Abstract

This paper proposes a metamodel for analyzing security aspects of enterprise architecture by combining analysis of cybersecurity with analysis of interoperability and availability. The metamodel extends an existing attack graph based metamodel for cybersecurity modeling and evaluation, P²CySeMoL, and incorporates several new elements and evaluation rules. The approach improves security analysis by combining two ways of evaluating reachability: one which considers ordinary user activity and another, which considers technically advanced techniques for penetration and attack. It is thus permitting to evaluate security in interoperability terms by revealing attack possibilities of legitimate users. Combined with data import from various sources, like an enterprise architecture data repository, the instantiations of the proposed metamodel allow for a more holistic overview of the threats to the architecture than the previous version. Additional granularity is added to the analysis with the reachability need concept and by enabling the consideration of unavailable and unreliable systems.

1. Introduction

Kordy et al. [1] find generalizing and unifying as one of the trends in their survey of attack and defense models. Metamodels that cover several enterprise architecture domains can have multiple benefits. First, a metamodel with a wide domain coverage is able to express more aspects of the modeled situation, thus enabling better understanding. Secondly, creating models from different metamodels manually can be costly time and effort wise. With a metamodel that has a broad domain coverage, the modeler can model a situation once, but reuse the model to analyze various aspects of the architecture according to the needs. Therefore, adding aspects of enterprise architecture to an enterprise security metamodel can be beneficial.

Enterprise and its assets are not only of interest to outside attackers. In some cases a legitimate user might develop bad intentions and become a threat to the enterprise. There are a multitude of reasons why this might happen, for example frustration, or just personal gain. Being already inside the organizational perimeter, this type of user has access to some systems and services. By having an overview of the systems that this kind of access exists to, we can evaluate how secure the architecture is against the assumed generic insider with the motivation to attack, but lacking penetration testing skills. An interoperability analysis is able show which parts of the architecture that kind of user would be capable of reaching. By interoperability we mean the satisfaction of a set of communication needs between two or more actors in a given architecture.

In certain conditions, the availability of assets impacts the level of security in an enterprise. While owners of IT assets generally desire them to be available and there is even a broad class of attacks targeting availability (denial-of-service attacks), the unavailability of an asset can preempt some attacks through not allowing them to proceed where they could do harm. Examples of this are switch-off schedules of wireless hotspots, or timed filtering of certain ports on firewalls. Such unavailability, however, makes IT assets unreachable and can thus distort interoperability across the enterprise architecture. Hence, the aspects of availability have also implications for interoperability in enterprise architecture, although temporarily, and are therefore beneficial to consider.

The aim of this paper is to propose an extension to an existing attack graph analysis solution, P²CySeMoL [2], both to widen its scope and to improve its accuracy. These goals are achieved by adding capabilities for analyzing aspects of interoperability and availability. The improvements presented in this paper allow to analyze the compromising potential of unauthorized activity of attackers without penetration testing skills (i.e., non-hackers), such as legitimate users inside a company. The ability to model communication intent and differentiate attacker capabilities presents an improvement over P²CySeMoL evaluation, which only considers a single type of attacker - a professional penetration tester.
The paper is organized as follows. The second section of the paper presents the three metamodels, which are being combined within this study. The third section explains how the metamodels are combined and the fourth section provides an example of a corresponding instance model.

2. Related work

The solution outlined in this paper combines three existing metamodels, which are based on a formalism for probabilistic, predictive simulation, and a tool that implements the formalism. Below, all these are introduced and a brief overview of alternative approaches is provided.

2.1. Framework for probabilistic simulation and its implementation

P²AMF [3] is a framework for probabilistic sampling and Monte Carlo simulation, which uses the Object Constraint Language (OCL) [4], a formal language for representing computational expressions on models in the Unified Modeling Language (UML) [5]. Moreover, P²AMF extends OCL with the ability to express uncertainties in attribute values and model structure. The attributes can be stochastic and expressed through probability distributions. In the model structure, the existence of objects and their connections can be uncertain. The Enterprise Architecture Analysis Tool (EAAT) [3] is a software tool that implements P²AMF. The tool can model two types of probabilistic models - metamodels (also called class models) and instance models (also called object models). A metamodel contains a representation of a certain modeling domain, represented as UML class diagrams, and the corresponding evaluation logic, implemented in OCL. P²CySeMoL is an example of such a metamodel. Instance models are created with regards to a given metamodel, and serve to represent a concrete architecture such as a current or prospective scenario of an organization’s enterprise architecture. Instance models can be created automatically from external data that originate for example from network or vulnerability scanners. That model generation functionality is based on the XSLT model transformation standard.

2.2. Metamodel for cybersecurity prediction

P²CySeMoL [2], an extension of the Cyber Security Modeling Language (CySeMoL) [6], is an attack graph based probabilistic metamodel for analysis and prediction of cybersecurity of enterprise architecture. P²CySeMoL predicts the probability with which a single attacker or multiple attackers can compromise different parts of the architecture, based on its structure and the properties of its parts (e.g., the attributes of the systems the architecture consists of). The metamodel assumes that the attacker is a professional penetration tester having access to any publicly available tools that support performing cyber attacks. Furthermore, the user can specify how much time, in days, the attacker can use to conduct an attack. To create an instance model of a real or a hypothetical IT architecture, the user has to model a number of elements (assets), each with its set of properties (defense mechanisms), and connect the elements to each other. Examples of the elements to model are network zones and network interfaces (e.g., network gateways), application services and application clients, web applications, operating systems, communication protocols and data flows, data stores, people, password accounts, password authentication mechanisms, physical and social zones, and even administration aspects such as network management processes and security awareness programs. Finally, having an attacker node added into the model and connected to an element such as a network zone, P²CySeMoL is ready to simulate different possible attacks and produce a map of probabilities, with which the attacker can compromise the different assets of the enterprise architecture (e.g., application servers, operating systems or data flows) in various ways (e.g., disruption of availability, unauthorized modification of data, or obtainment of access under administrator rights). In other words, P²CySeMoL evaluates the cyber attack reachability of an attacker in a given enterprise architecture.

P²CySeMoL consists of four basic concepts: attacker, asset, attack step, and defense mechanism. An attacker is the threat agent from whom an attack can propagate. An asset is a broad category including much of the above mentioned (e.g., operating systems, application clients and servers, data flows, protocols, etc.). An attack step is an action, through which an attack takes place and propagates (e.g., connect to, disrupt or obtain product information). A defense mechanism is a property that influences the probability of different attacks steps being reached from preceding ones by an attacker (e.g., cryptographic authentication, password repository hashed or public source code). For each asset, a number of attack steps and defense mechanisms are usually defined.

In P²CySeMoL, the attacks throughout the architecture represented by the instance model, consist of and proceed through attack steps, the probability of reaching which is calculated using logic implemented in OCL in the metamodel. As applies to P²AMF in general, the OCL expressions use object-relational
queries across the model structure to obtain information about the existence of model elements and the values of their attributes such as specific defense mechanisms, or attack steps. Normally, a single evaluation simulates a large number of samples (e.g., 1000), within each of which an attack step can either be reached or not. To produce the final map of probabilities, the Boolean values from the single samples (1 for reached attack step and 0 otherwise), are averaged into a final probability of the modeled attacker reaching that attack step in the model. This is done for all attack steps for all assets in the modeled enterprise architecture.

2.3. Metamodel for interoperability prediction

Ullberg et al.’s [7] metamodel for interoperability prediction evaluates the satisfaction of a set of communication needs, each between two or more actors, in a given architecture. There can be a multitude of communication needs in an architecture model, and in the evaluation of each simulation sample, each communication need either is satisfied or not. Having the interoperability evaluation use a number of samples, the satisfaction of each communication need becomes a percentual extent reflecting the average of the outcomes from the individual samples.

In the metamodel, an actor can take a variety of forms such as a computer system, an application component, human, or a whole enterprise. Actors share the capability to actively use information, operate on it, interpret it, and transform it. Actors however seldom communicate directly. The communication rather propagates through media of different kind, such as a data communication network, circuit-switched telephone line, postal mail, or the air between two people talking to each other. For this, the metamodel defines a concept named message passing system, which enables the propagation of actors’ messages and the addressing among the actors. The metamodel additionally defines the concept language, through which both actors and message passing systems communicate with each other. Furthermore, the metamodel defines two hierarchies among languages. First, one language can be a sub-language of another, i.e., the first language can express a subset of the semantics that the second language can express, such as an SQL query can only express a subset of what a free text in English can. Second, one language can be used to carry (i.e., transport) a communication in another language from one node to another, such as IP datagrams carry the higher-level communication of an SSH session over a computer network. Other aspects taken into account in the interoperability analysis are language translations, on a more granular level also constructs of which each language consists, addresses, which themselves are constructs and are necessary for a communication to take place, construct translations, and conversations, through which a communication takes place. Finally, the metamodel defines the concepts communication need and conversation communication need, which indicate the needs of communication between actors, and the satisfaction of whose is evaluated during the simulation.

2.4. Metamodel for analysis of availability of enterprise IT services

Franke et al.’s [8] metamodel for availability prediction builds on ArchiMate [9]. It extends a subset of ArchiMate entities with the attributes and calculations relevant for availability analysis. The metamodel uses fault tree structures to allow the analysis of aggregated availability of systems. The method to calculate fault trees is explained in [10].

3. The integrated metamodel

In order to address the issues of interoperability and availability as defined in this paper, in the context of security analysis, we propose a metamodel that combines all three aspects.

As P²CySeMoL, the new metamodel is based on attack graph logic, taking into account a series of attack steps and context-dependent possibilities. An attacker traverses through attack steps of assets to proceed with an attack. The attack steps are traversable, while defenses are simply attributes holding some value for the evaluation of how the attack can proceed through the attack steps. The combined metamodel incorporates several new elements (assets), attack steps and rules of how the attack steps can be traversed. It also allows the user to select whether to model the attacker as a skilled, professional penetration tester, or a user who lacks penetration testing skills. The newly introduced features are explained below.
Our integrated metamodel mainly builds on the \( P^2 \)CySeMoL metamodel, but extends the taxonomy and rules of \( P^2 \)CySeMoL. The aim of the combined metamodel is to allow reasoning about attacks where the attacker is known to possess certain communication capabilities and other access prerequisites (e.g., access credentials), but is not assumed to be a security professional. A stereotypical company employee (e.g., acting as an insider attacker) would fall into this category. A possibility to define arbitrary availability dependencies is added, and evaluated in a fault tree manner, as in Franke et al.’s metamodel. Consequently, the combined metamodel can take into account broader consequences of the unavailability of some asset (intentional or unintentional), which is not captured in the metamodel in a generic sense. The new metamodel is shown in Figure 1 with additions highlighted in black. The elements are grouped into four categories of assets, attack steps, defenses, and the rest. An element with three dots denotes that there are more elements of this type available, but have not been included in the Figure to keep it understandable. Figure 2 shows the taxonomy of the new metamodel. The taxonomy includes a full list of assets together with their attack steps and defenses. The extensions to the metamodel are explained in the following sections.

### 3.1. Interoperability

The main feature introduced by integrating the logic of Ullberg et al.’s interoperability metamodel, is the ability of the combined metamodel to distinguish between two different types of attackers. It allows to specify whether an attacker is assumed to be a security professional (i.e., a professional penetration tester) or a security layman who lacks penetration testing skills. Considering the former case, all the rules in the combined metamodel are taken into account. Conversely, considering the latter case restricts the set of traversable attack steps to only those that follow the interoperability evaluation logic, i.e., those that do not require penetration testing skills. In other words, if a security professional possesses access credentials to a target system, reaching it takes next to no time for the person, without employing penetration testing techniques, which in this case equally applies to the layman. If the attacker, however, does not possess such access credentials, the security professional might succeed with a certain probability, given a certain time frame for attacking, unlike the layman.

The central concepts for the interoperability extension are language, language translation and
actors. All three are of type asset, having attack steps and/or defense mechanisms defined, as in P²CySeMoL. Language is defined as a method of communication. In order for two actors to communicate in a direct manner, both need to speak the same Language. LanguageTranslation is the process of enabling communication between two actors who speak different languages. Language translation must be conducted by an additional actor, or multiple actors, in terms of mediating such communication. If two actors that need to communicate don’t speak the same language, their communication needs to be translated. For example, a human user is familiar with using the graphical user interface of an e-mail client, and so can read and write e-mails. The actor on the other side of the communication channel, the e-mail server, receives communication in a different form (language), such as IMAP, SMTP or Exchange. Hence, there must be an actor which translates between the user’s usage, and the computer usage of the e-mail server - the e-mail
client. One could further complicate this by adding a remote desktop connection, or a HTTP interface of the e-mail client, and a web browser.

Actors use languages to speak with each other on certain topics. By actors we mean assets that are capable of interaction and in $P^{3}$CySeMoL asset taxonomy these are ApplicationClient and ApplicationServer. The asset WebApplication is defined as a possible extension to ApplicationServer and thus only extends its functionality and cannot exist on its own. The metamodel includes the two hierarchies among languages (carrying language and sublanguage), as in Ullberg et al’s metamodel for interoperability analysis [7] (explained earlier in the text). Actor also communicates through Dataflow, which denotes an information flow between the client and one or more servers.

In this combined metamodel, communication on a certain topic (conceptually also including the usage of a system) is seen as an attack step of an actor. Hence, communication understood this way is seen as an activity with compromising potential. The attack steps that represent communication on a certain topic (or usage of a system in a certain way) are named CommunicateOnTopic, ExpressTopic, ProduceRequestOnTopic, ProduceResponseOnTopic, Speak, Use.

3.2. Reachability need

The aim of adding the concept of reachability need to the metamodel is to overcome an ambiguity in the original $P^{3}$CySeMoL. If there were two or more attackers, the $P^{3}$CySeMoL map, agnostic to individual attackers, would not display which of the attackers reached what steps, nor whether a single step was reached by one or more attackers. Reachability need eliminates this problem. More precisely, reachability need shows the probability of a single attacker or every attacker from a set of multiple attackers reaching one or more attack steps on one or more assets in the enterprise architecture. Reachability need, named ReachabilityNeed in Figure 1, is an abstraction of communication need and a hypothetical “attack need”.

3.3. Availability

Availability is part of some of the assets in the combined metamodel. The availability logic is based on the fault tree structure that was introduced in section 2.4. Availability is seen as a defense and is denoted as isAvailable. The name isAvailable is used instead of unavailable, so that availability values of assets can be set as evidence to instance models instead of inverse values. The assets that include the defense isAvailable are ApplicationClient, ApplicationServer, NetworkZone and OperatingSystem. The defense isAvailable influences both the possibility of an attacker to compromise any of the named assets, as well as the interoperability aspect of getting access to the asset. It influences the likelihood of traversing through any attack step of a particular asset it is part of. Availability is evaluated for both types of attackers that can be used in the analysis, a security professional and a layman. An example of availability analysis is modeling the probability of an application server breaking down. The result of the application server unavailability is that all the related systems (e.g., operating systems and/or other software) would become unreachable (i.e., unavailable). Given such a situation however, all the systems becoming unavailable for use could also render them unavailable for propagating attacks. In such a probabilistic analysis, the probability of an asset’s availability influences the likelihood of that asset becoming compromised by an attack step.

4. Example

Let us imagine a hypothetical situation with 3 attacks, which is presented in Figure 3 and Figure 4. Figure 3 presents a high level view of the attacks, while Figure 4 shows the analysis result from EAAT modeling. The different variations of red in Figure 4 represent the likelihood of an attack step being successful with up to over 90%. The color legend can be found in [2].
There is an attacker Craig who wants to speak about a certain conversation topic, represented by a language (WindowsOSGUI). Craig wants to be able to access ApplicationServer Win8_RemoteDesktop-Service. This is represented by a link between the Attacker and Language (Figure 4). To reach his target he needs first access to the network. Like Figure 4 shows, Craig has legitimate access to a Mac computer that has a Win8_RemoteDesktopClient installed. In P³CySeMoL, we can represent this situation by linking Craig to a PasswordAccount (it looks like Craig guessed the credentials, but they were actually provided by some administrator) and to the computer's AccessControlPoint. This gives access to the computer’s operating system, which in turn gives access to the Windows 8 Remote Desktop client (Win8_RemoteDesktopClient). Now, if Craig was a security professional, he could choose to try to hack into the ApplicationServer through NetworkInterface, NetworkZone, and Dataflow. This choice is shown as attack path 1 in Figure 3. If Craig was however a security layman with legitimate access, he could take attack path 2 instead.

In terms of interoperability analysis, if communication on a topic is successful, then the Attacker will be able to communicate with an asset on the desired topic. Since the application client is capable of speaking the very language that Craig wants to converse in (WindowsOSGUI), ApplicationClient communication is successful, i.e. Craig has satisfied the communication need with respect to the ApplicationClient. However, we are interested in exploring additional communication needs. To speak with the ApplicationServer, Craig will, as Figure 4 indicates, need to use a LanguageTranslation, since the ApplicationServer only speaks RDP. The provision of such a translator is, in fact, precisely one of the main features of Win8_RemoteDesktopClient, and since Craig has access to it, he can use that translation. Furthermore, Craig will need credentials to bypass the AccessControlPoint on the ApplicationServer. Then, a Dataflow between the client and server needs to be established. This dataflow is dependent on a network connection, which is provided by the NetworkZone (a specialized kind of message passing system). The NetworkZone speaks TCP/IP, which is able to transport the RDP language.

The example contains another attacker, Steve, who has a similar goal to reach Win8 RemoteDesktop-Service, represented as attack path 3 in Figure 3. Steve is, however, a security professional and according to our definition, can therefore employ hacking.
techniques. According to P²CySeMoL logic, he gets access to a password account by social engineering Wendy in a SocialZone and reaching the ApplicationServer that way. What is interesting here is the ReachabilityNeed measurement. By connecting a ReachabilityNeed to a specific attacker and asset, we can measure if the connected attacker is successful in its attack. Without reachability analysis the model might show us that the ApplicationServer gets compromised, but we could not evaluate the success of specific hackers in an easy manner.

Another aspect to consider in the example is the existence of availability. In Figure 3 the following assets are influenced by availability analysis: ApplicationClient Win8_RDPClient, OperatingSystem MacOS, ApplicationServer Win8_RDPService, and OperatingSystem WindowsServer8. The probability of availability of these systems influences the likelihood that these systems get compromised either by a security professional or a layman.

5. Discussion

The proposed metamodel is based on P²CySeMoL taxonomy. The taxonomy is an abstraction of a security structure of a typical organization. It includes a limited number of assets, attack steps and defenses, and thus limits the modeling and analysis to a certain granularity. By extending the security metamodel, the analysis capability of the metamodel was improved. The interoperability aspect introduced to the metamodel, considers the threat from legitimate users which was previously missing. Legitimate users with harmful intentions, sometimes referred to as inside attackers, might be motivated by personal gain, frustration or revenge, among other things. An attack by such a user could end up being as devastating as one by a professional attacker.

Using the combined metamodel might save modeler’s time, when compared to using three separate metamodels to model the same architecture. For example, if the initial goal is to only analyze interoperability, then the modeler can use the same
model later with minimal changes to analyze security. In this case there is an obvious benefit of modeling once, but being able to analyze several aspects of the architecture.

6. Conclusion and future work

A method to broaden the scope of security analysis of enterprise architecture was proposed. The approach extends an existing attack graph based metamodel P²CySeMol with the analysis of interoperability and availability. The analysis methodology has been implemented as a metamodel using a probabilistic analysis tool EAAT. An example is provided to demonstrate how the combined metamodel can be used for security evaluation.

The authors see several benefits with the proposed metamodel. The first benefit is that the metamodel is now capable of analyzing scenarios where a user might have legitimate access to some parts of the IT architecture. Such is the case with employees, for example. Combined with importing data from other sources, like an enterprise architecture data repository or network scanners, the instantiations of the proposed metamodel provide a more holistic overview of the looming threats. Another benefit is that the metamodel allows more granular analysis where availability issues might affect the actual attack paths and choices of an attacker. The concept of reachability need is another form of granularity improvement. Namely, in case of several attackers, P²CySeMol is not able to show which attack step was achieved by which attacker. With reachability need this ambiguity is removed.

There are several tasks for future work. The first task is related to improving the interoperability analysis of the combined metamodel by adding the interoperability concepts of addressing and conversations. For example ApplicationClient and ApplicationServer would need a common address for the communication between them in certain languages like TCP/IP to be successful. Another task is to validate the proposed metamodel with an empirical study.

7. Acknowledgement

This study has been financed by SweGrids, the Swedish Centre for Smart Grids and Energy Storage (www.swegrids.se); thereunder Fortum and the Swedish Energy Agency.

8. References