goals the success probability with the "strong" authentication configuration is significantly lower than with the "weak" authentication configuration, and in most cases it is even zero.

Conversely, for some goals, the probability of the attacker achieving them *increases* with the stronger authentication method. It is the case, for example, of the *LanNetworkAccess* goal: with a strong user authentication, the adversary (following the execution algorithm mentioned in Sect. 2.3) "realizes" that reaching some goals (e.g., the *Firewall1LogicalAccess* goal) would be too costly, and then she/he focuses on other goals that are more rewarding (e.g., the *LanNetworkAccess* goal). Note that this does not mean that the adversary is able to cause more damage, but only that it is more likely to access certain parts of the system.

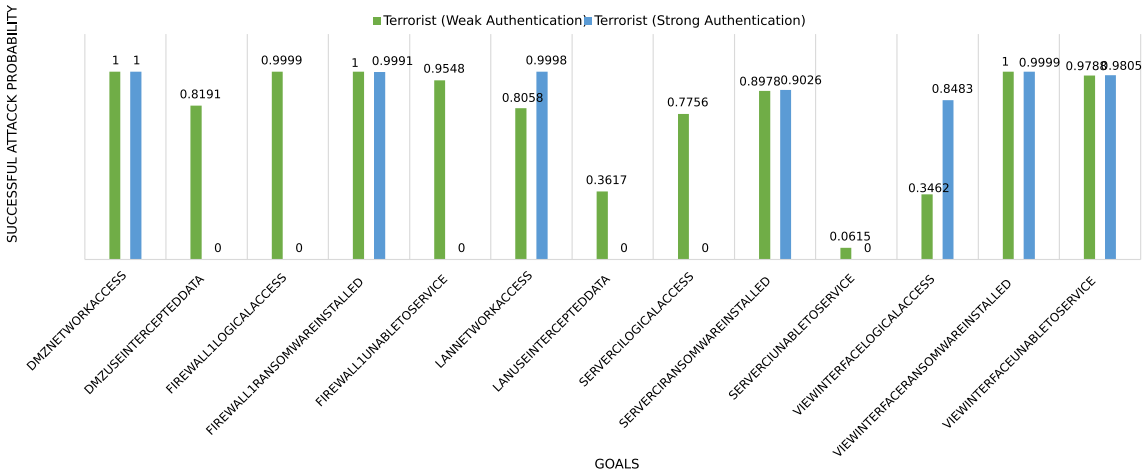

Fig. 8 Probability of successfully achieving goals for the Terrorist adversary as the user authentication level varies

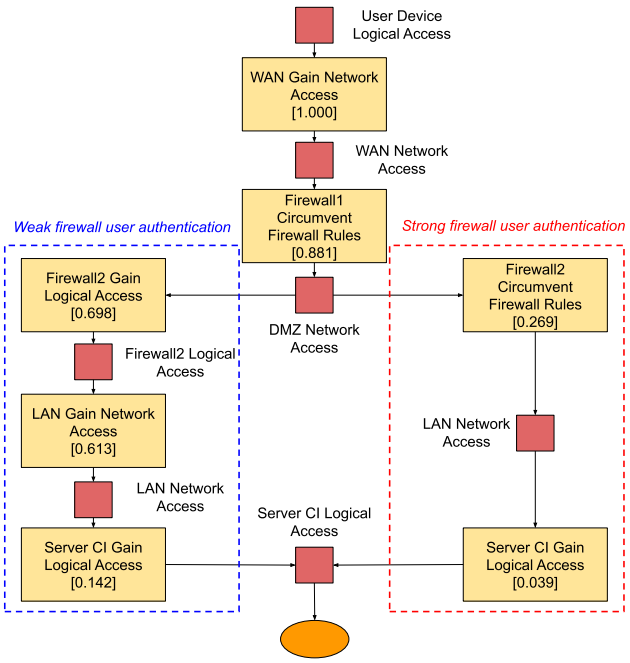

Fig. 9 Simplified ADVISE model for the Terrorist adversary with the *ServerCILogicalAccess* goal. Two different attack paths are taken by the adversary, depending on whether weak or strong firewall authentication is implemented in the system. For each attack step, its success probability is represented in square brackets

7.4 Analysis of most probable attack paths

As explained at the end of Sect. 2.3, in order to reach a goal, the adversary will follow the attack path that is more attractive depending on the adversary's skills and on the characteristics of the target system. It is thus possible to observe how the attack path followed by an adversary changes as some characteristics of the system vary.

In Fig. 9 a simplified representation (ADVISE style) of a combination of attack paths for the Terrorist adversary is shown. In this example the goal considered by the adversary is the *ServerCILogicalAccess* goal. Inside each yellow box representing an attack step, its success probability derived from the simulation is represented in square brackets. For this particular example, we computed the probability of completing the attack up to 24 time units (hours). Similarly to what have been done in Sect. 7.3, we ran two different simulations, one where the *strengthOfUserAuthentication* parameters of the two firewalls are set to "weak", and one where they are set to "strong".

It is possible to observe that, after the second attack step, two different attack paths are followed by the adversary. In the case of "weak" firewalls' authentication, the adversary decides to try three different attack steps: the adversary first tries to gain the logical access to the second firewall (i.e., she/he does not have the necessary preconditions), then to gain access to the LAN network, and finally to gain access to the Server CI. In the other case, when the firewalls' authentication is "strong", the adversary is not able to gain logical access to the second firewall, so she/he will try to circumvent the firewall rules. Even if the attack path is shorter (i.e., fewer attack steps) than the other, the success probabilities are significantly lower.

7.5 Analysis of the exposure level of system's components

On the basis of broad analyses such as the one shown in Sect. 7.1, it is also possible to obtain qualitative indications on the level of exposure of the system components. As we have already shown (e.g., in Sect. 7.1), we are able to compute the probability that an adversary can successfully reach a specific goal associated to a component. Now, we define the exposure level of a component as the average probability that an adversary successfully achieves all the goals associ-

ated to the component (i.e., she/he successfully attacks the component, thus achieving the associated goals), multiplied by 10. This average could also be weighted according to particular needs. Once an adversary is fixed, the exposure level of a certain component is, therefore, a score ranging from 0 to 10, which indicates how vulnerable the component is to a set of attacks carried out by the specified adversary, in the considered time window.

Figure 10 shows the exposure level of the SPaCe system components for the Vandal and Terrorist adversaries. The considered system configuration is the same used in Sect. 7.1, i.e., with the low authentication level. Note that the exposure level of components changes against different adversaries. In this case, for all the components, the exposure level against the Terrorist adversary is always equal to or higher than against the Vandal adversary. For the considered attacks and the target system, the Terrorist is, therefore, the most dangerous adversary. Besides Sensor1, which is the most exposed component for both adversaries, we have that (i) for the Terrorist, the second and third most exposed components are Firewall1 and DMZ, but, (ii) for the Vandal, the second and third most exposed components are instead View Interface, and Firewall1.

It is not surprising that these are among the most easily accessible components to adversaries: the sensors are available on board of vehicles, while the Firewall1 and the DMZ (and the View Interface) can be reached by users connected to the WAN. Paying more attention to these components is certainly a first step in strengthening system security because, on one side, the sensors represent the entry point for sensitive data coming from vehicles and, on the other side, the Firewall1 and the DMZ represent a first point of defense for the entire network infrastructure.

8 Related work

Security analysis has widely relied on models, especially qualitative models, as a means to organize the information on the system under analysis [3].

The survey in [12] proposes an extensive overview on attack and defense modeling techniques based on Directed Acyclic Graphs (DAGs). The authors analyze more than 30 formalisms and group them according to two main dimensions, which are (i) attack and/or defense modeling, where attack modeling focuses on attackers' actions while defense modeling focuses on defensive aspects, and (ii) static or sequential approaches, where sequential formalisms can model temporal aspects, while static approaches cannot. Static modeling of attacks includes, among others, attack trees, while sequential attack modeling includes Bayesian networks. Among the static formalisms that include defense aspects are security-activities graphs, while for sequential approaches for defenses include, e.g., attack-response trees. The authors also briefly illustrate a few formalisms which are not based on DAGs, like Petri nets and attack graphs.

Attack trees [16, 26] originated from adapting the idea behind fault trees to security analysis: basic attacks are combined in a tree-like structure, until reaching a top event, which represents a system-level security violation. Attack graphs allow for a more detailed modeling of the possible paths an adversary can follow, as they are not restricted to a tree structure. The ADVISE formalism [13] is a quantitative extension of attack graphs, in which the time required to perform attacks and their outcomes are determined by probability distributions. Further, ADVISE introduces specialized features to describe different attackers.

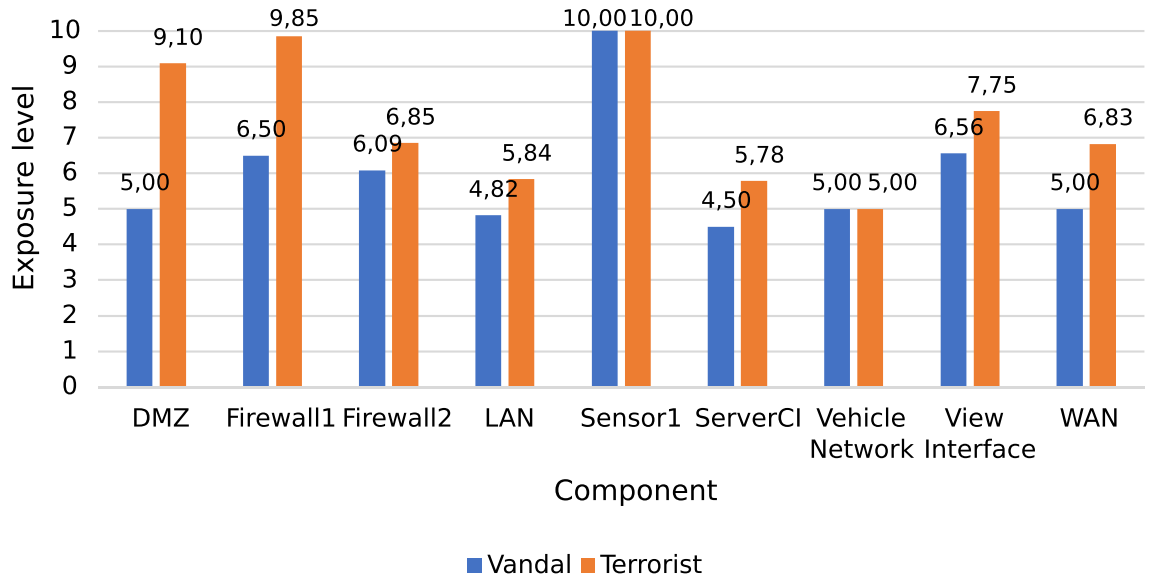

Fig. 10 Exposure level of SPaCe components for Vandal and Terrorist adversaries

The ADVISE Meta ontology framework [10] builds on the ADVISE formalism to increase its abstraction level. To the best of our knowledge, that is the only attempt in the literature to automatically generate detailed, stochastic security models from a collection of system meta-components and a concrete system configuration. For a more detailed discussion on the peculiarities of ADVISE Meta, and of its positioning with respect to other works sharing the same objective, we refer the readers to the original paper by Keefe et al. that introduces the framework [10].

In the following, we discuss the work in the literature from a perspective that is closer to ours. That is, we focus on work whose objective is to derive detailed security analysis models from higher level representations of the system, and we discuss the extent to which they consider a variety of adversaries' profiles and attack patterns.

The authors of [11] propose a tool that generates and simulates attack scenarios based on CAPEC. The input of the tool includes a detailed configuration of the network, information on the hosts, the profiles of the adversaries, and the CAPEC patterns. Their work shares some objectives with ours, most importantly the evaluation of possible attacks against with the inclusion of adversaries' profiles. However, the work in [11] is more tailored to advanced stages of system development, when details on the system are known; conversely, our approach can be applied since the early phases of the systems development life-cycle. Furthermore, we provide a more detailed characterization of adversaries, adopting the categorization defined in the TAL library.

The authors of [17] have analyzed and evaluated several existing conceptualizations on the topic of cyberthreat analysis, including different taxonomies, sharing standards, and ontologies. Like our work, they also covered both TAL and CAPEC. Their analysis concludes that no single taxonomy covers all the aspects and abstraction layers that are needed to perform an effective security analysis. We believe that such result confirms the need for cross-taxonomy mappings, like those we have presented in this paper. The work in [1] proposes a framework for assigning security scores to domain-specific Cyber-Physical Systems. The work also includes different attack types and adversaries' profiles. However, differently from our work, the attacks to the system are not simulated, but instead a scoring algorithm is used. Besides that, the approach adopts more general categories, both for attack types and for the capabilities of adversaries. In our work, we specify more detailed adversaries' profiles (in terms of accesses, knowledge, and skills), as well as more detailed attack patterns.

The authors of [8] propose a meta-language for modeling threats and simulating attacks. The approach is based on a textual meta-language that is used to specify domain-specific models, from which Java code for simulating the system is automatically derived. However, common attack patterns and adversaries' profiles are not included in the proposed meta-language. The work in [23] proposes extensions to UML for the specification and modeling of security aspects of critical infrastructures. Based on such specification, models for security analysis can be automatically generated. While UML is relatively widespread, building detailed UML models using customized profiles requires advanced modeling skills; conversely, the ADVISE Meta approach and our methodology focus on even higher abstraction and ease of use. Besides that, the work in [23] does not consider different types of adversaries' profiles.

To summarize, we believe our work is one of the first exploring the connections between: (i) quantitative security analysis formalisms at early-design stage, (ii) established taxonomies of adversaries' profiles (like the one defined in TAL), (iii) established taxonomies of attack patterns (like those defined in CAPEC).

In a very recent work [7], a computational environment based on the ADVISE formalism to model attack paths on CPSs has been developed, using a generalized stochastic optimization framework that allows to implement attacker agents based on different techniques, including approximate dynamic programming, reinforcement learning, or stochastic programming. While such work also builds on top of ADVISE, the focus is on its mathematical formulation on the evaluation algorithm. In the present work, we focus instead on how to build complex models and how to map them to real security taxonomies.

9 Conclusions and future work

In this paper, we proposed an approach for an early-stage security analysis and its application to a public transport supervision system. We focused on a meta-level modeling framework, called ADVISE Meta, which allows representing a system at a very high-level of abstraction, and then automatically deriving complex low-level stochastic models that represent possible attack steps and adversaries.

Our main objective was to enlarge the variety of the possible attack paths and adversaries considered in the analyses, and for this purpose, we extended the ADVISE Meta ontology integrating all adversaries described in the TAL and some representative CAPEC attacks. The paper provides a detailed discussion on the whole process for extending the ontology, which includes: the identification of the relationships between CAPEC, TAL, and ADVISE Meta elements, the definition of the methodology for integrating CAPEC attacks and TAL adversaries' profiles, and its application to show how specific attacks and profiles have been integrated.

In the second part of the paper, we made use of the extended framework for an early-stage security analysis of a public transport supervision system that has been developed

in the context of the SPaCe project. We provided a detailed view on the key aspects of the whole modeling process, showing how to define the architectural model of the system, the adversaries and their goals, and the targeted metrics.

We considered several security-oriented analyses, both targeting quantitative metrics, like the probability that a given adversary can successfully reach a particular goal, and qualitative metrics, like ranking the system's components based on their estimated exposure level. We compared different implementation solutions, we analyzed different system's scenarios at varying of the adversaries' profiles, and we analyzed the most probable attack path that can be followed by the adversary to reach the goal.

Ongoing work concerns the integration of defensive aspects in the modeling framework. Defensive strategies are currently embedded in the model, represented by the probability that an adversary succeeds in completing a given attack step, or by some specific property of components, for example the strength of the user authentication mechanisms. Our next objective is to explicitly integrate preventive and reactive defensive strategies in the framework, like the Moving Target Defence approaches, to capture the dynamic interplay between attackers and defenders. Finally, we are planning to use the proposed modeling extension in the domain of future cyber-physical ecosystems, as those addressed in the SERICS project EcoCyber (Risk management for future cyber-physical ecosystems [25]), where systems and services are characterized by increasingly interconnected and vulnerable digital components, aiming to understand how cyber threats can exploit the network environment.

Acknowledgements This work is a refinement and an extension of [15]. We would like to thank the PERFORM Group at the University of Illinois Urbana-Champaign, who developed ADVISE Meta and provided us with access to the framework. This work was partially supported by the Tuscany Region through the POR FESR Toscana 2014–2020 project SPaCe—Smart Passenger Center, and by the project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union—NextGenerationEU.

Author Contributions All the authors contributed equally to this work.

Funding Open access funding provided by Università degli Studi di Firenze within the CRUI-CARE Agreement. This work was partially supported by the Tuscany Region through the POR FESR Toscana 2014–2020 project SPaCe—Smart Passenger Center, and by the project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union—NextGenerationEU.

Availability of data and materials Not applicable.

Declarations

Conflict of interest None of the authors has competing interests related to the work described in the paper.

Ethical approval Not applicable.

References

1. Aigner A, Khelil A (2021) A security scoring framework to quantify security in cyber-physical systems. In: 2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS), pp 199–206. https://doi.org/10.1109/ICPS49255.2021.9468168
2. Avizienis A, Laprie JC, Randell B et al (2004) Basic concepts and taxonomy of dependable and secure computing. IEEE Trans Depend Secure Comput 1(1):11–33. https://doi.org/10.1109/TDSC.2004.2
3. Baadshaug ET, Erdogan G, Meland PH (2010) Security modeling and tool support advantages. In: 2010 International Conference on Availability, Reliability and Security, pp 537–542. https://doi.org/10.1109/ARES.2010.11
4. Casey T (2007) Threat agent library helps identify information security risks. Intel White Paper. https://doi.org/10.13140/RG.2.2.30094.46406
5. Courtney T, Gaonkar S, Keefe K, et al (2009) Möbius 2.3: an extensible tool for dependability, security, and performance evaluation of large and complex system models. In: 2009 IEEE/IFIP International Conference on Dependable Systems Networks, pp 353–358. https://doi.org/10.1109/DSN.2009.5270318
6. Ford MD, Keefe K, LeMay E, et al (2013) Implementing the ADVISE security modeling formalism in Möbius. In: 2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp 1–8. https://doi.org/10.1109/DSN.2013.6575362
7. Gonzalez SR, Betancourt Osorio J, Pardo González G, et al (2023) Modeling attacker behavior in cyber-physical-systems. In: Proceedings of the 11th Latin-American Symposium on Dependable Computing. Association for Computing Machinery, New York, NY, USA, LADC '22, p 117-124. https://doi.org/10.1145/3569902.3570188
8. Johnson P, Lagerström R, Ekstedt M (2018) A meta language for threat modeling and attack simulations. In: Proceedings of the 13th International Conference on Availability, Reliability and Security. Association for Computing Machinery, New York, NY, USA, ARES 2018. https://doi.org/10.1145/3230833.3232799
9. Kahani N, Bagherzadeh M, Cordy J et al (2019) Survey and classification of model transformation tools. Softw Syst Model 18. https://doi.org/10.1007/s10270-018-0665-6
10. Keefe K, Feddersen B, Rausch M, et al (2018) An ontology framework for generating discrete-event stochastic models. In: Remke A, Ballarini P, Barbot B, et al (eds) Computer Performance Engineering - 15th European Workshop, EPEW 2018, Proceedings. Springer, Germany, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pp 173–189, https://doi.org/10.1007/978-3-030-02227-3_12

11. Kotenko I, Doynikova E (2015) The CAPEC based generator of attack scenarios for network security evaluation. In: 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), pp 436–441. https://doi.org/10.1109/IDAACS.2015.7340774
12. Kordy B, Pièvre-Cambacédès L, Schweitzer P (2014) Dag-based attack and defense modeling: don't miss the forest for the attack trees. Comput Sci Rev 13–14:1–38. https://doi.org/10.1016/j.cosrev.2014.07.001
13. LeMay E, Ford MD, Keefe K, et al (2011) Model-based security metrics using ADversary VIew Security Evaluation (ADVISE). In: 2011 Eighth International Conference on Quantitative Evaluation of SysTems, pp 191–200. https://doi.org/10.1109/QEST.2011.34
14. Leone GR, Carboni A, Nardi S, et al (2022) Toward pervasive computer vision for intelligent transport system. In: 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), pp 26–29. https://doi.org/10.1109/PerComWorkshops53856.2022.9767376
15. Mariotti F, Tavanti M, Montecchi L, et al (2022) Extending a security ontology framework to model capec attack paths and tal adversary profiles. In: 2022 18th European Dependable Computing Conference (EDCC), pp 25–32. https://doi.org/10.1109/EDCC57035.2022.00016
16. Mauw S, Oostdijk M (2005) Foundations of attack trees. In: Proceedings of the 8th International Conference on Information Security and Cryptology. Springer-Verlag, Berlin, Heidelberg, ICISC'05, p 186-198. https://doi.org/10.1007/11734727_17
17. Mavroeidis V, Bromander S (2017) Cyber threat intelligence model: An evaluation of taxonomies, sharing standards, and ontologies within cyber threat intelligence. In: 2017 European Intelligence and Security Informatics Conference (EISIC). IEEE. https://doi.org/10.1109/eisic.2017.20
18. MITRE. Common Attack Pattern Enumeration and Classification. https://capec.mitre.org. (Accessed on April 14, 2023a)
19. MITRE. Common Weakness Enumeration. https://cwe.mitre.org (Accessed on April 14, 2023b)
20. OWASP. OWASP Top Ten. https://owasp.org/www-project-top-ten. (Accessed on April 14, 2023)
21. PERFORM Performability Engineering Research Group. ADVISE Meta Workshop 2016. https://www.mobius.illinois.edu/wiki/index.php/ADVISE_Meta_Workshop_2016 (Accessed on April 14, 2023a)
22. PERFORM Performability Engineering Research Group. Möbius Website. https://www.mobius.illinois.edu. (Accessed on April 14, 2023b)
23. Rodríguez RJ, Merseguer J, Bernardi S (2015) Modelling security of critical infrastructures: a survivability assessment. Comput J 58(10):2313–2327. https://doi.org/10.1093/comjnl/bxu096
24. Schmidt D (2006) Guest editor's introduction: model-driven engineering. Computer 39(2):25–31. https://doi.org/10.1109/MC.2006.58
25. SERICS. Risk management for future cyber-physical ecosystems (EcoCyber). https://serics.eu/en/services/spoke-8-gestione-rischio-governance/. (Accessed on April 14, 2023)
26. Ten CW, Liu CC, Manimaran G (2008) Vulnerability assessment of cybersecurity for scada systems. IEEE Trans Power Syst 23(4):1836–1846. https://doi.org/10.1109/TPWRS.2008.2002298