Extending a security ontology framework to model CAPEC attack paths and TAL adversary profiles

Francesco Mariotti*, Matteo Tavanti†, Leonardo Montecchi‡ and Paolo Lollini§

*†§Dipartimento di Matematica e Informatica 'U. Dini', University of Firenze — Firenze, Italy
‡Department of Computer Science, Norwegian University of Science and Technology — Trondheim, Norway
Email: *francesco.mariotti@unifi.it, †matteo.tavanti1@stud.unifi.it, ‡leonardo.montecchi@ntnu.no, §paolo.lollini@unifi.it

***Abstract*—Security evaluation can be performed using a variety of analysis methods, such as attack trees, attack graphs, threat propagation models, stochastic Petri nets, and so on. These methods analyze the effect of attacks on the system, and estimate security attributes from different perspectives. However, they require information from experts in the application domain for properly capturing the key elements of an attack scenario: i) the attack paths a system could be subject to, and ii) the different characteristics of the possible adversaries. For this reason, some recent works focused on the generation of low-level security models from a high-level description of the system, hiding the technical details from the modeler.**

In this paper we build on an existing ontology framework for security analysis, available in the ADVISE Meta tool, and we extend it in two directions: i) to cover the attack patterns available in the CAPEC database, a comprehensive dictionary of known patterns of attack, and ii) to capture all the adversaries' profiles as defined in the Threat Agent Library (TAL), a reference library for defining the characteristics of external and internal threat agents ranging from industrial spies to untrained employees. The proposed extension supports a richer combination of adversaries' profiles and attack paths, and provides guidance on how to further enrich the ontology based on taxonomies of attacks and adversaries.

***Index Terms*—adversary profile, ADVISE, attack path, CAPEC, modeling, ontology, security, TAL.**

I. Introduction

System security analysis is carried out by combining different approaches and technologies during all the development phases of the system. When performed at the very early stages of development (early design), models are typically used for an early assessment of the most critical architectural components that could be probable targets of cyber-physical attacks, providing preliminary indications on which components should require more attention. On the other hand, the challenge is to perform such analysis having only a preliminary knowledge of the system, without knowing the exact vulnerabilities of its components, which attacks could exploit them, which adversaries' profiles could carry out such attacks, and the possible consequences.

Due to the increasing complexity of systems, this type of activity can become very difficult and time-consuming. Tools have been developed in order to help with this complex task [1] [2]; however, even with their support, important decisions on which details and behaviors should be included in the system model must be made by human modelers. Still, most of these decisions depend on the kind of components and on their relations, and this is where Model-Driven Engineering frameworks [3] can play an important role, allowing analyzable low-level models to be derived from a high-level architectural description of the system.

In this work we analyze and extend a recent framework for security analysis, called ADVISE Meta [4], for modeling: (1) realistic attack patterns provided by the CAPEC (Common Attack Pattern Enumerations and Classifications) database [5], which is one of most complete collection of attack patterns, and (2) the different adversaries' profiles as defined in the Threat Agent Library (TAL) [6] by Intel, which provides an accurate characterization of different kinds of adversaries, and constitutes a reference work for describing the human agents that pose threats to IT systems.

We propose a methodology to guide the integration of CAPEC attack patterns and TAL adversaries' profiles to the ADVISE Meta ontology; the extended framework provides a richer combination of attack patterns and adversaries' profiles. Having a pool of common attacks can help modelers to understand if these attacks can be performed on the considered system, to combine them to represent more complex attacks, and to understand how the success of an attack depends on characteristics (resources, skill, intent, etc.) of the adversaries.

The rest of this paper is organized as follows. Section II provides some background information about CAPEC, TAL, ADVISE, and ADVISE Meta. In Section III we illustrate the methodology to map the CAPEC and TAL taxonomies into the properties of ADVISE Meta. In Section IV we extend the ADVISE Meta ontology to model TAL adversaries and some representative CAPEC attacks. As a proof-of-concept, in Section V we discuss the applicability of the proposed methodology in a simple example. Section VI describes some previous related work. Finally, in Section VII we conclude this paper.

II. Background

A. Attacks Reference Lists

Many successful attacks exploit old, well-known, vulnerabilities, on systems that are not sufficiently protected. At the same time, the complexity of today's computer systems makes it difficult for engineers to counter all existing security threats. For this reason, reference lists of known security

threats have been developed, e.g., the "OWASP Top 10" for web applications [7], or the MITRE Common Weakness Enumeration (CWE) [8].

While those lists focus on common weaknesses (i.e., "problems") that may exist in systems, the Common Attack Pattern Enumeration and Classification (CAPEC) [5], also provided by MITRE, is a catalog of common attack patterns to computer systems, i.e., descriptions of how weaknesses can be exploited.

CAPEC is a large online catalog of attack patterns (more than five hundreds entries), in which each attack pattern is mainly characterized by: a *Description*, providing a general description of the attack; a list of *Prerequisites* that are necessary for the attack; information on *Required Resources*, in terms of tools, devices and other resources needed to carry out the attack; information on *Required Skills*, in terms of specialized skills that an adversary must have to perform the attack; possible *Consequences*, in term of the scope and possible impact of the attack.

B. Threat Agent Library

Besides enumerating the possible attack patterns to a system, it is also important to understand what kind of adversaries the system might need to face with. The Threat Agent Library (TAL) by Intel [6] is a standardized library that provides a description of the human agents that can pose a threat to IT systems and related assets. One of the motivation for the development of TAL is to have a more precise description of possible security adversaries. In fact, ambiguous terms like "spy" or "hacker" are often used in the literature, while in reality capabilities and knowledge can differ substantially from an adversary to another.

TAL proposes the following eight different attributes to distinguish capabilities of adversaries (called "threat agents"): *intent*, *access*, *outcome*, *limits*, *resource*, *skill level*, *objective*, and *visibility*. Each attribute may assume certain pre-defined values: for example, the "intent" of an adversary can be "hostile" or "non-hostile", while the available "access" can be "internal" or "external". Combining different attributes' values, a total of twenty-one different human agents have been identified and characterized in the library (e.g., untrained employee, activist, government spy, and thief).

TAL can be used by risk managers to identify which human agents can threaten a system, and thus select the appropriate countermeasures.

C. ADVISE

The ADVISE formalism (ADversary VIew Security Evaluation) [2] [9] was introduced as a means to perform quantitative security analysis of complex systems, taking into account adversaries' profiles and preferences.

ADVISE allows modeling adversaries and attack steps, and then analyse, quantitatively, if a given adversary can achieve a certain goal, and the required effort. An ADVISE model is formed by two main components: the *Attack Execution Graph* (AEG) and the *Adversary Model*.

The AEG describes the actions that an adversary has to follow to reach a certain goal. The elements that can compose an AEG are: *skill*, *knowledge*, *access*, *attack step* and *goal*. The first three are items that can be held by adversaries at a certain point in time. They can be used as input for an attack step, to mean that they are the requirements that an adversary has to fulfill to execute that attack step. They can also be the results of an attack step, meaning in this case that the adversary can gain them as a result of the attack step. An attack step is a single step of an attack, which can have different outcomes (e.g., success or failure). Finally, a goal is an objective that the adversary wants to reach.

The Adversary Model describes the profile of an adversary, according to several attributes: *name*, *decision parameters* (planning horizon, attack preference weights), *skills*, *initial access*, *initial knowledge*, and *goals*. The values of these attributes determine whether a particular adversary can successfully reach a certain goal.

ADVISE is implemented as an atomic formalism in the Möbius framework [10] [11]. Once the models have been defined, Möbius allows specifying one or more measures of interest, typically related to the achievement of a certain goal or to the execution of a certain attack step. The ADVISE execution algorithm [2], based on Markov Decision Processes (MDPs), essentially consists in two steps that are repeated cyclically: i) selection of the optimal attack step to be attempted next, and ii) simulation of its outcome. Möbius can therefore simulate the behavior of the models at varying of certain parameters, to understand their impact on the measures of interest. For example, it can compute the probability that a given adversary can reach a certain goal at varying of a particular skill level.

D. ADVISE Meta Ontology

Building a low-level model (e.g., an ADVISE model) by hand can become a very difficult and time-consuming task, due to the increasing complexity of systems. It can be helpful to work at an higher abstraction level (i.e., at meta-level), and then derive low-level models automatically. This means, for example, modeling common patterns of attack and their relations with system components, so that this information can be reused in multiple, concrete, system models. Following this idea, the authors of ADVISE have proposed a meta-level modeling framework called ADVISE Meta [4], [12].

ADVISE Meta is an ontology framework that automatically generates detailed, discrete-event, stochastic models from high-level system design primitives. The approach adopted by ADVISE Meta is to describe the system using generic built-in blocks and relationships (defined by the ontology), which bring information on possible attacks in their definition.

The elements that compose the ADVISE Meta ontology are summarized in the following; those that are also present in the basic ADVISE formalism are marked with a star (*).

- **Component**: defines a base category of elements that can be part of a system. Examples: *Device*, *OperatingSystem*, *Network*.

- **Relationship**: defines a relation that can link a component to another one. A certain relationship only applies to specific kinds of components. Examples: *onNetwork*, *storageDevice*, *canDamage*.
- **Attribute**: represents a characteristic of a component. Such attributes can be used as parameters of the attack steps attached to the component. Examples: *dataIntegrityControl*, *mediaPortEnabled*, *userAuthenticationType*.
- **Access***: defines an access that an adversary may have at the start of an attack or gained during the attack. Examples: *InsiderAccess*, *LogicalAccess*, *PhysicalAccess*.
- **Skill***: defines a skill that an adversary may possess in varying degrees of proficiency. Examples: *BasicCyberOffense*, *Cryptanalysis*, *NetworkPenetration*.
- **Knowledge***: defines something that the adversaries may know beforehand, or that they may acquire during the attack. Only one knowledge (*FirewallRulesetKnowledge*) is defined in the base ADVISE Meta ontology.
- **Other State Variable**: can be used to define state variables related to system components; typically, these are also adversary attack goals. Examples: *Damaged*, *Disabled*, *MalwareInstalledOn*.
- **Attack Step***: defines a step of an attack that can be performed by an adversary. Examples: *PhysicalDisable*, *GainUserCredentials*, *ModifyDataLocally*.
- **Adversary***: defines an adversary's profile with several characteristics. The main difference with the base ADVISE framework is that here built-in adversary templates are provided. Examples: *ForeignGovernment*, *HackerGroup*, *OrganizedCrime*.
- **Metric**: only one metric (*goalAchieved*) is defined in the base ontology, but other metrics can be added.

To create a concrete model of a system, based on an existing ADVISE Meta ontology, the first step is to add the components that are part of the system into the System Instance Diagram (SID) and set the corresponding attributes. Then, components are linked to each other with the available relationships.

Once the model has been defined, the low-level models (i.e., ADVISE models) can be generated starting from a particular configuration of the system, which consists in one system diagram, one adversary, and a subset of available metrics. The generated ADVISE models are usually complex and difficult to understand: without this approach they would have required a lot of time-consuming and error prone manual effort. Moreover, ADVISE Meta can automatically generate several additional elements (i.e., performance variable reward models, set studies, and simulators) that can be used to perform analyses of the generated ADVISE model.

III. Methodology: linking CAPEC, TAL and ADVISE Meta elements

From the overview provided in Section II-D, we can note that the current version of the ADVISE Meta ontology does not include all the elements needed to represent CAPEC attacks and TAL profiles. The attack steps provided by the framework are classified into a few categories (e.g., Gain access, Damage or disable, Malware) that are not sufficient to cover the vastness of the CAPEC database. Moreover, the adversaries templates of the ontology are generic (e.g., Hacker Group) while those provided by the TAL library are more specialized (e.g., Thief, Vandal, Anarchist).

In this section we propose a methodology to align the ontology with these elements. The methodology is based on identifying the relationships between the properties of the different frameworks that are involved, that is, ADVISE Meta, the CAPEC database, and TAL. Such mapping is actually carried out in three separate steps, as described in the following.

1) Identifying the relationships between TAL and CAPEC, to understand how the information available in the CAPEC sections can be mapped into TAL attributes. This mapping is completely independent of the underlying modeling framework.
2) Identifying the relationships between TAL and ADVISE Meta elements, to represent TAL adversaries' profiles in the ADVISE Meta framework.
3) Identifying additional information in CAPEC sections that can be related to ADVISE Meta elements, to represent additional aspects (e.g., architectural requirements) when modeling CAPEC attacks.

A. From CAPEC sections to TAL attributes

First of all, we highlight the connections between TAL attributes and CAPEC elements, which allow describing CAPEC attack patterns in terms of the properties of TAL adversaries that are required to execute them (see Table I, upper part).

- The TAL *Intent* attribute defines whether the adversary intends to cause harm, and it can take values "Hostile" or "Non-Hostile". This information can usually be deduced from the "Description" section of a CAPEC entry.
- The *Access* attribute in TAL denotes the extent of the adversary's access to the system's assets and it may have values "Internal" or "External". The "Prerequisites" section in CAPEC lists all the prerequisites that an adversary must fulfill in order to perform the attack, including access to assets. Sometimes this section is not detailed enough, so additional information must be extracted from the "Description" section.
- *Limits* in TAL defines the legal and ethical limits that may constrain the adversary, and the extent to which the adversary may be prepared to break the law. It can assume four different values ("Code of Conduct", "Legal", "Extra-Legal Minor" and "Extra-Legal Major"). In CAPEC there is not a dedicated section for legal and ethical limits, so this information must be deduced from the "Description" section.
- The TAL *Resource* attribute defines the organization level of the adversary and so the resources available to run the attack. It can have six different values ("Individual", "Club", "Contest", "Team", "Organization", and "Government"). Such information can be partially found in the section "Resources Required", where software tools,

TABLE I
TRANSFORMATION FROM CAPEC SECTIONS TO TAL ATTRIBUTES, AND FROM TAL ATTRIBUTES TO ADVISE META ELEMENTS

		TAL Attributes						
		Intent	**Access**	**Limits**	**Resources**	**Skill Level**	**Objective**	**Visibility**
CAPEC Section	**Description**	●	●	●	●		●	●
	Prerequisites		○			●		
	Resources Required				○			
	Skills Required					○		
	Consequences						○	
ADVISE Meta Element	**Knowledge**	*Intent*						
	Access		*(InsiderAccess)*					
	Skill			*Limits*	*Resources*	*SkillLevel*		*Visibility*
	Goal						Attack dependent	

● Usually present, information might not be complete and must be deduced — ○ Might be present, information is usually complete

devices, and other resources required for the attack are listed. However, this section is not always present in the CAPEC entries, so in these cases the "Prerequisites" section must also be checked.

- The *Skill Level* in TAL determines the special training or expertise an adversary typically possesses. It can take four different values ("None", "Minimal", "Operational" and "Adept"). CAPEC contains a dedicated section called "Skills Required", where an indication of the skill level required to perform the attack is given. Otherwise, if this section is not present in the entry, the "Prerequisites" section must be analyzed.
- The *Objective* in TAL defines the action that the agent intends to take in order to achieve a desired outcome. It can have five different values ("Copy", "Destroy", "Injure", "Take" and "Don't Care"). This information can be found in the "Consequences" section in CAPEC, where the scope (e.g., confidentiality, integrity, or availability), and the impact of the attack are described. This section is not always present, so also in this case, it might be required to also check the "Description" section.
- *Visibility* is the extent to which the agent intends to conceal or reveal his or her identity. It can take four different values ("Overt", "Covert", "Clandestine" and "Don't Care"). CAPEC does not include a dedicated section for such information; however, from the "Description" section one can often deduce to what extent it is important for the adversary not to be detected.

The mapping of an attack pattern is not always straightforward, as not all the required elements are always present in the CAPEC database. The "Description" and the "Prerequisites" sections are always present in a CAPEC entry, but other useful sections like "Skills Required" and "Resources Required" might be missing. Therefore, adding a CAPEC attack to the ontology requires a certain degree of interpretation of the CAPEC entry.

B. From TAL attributes to ADVISE Meta elements

To represent TAL adversaries in the ADVISE Meta framework, a mapping that highlights the relationships between them is needed. We have mapped ADVISE Meta elements to the corresponding TAL attributes as follows (see Table I, lower part):

- The TAL *Intent* property is mapped to the *Knowledge* concept of ADVISE Meta, because having an hostile intent is something known to adversaries. A malicious intent is necessary to perform a malicious attack, for this reason a new Knowledge element named *Intent* has been added to the ontology.
- The *Access* concept from TAL has already its corresponding element in ADVISE Meta, namely the Access element (ADVISE concept) called *InsiderAccess*.
- In TAL, *Limits* represents the legal and ethical limits of the adversary. We mapped this concept into a Skill in ADVISE Meta, which can be interpreted as the ability of the adversary to operate at different levels of legality. To represent this aspect in ADVISE Meta, a new Skill element named *Limits* has been added to the ontology.
- The *Resources* concept in TAL can also be mapped into a Skill, which can be seen as the ability of the adversary in obtaining resources. For this property we have added to the ontology a new Skill element named *Resources*.
- Adversaries in the TAL have a *Skill* level, which represents a general indication of their level of expertise. We have mapped TAL Skill Level to the equivalent element in ADVISE Meta, that is, *Skill*. Thus, we have added to the ontology the new Skill element *SkillLevel* associated to each adversary.
- The *Objective* attribute of TAL does not have a corresponding element in ADVISE Meta. The *Objective* cannot be interpreted as a Skill, like we have done in the other cases. Conceptually, the closest element in the ADVISE formalism is the *Goal*, which is however associated to attack steps. This means that this element must be added to the ontology when attack steps are modeled,

TABLE II
ASSIGNMENT OF NUMERICAL VALUES TO TAL ATTRIBUTES

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
	Don't Care	0

and not when modeling the adversaries' profiles.

- The TAL *Visibility* attribute is the degree of importance for the adversary to remain invisible. Conceptually it is something known to the adversary, but since it can assume more than two values it cannot be mapped into a Knowledge, which is instead a Boolean property (either the adversary has it or not). Lacking an equivalent concept in ADVISE Meta elements, we have added a new Skill element called *Visibility* that represents the extent to which the adversary intends to conceal or reveal his or her identity.

We have not mapped any ADVISE Meta element to the TAL element *Outcome*. The reason for this decision is that a TAL adversary can have multiple outcomes, so it is impossible to model such situation by using a single ADVISE Meta element associated to the adversary's profile. Furthermore, the possible outcomes (e.g., "Copy" or "Destroy") are strictly related to the kind of attack, and not only to the adversary's profile.

In the ADVISE description of an adversary, a Skill level is defined as an integer value in the range between 0 and 1000. Therefore, for each TAL attribute that was mapped to a Skill, threshold values are defined to represent the different TAL attribute values (see Table II). These intervals can be changed according to the need of the modeler or be parameterized with global variables.

C. From CAPEC sections to ADVISE Meta elements

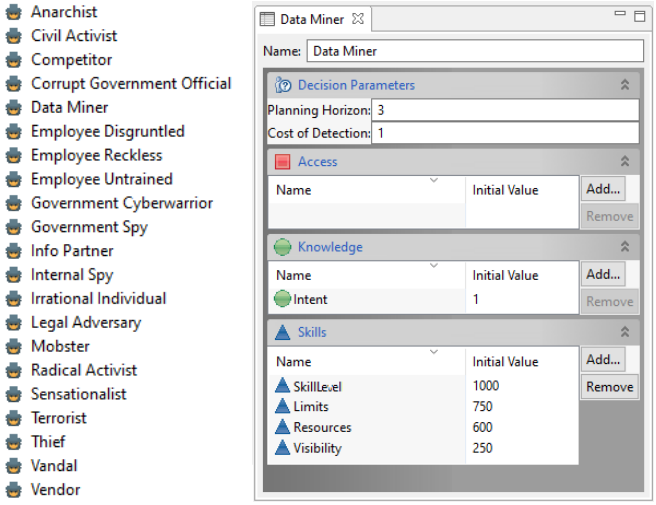
When modeling CAPEC attacks in ADVISE Meta, additional information should be derived from CAPEC sections. In particular, when adding new attack steps, the target of the attack (i.e., the affected component of the system) should be specified. This type of information can be deduced from the "Description" section of the CAPEC entry. Further information, like attack preconditions related to the presence of specific architectural components (e.g., the presence of a communication network for a flooding attack), should be retrieved from the "Description" and "Precondition" sections.

IV. EXTENSION OF ADVISE META

We have used the methodology illustrated in Section III to extend the ADVISE Meta ontology for representing the adversaries' profiles provided by TAL and some representative CAPEC attack patterns.

A. Extension of ADVISE Meta with TAL adversaries' profiles

We have added to the ontology all the twenty-one adversaries' profiles that compose the TAL library (Figure 1-a). We translated the TAL adversaries' profiles to ADVISE Meta adversaries, based on the TAL/ADVISE Meta mapping described in Section III-B, and on the values for attributes specified in the TAL library.



(a) (b)

Fig. 1. List of the TAL adversaries' profiles added to the ontology, visualized in ADVISE Meta framework (a) and TAL "Data Miner" adversary's profile visualized in the ADVISE Meta framework (b).

As an example, we show here the application of the methodology to the *Data Miner* profile. In TAL this adversary has "External" Access, "Hostile" Intent, "Adept" Skill Level, "Extra-Legal Minor" Limits, "Team" Resources and "Clandestine" Visibility. We have created a new ADVISE Meta adversary called "Data Miner" and we have associated to it the attributes that we have previously added to the ontology when developing the methodology (Section III-B).

According to the values assignment proposed by the methodology (Table II), the Data Miner profile is defined by the following attributes (see also Figure 1-b):

- *InsiderAccess* with value 0 ("External"), meaning that the Access is not added to the profile;
- *Intent* with value 1 ("Hostile");
- *SkillLevel* with value 1000 ("Adept");
- *Limits* with value 750 ("Extra-Legal Minor");

- *Resources* with value 600 ("Team");
- *Visibility* with value 250 ("Clandestine").

The same has been done for all the twenty-one adversaries of the TAL library. Following the proposed methodology, other adversaries can be added to the ontology with minimal effort.

B. Extension of ADVISE Meta with CAPEC attacks

Another objective of this work is to allow the representation of CAPEC attacks in the ADVISE Meta framework, which requires the addition of some attack steps to the ontology of the framework. The abstraction level of these attacks should be as high as possible (i.e., attacks should be independent from low-level details of the system) to guarantee reusability. To add a CAPEC attack to the ontology, the CAPEC description must be translated to an implementation in ADVISE Meta. For this purpose, the TAL/CAPEC mapping (Section III-A) and the TAL/ADVISE Meta mapping (Section III-B) should be used jointly, effectively using TAL attributes as a bridge.

In this paper we have focused on some representative attack patterns, to show the applicability of our methodology. We have selected attack patterns that cover all the main attributes that characterize security, i.e., Confidentiality, Integrity and Availability [13]. In particular, we have focused on the following five CAPEC attacks:

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

In the following, we give a detailed description of the application of the methodology to one of the five attacks that we have added to the ontology.

C. CAPEC-94: Adversary in the Middle

Also known as Man in the Middle, this attack consists in the adversary targeting the communication between two nodes of a network (e.g., a client and a server), in order to retrieve information or modify data. The adversary takes position in the middle of the communication (i.e., behaving like one of the two end nodes), so that when a node wants to communicate with the other, the adversary receives the data and can potentially read or modify them before forwarding them to the other node. In CAPEC this attack is classified under the "Software" and "Communication" domains, and under the "Subvert Access Control" attack mechanism. Targets of this attacks are system components classified as Device; this information has been extracted from the "Description" section of the CAPEC entry.

We have modeled this attack by adding three different attack steps to the ontology, adapted from the "Execution Flow" section of the CAPEC entry. The first step, called *Determine Communication Mechanism* (Figure 2-a), is where the adversary identifies the communication mechanism used by the two nodes. Preconditions for this attack are "Extra-Legal Minor" Limits (i.e., with value of at least 750), "Hostile" Intent (i.e., Intent element is present), and "Team" Resources (i.e., with value of at least 600). This information has been derived 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). These preconditions are specified by C++ code in the Preconditions Expression (Figure 2-b).

If the attack is successfully completed, the adversary gains the *CommunicationAccess* Access, which is a precondition for the second attack step. In the second attack step, called *Position In Between Targets*, the adversary takes position inside the network to intercept the messages between the two nodes. This attack step has the same preconditions of the previous one, with the addition of the *CommunicationAccess* Access. If this step is successfully executed, the *MonitoredNetworkAccess* Access is gained by the adversary.

In the last attack step, *Monitoring Network Access*, the adversary reads or modifies the intercepted data (thus gaining the Access *UseInterceptedData*). The preconditions for this attack are the *MonitoredNetworkAccess* Access and "Extra-Legal Major" Limits. For the previous attack steps no major law breaches were required, while a major illegal action is required for to carry out this final step.

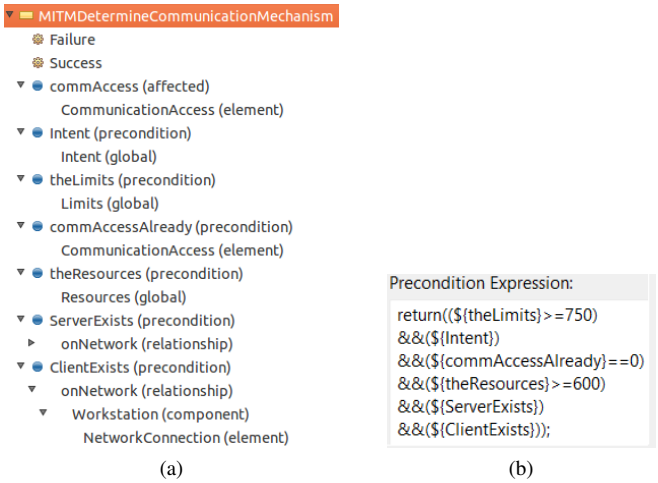


Fig. 2. The "Determine Communication Mechanism" attack step, which is part of the "Adversary in the Middle" attack.

V. Application Example

In this section we illustrate an application example to show the usability of the extension proposed in Section IV. Figure 3-a shows the SID model of a simple networked system in ADVISE Meta; the model consists of three elements taken from the base ontology: a *Server*, a *Workstation* and a *WirelessNetwork*. The Server and the Workstation are linked to the Wireless Network through the *onNetwork* relationship.

From the high-level SID model in ADVISE Meta notation, a concrete ADVISE model (Figure 3-b) is generated by the

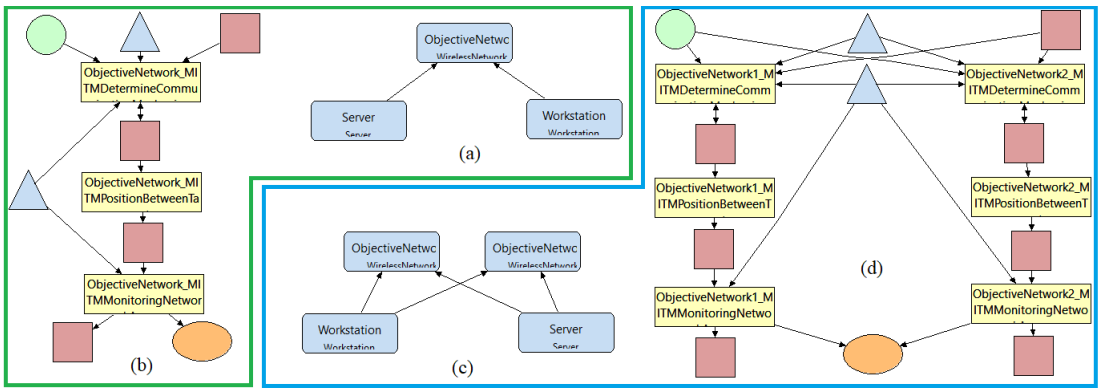


Fig. 3. Architectural model of the 'Man in the Middle' attack (a) along with the generated ADVISE model (b). In (c), the modified architecture with an additional network, and the new generated ADVISE model (d).

tool. The attack steps in the generated model are derived from attacks specified in the meta-elements in the ontology (e.g., *WirelessNetwork*), and the elements in the SID instance model. In the example, the following attack steps are generated: *MITMPositionBetweenTargets*, *MITMDetermineCommunicationMechanism* and *MITMMonitoringNetworkAccess*. Limits and Resources Skills (the two blue triangles) and Intent Knowledge (green circle) are preconditions for the first attack step along with the Access to the network (red square). The other Access elements *CommunicationAccess*, *MonitoredNetworkAccess* and *UseInterceptedData* are obtained by an adversary after successfully completing the first, second and third attack steps, respectively.

We simulated the ADVISE model using all the TAL adversaries added to the ontology to figure out which of them are able to successfully execute the attack. Figure 4 shows the mean number of attack steps successfully performed at varying of the TAL adversaries. Most of the adversaries are not capable to successfully complete the attack (many of them are not even able to carry out the first attack step), while only four adversaries reach the goal. This type of analysis can be used as an early-stage support to identify the most dangerous adversaries for a given system and to guide the adoption of appropriate countermeasures.

Let's now assume that the SID model is modified by adding a redundant network channel between the server and the workstation, as depicted in Figure 3-c. As expected, the new ADVISE model (Figure 3-d) contains two different attack paths automatically generated from the information in the new system model (with two networks) and the information in the ontology (individual attack steps).

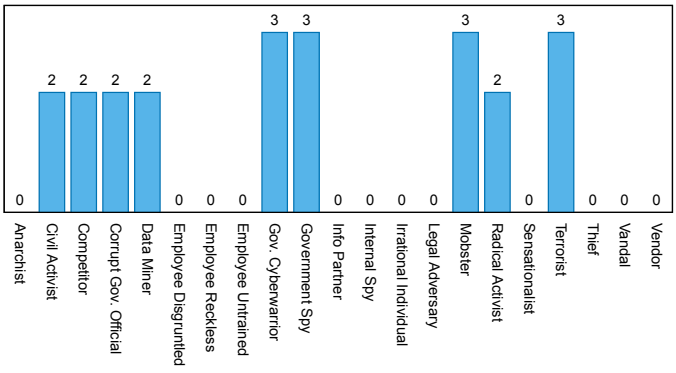


Fig. 4. Mean number of attack steps successfully performed by TAL adversaries for the 'Man in the Middle' attack.

VI. Related Work

Modeling, especially qualitative models, have been widely used in security analysis [1]. Attack trees [14], [15] are an adaptation of fault trees to security analysis: following a similar principle, basic attacks are combined to reach a top event, representing the compromise of the system. Attack graphs allow for a more detailed modeling of possible paths an adversary can follow. As explained in [2], ADVISE is a quantitative extension of attack graphs, with further specialized features.

To the best of our knowledge, the ADVISE Meta ontology framework [4] is the only available attempt to automatically generate detailed, stochastic security models from an ontological definition of system components and a concrete system configuration. The related work presented in [4] discusses the peculiarities of the ontology framework and its relations with other works sharing the same objective; we refer the reader to that work for the positioning of ADVISE Meta with respect to existing literature. In this section we discuss the related works where detailed security analysis models are derived from higher-level representations of the system, focusing on the extent to which they consider a variety of adversaries' profiles and attack patterns.

In [16] the authors propose a software tool for the automatic generation and simulation of attack scenarios based on CAPEC. The tool takes as input the detailed network configuration, the hosts information, the adversaries' profiles and the CAPEC patterns. Even if their work shares some of our objectives (i.e., the evaluation of possible attacks with the inclusion of adversaries' profiles), they focus more on an advanced stage of system development life-cycle where the system details are available, while the extended ADVISE ontology can be applied at the early phases of systems development life-cycle. Moreover, adversaries' profiles are just defined with a range of skill levels (low, medium, high), while we

formalize the profiles' descriptions adopting the more general definitions and categories defined in the TAL library.

The authors of [17] analyze and evaluate several existing taxonomies, sharing standards, and onthologies on the topic of cyber threat analysis, including both TAL and CAPEC. They conclude that there is not yet an existing taxonomy covering all the aspects and abstraction layers that are needed for an effective security analysis. Such result confirms the need for cross-taxonomy mappings, like the ones presented in this paper. The authors of [18] propose a framework for scoring security in domain-specific Cyber Physical Systems including attack types and adversaries' profiles. It is meant to be used at a later stage of the system development life-cycle, and it adopts general attack categories instead of specific attack patterns. Also, the adversaries' profiles are defined just by a generic list of capabilities, while we specify more details of adversaries' profiles in terms of accesses, knowledge, and skills.

In [19] a meta language for modeling threats and simulating attacks is proposed. Domain-specific models can be specified by using a textual meta language that automatically generates Java code for simulations, but common attack patterns and adversaries' profiles are not included in the meta language. In [20] the authors generate models for security analysis starting from a formal UML definition of the system. This approach requires advanced skills in order to build detailed UML models, while the ADVISE Meta approach and our methodology focus on high abstraction and ease of use. Moreover, this work do not consider different types of adversaries' profiles.

In summary, we could not find any work that enables a security analysis at early-design stage from a high-level system description, which explicitly considers and integrates a comprehensive library of adversaries' profiles (like the one defined in TAL) and how attack patterns (like those defined in CAPEC) are related to such profiles.

VII. Conclusions

In this paper we have presented an extension of the ADVISE Meta ontology to enable a broader set of security analyses, whose distinguishing feature is to integrate the standardized Threat Agent Library for describing the adversaries' profiles and attack patterns defined in the CAPEC repository. To accomplish this, we have proposed a methodology where the relationships between TAL, CAPEC and ADVISE Meta elements are highlighted, constituting a guideline for integrating new adversaries' profiles or attack patterns. By following the proposed methodology we have added all the twenty-one adversaries' profiles that are part of TAL and some representative CAPEC attacks.

Ongoing work concerns the application of the methodology for the security analysis of the system under development in the context of the SPaCe project, where coaches of a train are equipped with multi-medial solutions to orchestrate surveillance and mobility services.

Acknowledgment

We would like to thank the PERFORM Group at the University of Illinois who developed ADVISE Meta and provided us the access to the framework. This work was supported in part by the Tuscany Region through the POR FESR Toscana 2014-2020 project SPaCe – Smart Passenger Center.

References

[1] E. T. Baadshaug, G. Erdogan, and P. H. Meland, "Security modeling and tool support advantages," in *2010 International Conference on Availability, Reliability and Security*, 2010, pp. 537–542.

[2] E. LeMay, M. D. Ford, K. Keefe, W. H. Sanders, and C. Muehrcke, "Model-based security metrics using ADversary VIew Security Evaluation (ADVISE)," in *2011 Eighth International Conference on Quantitative Evaluation of SysTems*, 2011, pp. 191–200.

[3] D. Schmidt, "Guest editor's introduction: Model-driven engineering," *Computer*, vol. 39, no. 2, pp. 25–31, 2006.

[4] K. Keefe, B. Feddersen, M. Rausch, R. Wright, and W. H. Sanders, "An ontology framework for generating discrete-event stochastic models," in *Computer Performance Engineering*. Cham: Springer International Publishing, 2018, pp. 173–189.

[5] MITRE, "Common Attack Pattern Enumeration and Classification." [Online]. Available: https://capec.mitre.org

[6] T. Casey, "Threat Agent Library helps identify information security risks," *Intel White Paper*, 2007.

[7] OWASP, "OWASP Top Ten." [Online]. Available: https://owasp.org/www-project-top-ten

[8] MITRE, "Common Weakness Enumeration." [Online]. Available: https://cwe.mitre.org

[9] M. D. Ford, K. Keefe, E. LeMay, W. H. Sanders, and C. Muehrcke, "Implementing the ADVISE security modeling formalism in Möbius," in *2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2013, pp. 1–8.

[10] T. Courtney, S. Gaonkar, K. Keefe, E. W. D. Rozier, and W. H. Sanders, "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*, 2009, pp. 353–358.

[11] PERFORM Performability Engineering Research Group, "Möbius Website." [Online]. Available: https://www.mobius.illinois.edu/

[12] ——, "ADVISE Meta Workshop 2016." [Online]. Available: https://www.mobius.illinois.edu/wiki/index.php/ADVISE_Meta_Workshop_2016

[13] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11–33, 2004.

[14] C.-W. Ten, C.-C. Liu, and M. Govindarasu, "Vulnerability assessment of cybersecurity for scada systems using attack trees," in *2007 IEEE Power Engineering Society General Meeting*, 2007, pp. 1–8.

[15] S. Mauw and M. Oostdijk, "Foundations of attack trees," in *Information Security and Cryptology - ICISC 2005*, D. H. Won and S. Kim, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 186–198.

[16] I. Kotenko and E. Doynikova, "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)*, vol. 1, 2015, pp. 436–441.

[17] V. Mavroeidis and S. Bromander, "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, sep 2017.

[18] A. Aigner and A. Khelil, "A security scoring framework to quantify security in cyber-physical systems," in *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, 2021, pp. 199–206.

[19] P. Johnson, R. Lagerström, and M. Ekstedt, "A meta language for threat modeling and attack simulations," in *Proceedings of the 13th International Conference on Availability, Reliability and Security*, ser. ARES 2018. New York, NY, USA: Association for Computing Machinery, 2018.

[20] R. J. Rodríguez, J. Merseguer, and S. Bernardi, "Modelling security of critical infrastructures: A survivability assessment," *The Computer Journal*, vol. 58, no. 10, pp. 2313–2327, 2015.