Insider Threat Assessment: a Model-Based Methodology

Nicola Nostro, Andrea Ceccarelli, Andrea Bondavalli
University of Firenze, Viale Morgagni 65, Firenze, Italy
{nicola.nostro, andrea.ceccarelli, bondavalli}@unifi.it

Francesco Brancati
Resiltech S.r.l.
Piazza Nilde Iotti 25,
Pontedera (Pisa), Italy
francesco.brancati@resiltech.com

ABSTRACT

Security is a major challenge for today's companies, especially ICT ones which manage large scale cyber-critical systems. Amongst the multitude of attacks and threats to which a system is potentially exposed, there are insider attackers i.e., users with legitimate access which abuse or misuse of their power, thus leading to unexpected security violation (e.g., acquire and disseminate sensitive information). These attacks are very difficult to detect and mitigate due to the nature of the attackers, which often are company's employees motivated by socio-economical reasons, and to the fact that attackers operate within their granted restrictions. It is a consequence that insider attackers constitute an actual threat for ICT organizations. In this paper we present our methodology, together with the application of existing supporting libraries and tools from the state-of-the-art, for insider threats assessment and mitigation. The ultimate objective is to define the motivations and the target of an insider, investigate the likeliness and severity of potential violations, and finally identify appropriate countermeasures. The methodology also includes a maintenance phase during which the assessment can be updated to reflect system changes. As case study, we apply our methodology to the crisis management system Secure!, which includes different kinds of users and consequently is potentially exposed to a large set of insider threats.

Categories and Subject Descriptors

C.2.0 [**General**]: Security and protection K.6.5: [**Computers and Education**]: Security and Protection: *authentication*, *unauthorized access* K.6.m: [**Computers and Education**]: Miscellaneous: *security*

General Terms

security, standardization, verification.

Keywords

security; insider threats; risk assessment; attack path.

1. INTRODUCTION

Today's ICT organizations are constantly facing the challenge of ensuring high degrees of security and privacy. Security measures are attentively selected and maintained, mainly with the intent of protecting the organization from external threats. Several tools and solutions are available for this purpose, for example firewalls. A lesser amount of solutions is instead available for mitigating threats coming from within the company, that is, from its own employees; these threats, that we refer to as *insider threats*, are most often mitigated almost exclusively through regulations and policies [6]. For example, insiders to an organization such as former, or newly fired employees or system administrators might abuse their privileges to conduct masquerading, data harvesting, or simply sabotage attacks. Although some intrusion detection systems offer insider threats capability, it is still very difficult to characterize all the threats, transform them into rules (or, in case of anomaly-based intrusion detection, instruct the detector to identify them as anomalies), and effectively detect intruders.

The problem of insider threats have been, and currently is, largely discussed in literature, because it is particularly challenging to identify insiders and mitigate the possible threats they pose to a system. In fact it should be considered that an insider may have socio-economical roots, and the detection of false positive in insider attacks may have severe consequences on an organization (e.g., due to the impact of false accusations of insider threats on both the individual and the organization [7]). Mitigation may be composed of prevention including deterrents as strict regulatory aspects, surveillance, legal implications, or detection methods and procedures that may help protecting the system.

It appears evident that protecting from insider threats requires to study the socio-economical profiles of the users, the assets they use, their actions, and the impact of the actions on the assets, systems and organization. This calls for a tailored *insider threats assessment* activity, which takes into account socio-economical aspects while identifying the attacks, their impact on the system and organization, and possible countermeasures.

We aim to tackle this problem proposing in this paper a methodology for insider threats assessment and mitigation. The methodology presents the following features: i) it is tailored for the challenges posed by insider threats, ii) although it benefits of the support of attack libraries and tools for system and attack modeling, it does not impose restrictions on the characteristic of the libraries and tools to use, iii) it takes into account socio-economical aspects, including a description of the profile of the attacker, iv) relies on model-based formalisms of the system and of the attack paths to analyze threats and evaluate countermeasures. The methodology first defines the system requirements and the attackers profiles, then identifies the threats, the attack paths and the potential countermeasures. The methodology also includes a maintenance phase during which

This work is based on an earlier work: "A methodology and supporting techniques for the quantitative assessment of insider threats", in the Proceedings of the 2nd International Workshop on Dependability Issues in Cloud Computing, (DISCCO'13, September 30 2013, Braga, Portugal), Copyright ACM, 2013. http://dx.doi.org/10.1145/2506155.2506158

reconfiguration facilities of the system are supported to update the assessment.

This paper extends our previous work [23], which reported a preliminary description of our approach for insider threats assessment and a simple example. In this paper, we extend [23] mainly presenting a conclusive, largely rewritten description of the methodology, and applying it to a case study in which different kinds of users of a crisis management system are analyzed.

The rest of the paper is organized as follows. In Section 2 we clarify our notion of insider threat. In Section 3, we survey the state of the art on insider threats assessment. In Section 4, we describe our methodology, and in Section 5 we apply the methodology to our case study. Finally, conclusions are reported in Section 6.

2. DEFINITION OF INSIDER THREAT

Talking of insider is often confusing. In literature there are different definitions of insiders, each of which, in general, has a negative meaning of the concept of insider. Although the question *"who is an insider?"* seems simple, a well established definition is still missing. Definition in [10], [11] suggests that an insider must be defined with respect to a set of rules that is part of a security policy:

"A trusted entity that is given the power to violate one or more rules in a given security policy... the insider threat occurs when a trusted entity abuses that power."

The CERT Program goes further by providing a definition of *malicious insider* [9]:

A malicious insider is defined as a current or former employee, contractor, or business partner who meets the following criteria: i) has or had authorized access to an organization's network, system, or data; ii) has intentionally exceeded or intentionally used that access in a manner that negatively affected the confidentiality, integrity, or availability of the organization's information or information systems.

In literature several uses of term *insider* can be found [34]. In general, we can simply define an insider as *an entity that has been given the privileges to act within a specific environment.* What is of interest is the use of privileges (being it an abuse or misuse of privilege, or simply a mistake) in such a way that it constitutes a threat (being it malicious or accidental [7]) i.e., an *insider* threat.

3. STATE OF THE ART AND ADVANCEMENTS

A wide literature exists on the issue of insider threats, although most of it is devoted to the description of techniques and methods useful to *prevent* insider attacks and to protect the system or the infrastructure. Examples are initiatives as restrict remote access and system administrator access, or inhibit the use of removable drives. Also works have been done to *predict* insiders activities, for example [14] presents a prediction model based on graph theory approaches, where alarms are raised when it is detected an increasing risk that users actions might lead to compromise systems resources.

Recent years have witnessed an increasing number of works devoted to research solutions for (early) *detection* of insider attacks [9], [7]. However, to the best of our knowledge, most of the proposed works have been devised for specific and well defined case studies, sometimes also requiring to introduce simplifying assumptions, and without offering good portability to different systems or environments.

In [1], the authors aim to detect masquerade attacks, where a user impersonates another user, by profiling user behaviors. To evaluate these attacks, UNIX command data were collected from a certain number of users and then the data were contaminated with masqueraders. The experiment was performed and compared with six masquerade detection techniques: Bayes one-step Markov, Hybrid multi-step Markov, IPAM, Uniqueness, Sequence-Match and Compression.

In [2] an approach is pursued to predict financial fraud from insiders by considering: i) the audit data, gathered during the operation of the system, and ii) the human factor, as a qualitative aspects to integrate with the classic quantitative analysis of financial transactions.

The works [3] and [4] focuses on the analysis of anomalous commands executed on data bases. In [3] a user profile has been generated according to a syntax-centric approach, that represents the structure of the SQL queries submitted by the users, in order to detect anomalous queries. Another approach which consider the a relational database management system as case study is in [4], where the approach is data-centric unlike the previous one. In this approach the anomalies are identified looking at the data that are retrieved following the user's request.

In [5] it is proposed a graph-based model of the basic network connectivity and access control mechanisms of the system, to identify system vulnerabilities which can be exploited by a malicious insider. In [6], a study within an enterprise was conducted, where analysts were monitored while performing analytical operations. Using Grounded Theory research method (Grounded Theory is defined as the discovery of theory from data systematically obtained from social research), the objectives are: i) to understand how security event analysis works, and ii) to create an event-based model to analyze insider threats. This model shows how alerts and events created can be analyzed to determine if malicious insider behavior is present. This approach defines a process and a model to identify and characterize insider threats.

While great efforts have been devoted for the research of solutions to the thorny problem of the insider threats, few efforts have been made to identify and delineate a methodology for the *insider threats assessment*. Threats assessment processes, not specific for insider threats, exist; for example we mention the guide elaborated by NIST [12], as we believe it is a representative and comprehensive example of a threats assessment process. Although generic and largely applicable, it does not propose methodologies and libraries to support the assessment process, and most important it does not differentiate or specialize for *insider threats*. In fact, in our view the assessment process for insider threats requires to shift the focus from the architectural view of the system (being it a logical or physical view, depending on which one is available at the time threats assessment is performed) to a *user-centric view*, where socio-economical aspects, motivations and permitted actions of the users are at the very foundation of the analysis and are bounded to functional requirements. The attack goals and the vulnerabilities exploitable to achieve them come from the definition of the user and of its permitted actions described in the functional requirements, with little knowledge on the system architecture.

Although compliant to the steps of the NIST process, our approach focuses on insider threats assessment, defining a methodology supported by libraries, techniques and tools. Note that libraries as the threat models from [5], the attacks list from [27], the security vulnerabilities and exposure dictionary from [28], if considered useful, can be easily applied in the appropriate step of the methodologies. However we believe they mostly result out of scope because they are specific for external threats and discuss system or software vulnerabilities rather than exploitable privileges in well-functioning and apparently secure services. At present, we are aware of *no public databases or libraries* specific for insider threats analysis.

Summarizing, we propose to advance the state of the art by:

- showing a methodology for the insider threats assessment, valuable both during system design, and maintenance activities;
- starting the analysis from socio-economical aspects and user motivations, rather than from the system architecture;
- presenting an example of threat libraries, specifically built for the case study but easily reusable in different contexts; and
- showing the possible usage of templates, libraries, and tools as well as formal or semi-formal languages to structure the methodology application and favour its reuse and update through the system lifecycle.

4. OUR METHODOLOGY IN SIX STEPS

The objective of this work is to provide an incisive, clear and supported methodology to handle systems, and related risk assessment processes, in which the insider threat issue could be relevant. Especially the proposed methodology will be independent from the type of system.

In the following Sections 4.1 to 4.6 we present our methodology, pointing out the description of six key iterative phases around which the methodology has been developed. Compliance with NIST's risk assessment procedure is not shown for brevity, but can be easily proven intuitive. The specialization of the proposed methodology on a real case study will be given in Section 5.

4.1 System under analysis

A *system* is characterized by a number of resources (e.g., services, computers, removable drives, etc.), one or more communication networks, and users, which can use the system or in general interact with it. In addition, new features of the system can be integrated over time, due to the evolution of technologies, and the update of system specifications.

When characterizing a system, we can identify macro-components which constitute the overall architecture. A system can thus be seen as a set of n macro-components. A description of the possible actions of the users in the system should be provided and used here, being it a description of requirements in natural language or supported by semi-formal or formal languages. Although not mandatory, a semi-formal or formal description is very important to guide the analysis and also to support possible future updates of the analysis in case of system modifications.

Thus, given that precise requirements and a model of the system is preferable (especially as input for the successive phases of our methodology), the methodology does not provide restrictions on the description method and level of detail. For example, we are aware that in practise a threats assessment may incur while the system requirements definition is still ongoing and thus only description in natural language is available (note that this may call for additional iterations of the procedure once a semi-formal description is available).

In our previous and preliminary paper [23], we provided an informal description of the system through natural language, which is source of potential ambiguity. Anyway, providing a formal description of the overall system, in particular if complex, may be somewhat expensive in terms of time. For this reason we consider convenient and sufficient to have a semi-formal description limited to the aspects of interest of the system (services used) and the interactions that users i.e., potential insiders, may have with it. Through a semi-formal notation, it is possible to immediately understand the description of the system using graphical notations along with natural language descriptions, thus reducing ambiguous interpretations.

There exists several notations that can be found adequate and used: Data-Flow Diagrams, Finite State Machines, Activity Diagrams, etc. In Section 5 we propose the usage of the Use Case Diagrams: they allow to describe the system functionalities and use case scenarios, from the point of view of the users/insiders.

Templates to help the system characterization are proposed in the case study in Section 5 where use case diagrams show the interaction between the offered functionalities and the actors involved. The description of each diagram is explored in a table containing the relevant information.

4.2 Insiders

The second phase of the methodology consists in profiling potential insiders, determining their dangerousness to the system and their key attributes. This is organized in two steps.

In the first step, all possible users involved in the system under analysis are identified.

In the second step their potential attributes as insiders are defined. To support this task, we define a library of insiders. We refer to the attributes presented in [8], where a detailed threat agent library (TAL) has been provided, which constitute a consistent reference library describing the human agents involved in IT systems and that could pose threats to such kind of systems, although not limited to insider threats. The idea is *not* to represent specific individuals, but instead the library is intended to create a taxonomy of specific attributes useful to uniquely identify the users/insiders.

Specifically, according to the threat agent library [8], eight attributes are described in the following and specialized for insider threats:

Intent. Whether the insider intends to cause harm. Insiders fall into two categories based on their intent: *Hostile* and *Non-Hostile*.

Access. Defines the extent of the insider's access to the company's assets. There are two options: *Internal* or *External*.

Outcome. Defines the insider's primary goal, that is, what the insider tries to accomplish with a typical attack. Possible outcomes are: i) *Acquisition/Theft* of essential assets or sensitive data; ii) *Business Advantage* to acquire business processes or assets; iii) *Damage* to injury physical or digital assets, or intellectual and industrial property; iv) *Embarrassment* to generate loss of credibility, influence, and competitiveness; v)

Technical Advantage to acquire production processes or assets rather than a business process.

Limits. Legal and ethical limits that may constrain the insider, e.g., the extent to which the insider may be prepared to break the law. Options are: i) *Code of Conduct*, when the insiders follow both the applicable laws and an additional code of conduct accepted within a profession or an exchange of goods or services; ii) *Legal*, where the insiders act within the limits of applicable laws; iii) *Extra-legal, minor*, when the insiders may break the law in relatively minor, non-violent ways, such as minor vandalism or trespass, e.g., activist; iv) *Extra-legal, major*, when insiders take no account of the law and may engage in felonious behavior resulting in significant financial impact or extreme violence, e.g., members of organized crime.

Resource. Defines the organizational level at which an insider typically works, which in turn determines the resources available to that insider for use in an attack. Options are: i) *Individual*: insider acts independently; ii) *Club*: members interact on a social and volunteer basis; iii) *Contest*: participants interact together for a very short period of time and perhaps in anonymous way with the objective to achieve a single goal; iv) *Team*: a well organized group with a leader, typically motivated by a specific goal and organized around that goal; v) *Organization*: a larger and better resourced than a *Team*; vi) *Government*: controls public assets and functions within a jurisdiction, it is very resourced and persists long term.

Skill Level. The special training or expertise an insider typically possesses. Options are: i) *None*, when the attacker has no expertise in the methods useful to attack the system, but has the ability to perform random acts of disruption or destruction; ii) *Minimal*, when the insider has the ability to copy or use existing attack methods; iii) *Operational,* if the insider knows the system and the underlying technology, and he is able to create new attacks within the system domain; iv) *Adept*, the insider is an expert in technologies and attack methods and can apply existing attacks or create new ones getting excellent results.

Objective. The action that the insider intends to take in order to achieve a desired outcome. Options are: i) *Copy* (make a replica of the asset); ii) *Destroy* (destroy the resource in order to make it useless); iii) *Injure* (compromise the asset in order to limit its functionality or value); iv) *Take possession* (get exclusive possession of the asset); v) *Don't care* (the insider does not have a clear plan and/or may decide only at the time of the attack).

Visibility. The extent to which the insider intends to conceal or reveal his identity. Options are: i) *Overt*, when the insider intentionally carries out an attack to the system and his identity is known at the time of the attack; ii) *Covert*, when the attack is revealed at the time of occurrence or soon after, but the identity of the attacker remain unknown; iii) *Clandestine*, when the insider aims at keeping secret his identity and the attack; iv) *Don't care*, if the insider does not place importance on secrecy.

4.3 Insider Threats

The third phase is the identification and description of possible threats to which the system could be vulnerable. This activity is of critical relevance because it allows to identify the damage that can be done on the system and the potential consequences. Through this phase it is possible to define the threats that need to be addressed with higher priority. Goals and threats may also be ranked on the basis of the level of dangerousness for the system assets, as often performed in risk analysis as for example [12]. It is important to note that for the mere identification of the threats we are not interested on the motivations that lead an insider to put into practice an attack.

For example, the potential threats which a generic system may be subject are: installation of improper software/apps (e.g., viruses, Trojans, spyware, backdoors, key loggers, logic bombs, etc.); improper data operations (e.g., data removal, exporting data, producing fake data, data modification, etc.); managing of user profiles (e.g., creation of fake users). The number of threats to be analyzed can be considerable, especially if the target system is complex. Usually a good compromise is to identify a subset of potential threats with regard to specific assets.

The approach we propose (and explore in Section 5) consists in identifying all the possible threats of interest and associate them to the insiders identified in the previous step of the methodology. As a result we get a table that lists the threats and the corresponding matching with the insiders e.g., *YES* if the insider may realize the threat in a legitimate way, *NO* otherwise. Further details for a better understanding will be given in the case study in Section 5.3.

4.4 Attack paths

This phase has the objective to identify the path(s) exploitable by the insider to realize the threat(s) and achieve the goal(s).

Several techniques exist and can result very useful for determining which threats exist in a system and how to deal with them, e.g., attack trees [18], attack graphs [19], privilege graphs [20], or adversary views [17]. The latter extends the concept of attack graph by considering different attack goals, attack preferences of specific attackers, and creating customizable models to produce quantitative analyses based on specific metric of interest.

Regardless of the adopted approach, the capability to build an attack path is of paramount importance, as it allows for example to get information on the probability of occurrence of an attack, the success rate, the weaknesses in the system, etc. There exist approaches for quantitative evaluation of attacks and attack paths [17] [35]. Such approaches are useful to quantify the probability of successful attacks, given the "success probability" of single steps, exploitable to realize the goal. For example, to perform quantitative evaluation, the ADVISE (ADversary VIew Security Evaluation) modeling formalism [17] could be used. ADVISE is a formalism for security evaluation that extends attack graphs, by including the concepts of time, costs, and success probability of the attack steps, and takes into account the attack behavior and proficiencies of different attack profiles. However quantitative security analysis may be unfeasible for insider threats because the steps of the attack paths are "authorized steps", thus "doable", and consequently it is difficult if not meaningless to define a success probability of a path.

The identification of the overall set of attack paths is a critical step, especially if we think of *unknown* paths. Attackers may be able to explore unexpected attack paths, that are unknown w.r.t. the set of attack paths considered during the threats assessment [22]. Threat analysis is limited by the ability to describe the use cases and the system paths. System evolutions may bring the emergence of new paths (thus iterations of the threats analysis can be performed to identify them), and the activity of discovering unknown paths is strongly linked to an analysis of the system requirements and functionalities.

4.5 Countermeasures selection

This phase consists in the selection of the proper countermeasure(s), in order to avoid or mitigate the identified threat(s), by preventing the execution of the attack paths, or detect or discourage its execution. The countermeasures identification and selection can be supported by i) libraries which list the countermeasures for determined attacks, and ii) techniques from the state of the art.

In general we propose to classify countermeasures in *preventive*, *deterrent*, and *detection*. Preventive countermeasures prevent the execution of an attack path (e.g., making a user unable to perform specific operations without supervision). Deterrent countermeasures do not allow to block an attack, but aim to discourage attackers (e.g., forensic data logging for a-posteriori analysis [29]). Detection countermeasures aim to timely identify the attack (e.g., the early detection solutions in [9], [7]).

Introduction of such countermeasures may require to re-assess the system, improved with the countermeasures, to verify that each threat is mitigated and give evidence of the security improvement. In case a model of the system and of the countermeasure is available, these can be integrated with the attack paths discussed in Section 4.4.

4.6 Iteration and Update

It is well known that current large scale systems and infrastructures are subject to changes and evolutions; this can require to iterate the phases of the methodology in order to align the assessment outcomes to the most recent status of the system. This require to: i) collect feedbacks about system status, ii) use those feedback to understand the evolution of the system, its users, and possible changes of system requirements, and iii) update the insider threats library and the attack paths. This will ultimately lead to define new countermeasures to be applied.

This is especially relevant in systems characterized by evolutionary and dynamic behavior, as for example Internet of Services, large-scale architectures, or cyber-critical infrastructures, in which traditional, pre-deployment verification and validation is inefficient and thus continuous (if not run-time) verification steps [25], [26], [31], [15] to cope with the changes in the system and its environment are required.

5. METHODOLOGY APPLICATION

We apply our methodology to a compact but concrete case study. For this purpose we consider our work in the context of the Secure! [13] project. We point out that the sixth phase is currently not considered, because the Secure! project is still in its design phase; consequently the assessment will cover phases 1 to 5.

5.1 System under analysis

Secure! [13] is a service-oriented system whose objectives are: i) prevent crisis as terrorist acts, vandalism, sabotage, and ii) support crisis management (e.g., in case of environmental catastrophes, human sabotages, and authorized or not-authorized demonstrations). The Secure! system will be able to integrate several heterogeneous information (audio, video, images, text) originated from different sources, including social networks (e.g., Facebook, Twitter, Flickr), emergency numbers, surveillance camera, telecommunication network, and data provided by the users on field through a Secure! application for mobile devices (anyone can be a user, after downloading the application). The Secure! system includes instruments to semi-automatically correlate, query and analyze the data, supporting both the intervention of crisis management teams, and a-posteriori forensic analysis. The framework runs on cloud system, in charge of the big-data management.

Secure! is subject to severe security and privacy requirements. In fact, in order to have users trust the Secure! system and to make it acceptable to the community, it is mandatory to provide guarantees that users authorized to work with the data collected do not compromise, counterfeit, steal or even unnecessarily query them, or do not abuse of the data correlation and data search capacity behind what is strictly necessary for their work. This calls for an attentive evaluation of insider threats. Use case diagrams for the Secure! insiders are presented in Section 5.2.

5.2 Insiders

A taxonomy of users physically or logically involved within the system has already been organized in six plausible groups of users in [13] and is described below, investigating their role as potential insiders.

Operator. A Secure! user who works in the security field, e.g., trained firefighter, or rescue personnel. He is a potential insider because he can access the system to acquire stored information, or can transmit false information.

Human Sensor. Any citizen voluntarily registered to the Secure! platform in order to cooperate using the Secure! application. He is a potential insider because he could send false information on accidents.

Domain expert. Represents an expert in the homeland security field, e.g., a police officer or a component of national intelligence. He is a potential insider because he can access private data stored in Secure!, and manipulate the data.

Unknown user. A citizen who is not registered to the Secure! platform. He may unwittingly interact with the system, e.g., by posting relevant information on social media, blog, etc. This user is in general very little expert of Secure!, and he is not acknowledged as a potential insider.

System Expert (SE). A special user which can access the system in order to use, install and configure the various services of Secure!. He is a potential insider because he has access to the internals of the Secure! system.

System Administrator (SA). A special user which has remote and local access to Secure! in order to perform maintenance tasks, removal, exports and drop of data, managing user profiles, etc. Moreover, the SA is in charge of managing emergency situations in which the system is undergoing maintenance, or under cyber-attacks, etc. He is a potential insider because he has wide access to the internals of the Secure! system.

Given the dimension of the Secure! system [24] and the amount of different interactions that groups of users can have with it, in the following of this paper we explore only the SA and SE. For a complete analysis of all groups of users we refer to [24]. Figure 1 and Figure 2 show the use case diagrams of the system related respectively to the SA and the SE.

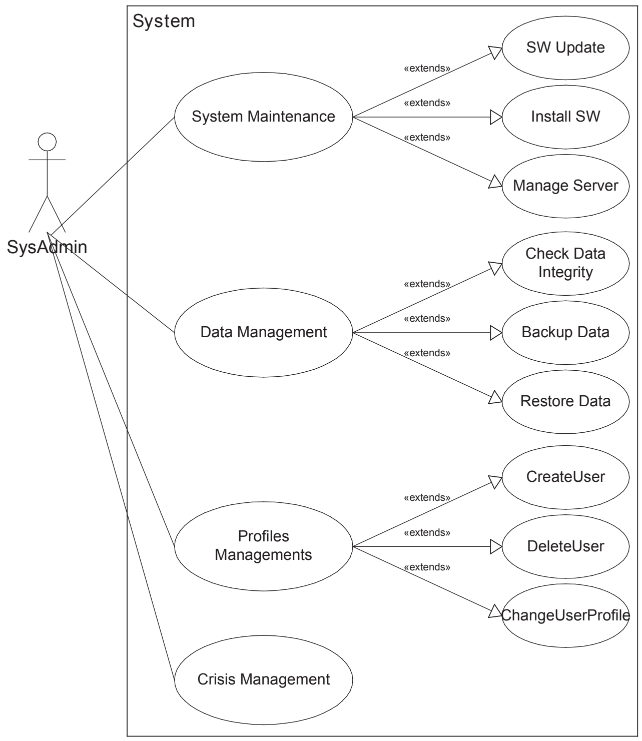

Figure 1. System Administrator UML Use Case Diagram.

Table 1. Description of UML Use Case Diagram - SA.

System Maintenance Use Case
<u>Actor/s:</u> SA <u>Pre-condition:</u> The SA must be authenticated. <u>Post-condition:</u> The SA has full access to the system. <u>Description:</u> Apply OS patches and upgrades on a regular basis the system, the administrative tools and utilities. Configure/add new services as necessary. Upgrade and configure system software or Asset Management applications. Maintain operational, configuration, or other procedures. Perform periodic performance reporting. Perform ongoing performance tuning, hardware upgrades, and resource optimization.
Data Management
<u>Actor/s:</u> SA <u>Pre-condition:</u> The SA must be authenticated. <u>Post-condition:</u> The SA has full access to the data. <u>Description:</u> Perform daily backup operations, ensuring the integrity and availability of data.
Profile Management Use Case
<u>Actor/s:</u> SA <u>Pre-condition:</u> The SA must be authenticated. <u>Post-condition:</u> The SA has full access to the system data. <u>Description:</u> Create, change, and delete user accounts.
Crisis Management Use Case
<u>Actor/s:</u> SA <u>Pre-condition:</u> The SA must be authenticated. <u>Post-condition:</u> The SA has full access to the system data. <u>Description:</u> Repair and recover from hardware or software failures or from cyber attacks. Coordinate and communicate any recovery actions.

Table 1 describes the SA Use Case Diagram, where for each use case, it is reported: the actors involved, the pre- and post-conditions, and the general description of the use case.

Similarly, Figure 2 shows the Use Case Diagram relating to the SE which is in charge of managing the Secure! dataset. The diagram shows the interactions of the system with system experts (SE). In Table 2 it is provided a detailed description of the SE Use Case Diagram shown in Figure 2.

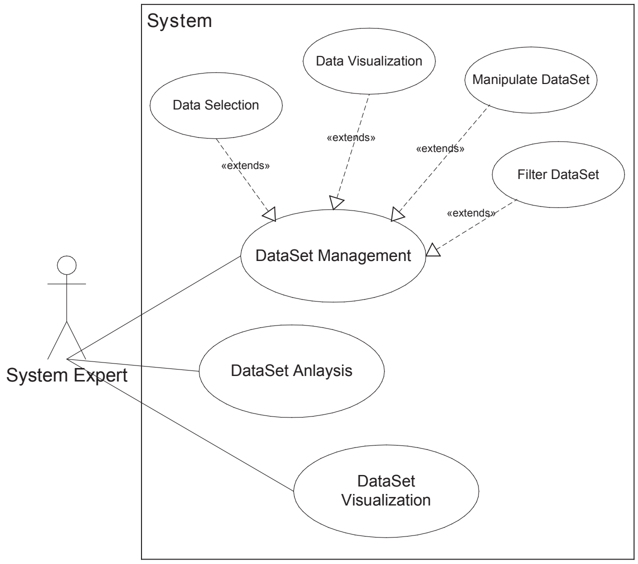

Figure 2. System Expert UML Use Case Diagram.

Table 2. Description of UML Use Case Diagram - SE

Dataset management Use Case
<u>Actors:</u> SE <u>Pre-condition:</u> The SE must be authenticated. Availability of a system dataset. <u>Post-condition:</u> Availability of the list of events analyzed (aggregates and related). <u>Description:</u> This use case allows the SE to access the data collected by the system, selecting them according to criteria established by the same (data selection), and displaying the attributes, e.g., references to time and / or geographical (data visualization). The data set can also be filtered (filter dataset) for the removal of data with low informative content. It may also be necessary for SE to carry out a data integration (manipulate datasets) from the data sources.

Table 3. Preliminary matching attributes-values.

Attribute	**Value**
Intent	Hostile/Non Hostile
Access	Internal/External
Outcome/Goal	Acquisition/Theft, Embarrassment, Damage
Limits	Code of Conduct, Legal, Extra-legal
Resources	Individual, Organization, Team
Minimum Skills	Operational, Adept
Objective	Copy, Destroy, Take, Injure
Visibility	Covert, Clandestine

According to the eight attributes defined in Section 4.2, in this step we can provide a preliminary matching Attributes-Values,

Table 4. Mapping Insiders to Threats

	Threats														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Insider	Disable system logs	Corrupt data	View confidential data	Add not required services	Improper configuration	Improper user management	Elevate users privileges	Install vulnerable supporting sw	Install vulnerable Secure! services	Use of defective hw	Transfer confidential files	Access to crypto keys	Putting Trojan horses	Disabling protection of components	Altering audit trails and logs
SA	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
SE	NO	YES	NO	NO	NO	NO	NO	NO	YES	NO	YES	NO	YES	YES	NO

showed in Table 3, which will be refined during the next *Insider threats* phase. In fact, the values assigned to the attributes could vary based on the threat to be considered. Table 3 refers to attributes and values identified for the SA. The corresponding table of SE, that is not shown for brevity, differs only in the values assigned to the Minimum Skills attribute; in this case that value for SE is set to Operational.

5.3 Insider threats

In the context of the Secure! project, we can identify a number of threats of different type of severity, which are related to the actions performed by the insiders. This phase of the methodology aims at identifying all threats whose impact, in case of successful attack, is intolerable in terms of economic losses, image, damages to the infrastructure and/or to the environment, etc.

In Secure!, for example, *human sensors* could use their own Secure! apps in order to intentionally provide fake information to the system, causing delay on supporting operations on field. System Experts and System Administrators could install and improperly configure components on the Secure! system.

The SA and SE can potentially realize a consistent number of threats with respect to the other insiders. Specifically, thanks to their privileges, they could install malicious software/code, create backdoors, disable system logs and anti-virus, create new users or change users privileges, install remote network tools, plant malware as logic bombs [16], perform operation on data base in order to have Secure! creating erroneous reports, modify, delete, and copy data.

The approach is to list all the possible threats of interest and try to associate them to the previously identified insiders, as reported in Table 4 which shows the potential threats achievable individually by SA and SE, indicated in the rows of the table. The numbers in the second row are used to identify the threats; such values are used in the logical formulae in Section 5.4, useful to identify the attack paths followed to achieve the attack goals. The threats reported in Table 4 are: i) disable system logs, ii) data corruption, iii) visualization of confidential data, iv) add additional services e.g., to collect information, v) set an improper configuration of systems components (e.g., server, firewall, antivirus, intrusion detection, etc.), vi) abuse of system management privileges, vii) change privileges to target users, viii) install unsecure, vulnerable software, ix) install unsecure, vulnerable services, x) installation and usage of defective hardware, xi) transfer confidential data, xii) disclosure access keys of users (this is typically mitigated by systems setting, at least in default configurations), xiii) add Trojan horses in the system, xiv) disable components as for example STC (*Secure-Two-party Computation*) and PEP (*Policy Enforcement Point*) [29], xv) modify log files and audit trails.

Based on the insiders under analysis and the above considerations, that are the threats of interest, we can refine the previous Table 3 and we obtain Table 5, which represent the matching attributes-values respectively for SA and SE, thus this gives a clear representation of the insiders.

Table 5. Final matching attributes-values for SA and SE.

Attribute	Value - SA	Value - SE
Intent	Hostile	Hostile
Access	Internal, External	Internal
Outcome/Goal	Damage, Acquisition/Theft	Damage, Acquisition/Theft
Limits	Code of Conduct, Legal, Extra-legal	Legal
Resources	Individual	Individual, Organization
Minimum Skills	Adept	Operational
Objective	Copy, Destroy, Take	Injure, Copy, Destroy
Visibility	Clandestine	Covert, Clandestine

5.4 Attack paths

To achieve his attack goals, that are the goal *damage* in Table 5 (it may be the performance degradation of the system or the delay/sabotage of the Secure! operations as for example emergency management operations), and the goal *acquisition/theft* in Table 5 (theft of sensitive data), the SA and the SE perform the steps described in the following of this Section.

5.4.1 *Performance degradation*

The SA may be willing to achieve the degradation of system performance. To achieve this goal, SA may succeed in realizing one or more threats among those listed in Table 4, in conjunction or separately, thus leading to the identification of different exploitable attack paths. The logical formula (1) states the threats that directly make him succeed in the goal.

$$\mathbf{[4 \vee 10 \vee [\,(\,8 \vee 9 \vee (13 \wedge 14) \wedge 5\,]] \vee}$$
$$\mathbf{\vee\ [1 \wedge [4 \vee 10 \vee [\,(\,8 \vee 9 \vee (13 \wedge 14\,) \wedge 5\,]]]\quad (\,1\,)}$$

In order to assemble defective hardware (threat 10 in Table 4), SA needs: i) physical access to the system and ii) to assemble the defective part, consequently causing a worsening of system performance.

To realize the other threats listed in (1), the SA needs to perform various actions, that constitute his attack paths. First of all, SA has to log into the system, being it locally or remote (the login is obviously not considered a threat). Then, SA can add not required services (threat 4), or perform improper configuration and system management (threat 5), as install vulnerable supporting software (threat 8), install vulnerable Secure! services (threat 9), and put

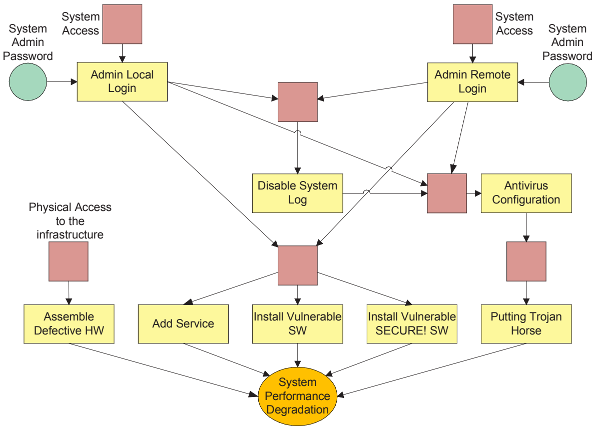

Figure 3. ADVISE attack execution graph for performance degradation.

Trojan horses (threat 13); before realizing these threats, SA may also disable system log (threat 1 in Table 4), to hide the attack. At the same time, to infect the system with a Trojan horse (threat 13), the SA after authentication may have to re-configure the antivirus or protection software (threat 14) e.g., by adding some exceptions, and again he can optionally disable the system log to leave no trace of the configuration.

Figure 3 shows the ADVISE attack execution graph representing the above-mentioned paths exploitable by the SA to realize its attack goal. To allow the reader a clear understanding of the ADVISE diagram, we briefly describe the meaning of the graphical notation: the rectangular boxes represent the attack steps; the squares are the access domain; the circles are the knowledge items and finally the ovals represent the attack goal. More details on the formalism and graphical notation can be found in [17].

As regard to the SE, the logical expression of reference is shown in (2), where we can notice that the identified threats are a subset compared to the SA.

$$\mathbf{9} \vee \mathbf{13} \qquad \mathbf{(\ 2\ )}$$

The differences are mainly due to the access modes to the system; in fact the SE can access the system solely through the Secure! Graphical User Interface (GUI), with limited privileges, and without the right to change the system configuration. We do not report the corresponding attack graph but it can be easily derived from Figure 3.

5.4.2 Theft of sensitive data

Another relevant goal that the SA may be interested to achieve is the theft of sensitive data. To accomplish this goal, SA may succeed in realizing one or more threats among those listed in Table 4. The logical formula (3) states the threats that directly make him succeed in the goal:

$$\mathbf{3} \vee \mathbf{11} \vee (\ \mathbf{5} \wedge \mathbf{13}\ ) \qquad \mathbf{(\ 3\ )}$$

The ADVISE diagram representing the attack paths to reach the goal is shown in Figure 4, where we can notice that the first attack step is the login into the system, either locally, from remote, or via the Secure! GUI. The SA may directly access the data base and execute queries, or he may execute reporting tools through the Secure GUI, in order to embezzle sensitive data. Moreover, to keep his actions hidden, the SA may accomplish some intermediate steps e.g., disable system logs, configure the antivirus, etc.

Regarding the SE, the corresponding logical expression is shown in (4), where the considerations of the previous attack goal with respect to the exploitable paths are still valid.

$$\mathbf{9} \vee \mathbf{11} \vee \mathbf{13} \qquad \mathbf{(\ 4\ )}$$

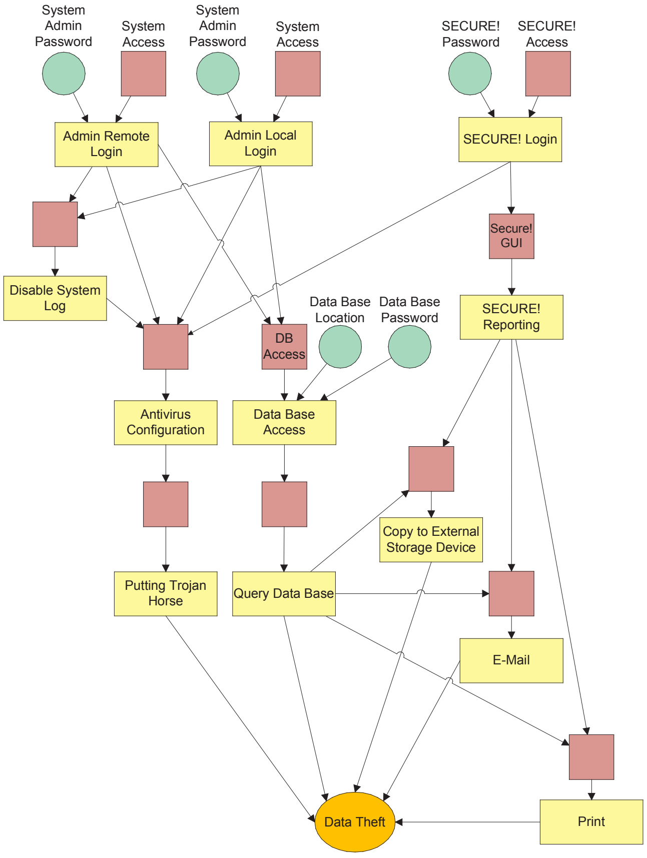

Figure 4. ADVISE attack execution graph for Data Theft.

5.4.3 Delay of the Secure! rescue operations

Another means to achieve delay/sabotage is by delaying the Secure! operations for crisis management. This approach is very relevant to the Secure! functionalities, that are i) supporting rescue teams and ii) managing crisis. The logical expression that represents the correlation between the identified threats that allow achieving the goal is shown in (5).

$$\mathbf{2} \vee \mathbf{4} \vee \mathbf{8} \vee \mathbf{9} \vee \mathbf{10} \vee \mathbf{14} \vee (\ \mathbf{5} \wedge \mathbf{13}\ ) \qquad \mathbf{(\ 5\ )}$$

Figure 5 shows the ADVISE execution graph of the attacks, where the individual attack steps are fairly self-describing or already described in previous sections.

In (6) the logical expression related to the SE is shown, where once again we observe that the considered threats are a subset of the SA case.

$$\mathbf{2} \vee \mathbf{9} \vee \mathbf{13} \vee \mathbf{14} \qquad \mathbf{(\ 6\ )}$$

The ADVISE diagram in this case, as in the previous one, should be slightly simpler because of the limited privileges owned by the System Expert.

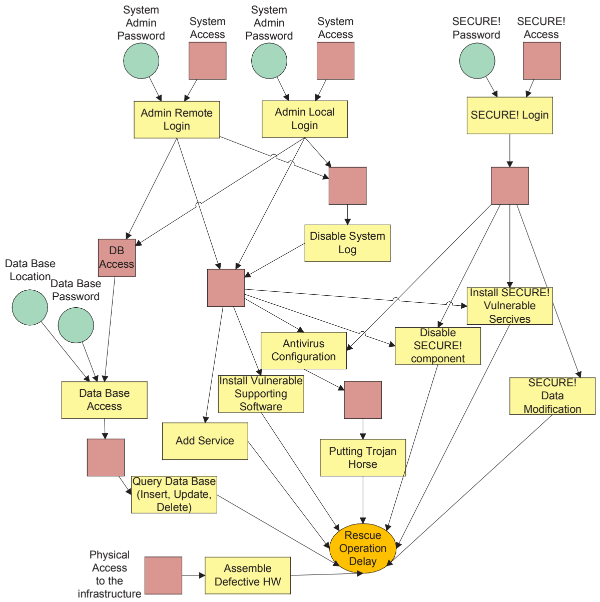

Figure 5. ADVISE attack execution graph for Delay of Secure! rescue operations.

5.5 Countermeasures selection

The individual actions which constitute the graphs in Figure 3, Figure 4, and Figure 5 are legitimate; it is their sequence and semantics that constitute the attack. Countermeasures, being them oriented to *prevention*, *deterrence*, or *detection*, must be imposed.

This subsection provides the countermeasures we have proposed to the Secure! project consortium in order to consider their implementation within the Secure! framework.

Considering the attack goals described previously, possible countermeasures valid for both the SA and SE, unless otherwise specified, are listed in the following.

Preventive countermeasures.

- Keep the technical room locked, allowing access only in presence of staff of the same grade or higher, and under authorization from a higher-ranking person (SA only).
- Avoid to log into the system during holiday days or outside the office hours, except under authorization from a higher-ranking person;

Deterrent countermeasures

- Install monitoring cameras (SA only).
- Allow forensic log of users access [30], keeping track of the username, date and time of the event (timestamp), the event description (computer system, devices, utilized software, software installation, error condition, etc.) and any other relevant metric for a-posteriori analysis [32].
- Introduce a biometric continuous authentication system [33], which every predetermined time (minutes), performs an identity check thus validating and logging user identity.
- Discourage users actions by keeping forensic logs and capability log analysis.

Detection countermeasures

- Identify the sensitive data and set up a detection system that prevents all queries on such data, except under specific authorization from a person having a higher rank, and at the same time keeping track of the activity.
- Allow printing reports only in specific printers, which are physically located in a specific protocol office in charge of release of documents under authorization;
- Implement an e-mail system with an automatic *cc* forwarding to a higher-ranking person.

Selecting one or more among the proposed countermeasures and implementing them properly in the original model, allows to re-evaluate the security of the Secure! system with respect to the considered attack.

6. CONCLUSIONS AND FUTURE WORK

Increasing attention is being paid to insider threats and attacks. Several techniques exists to avoid or detect the risk that a legitimate user abuses of its authority in the usage of the system. However, we identified a lack in the definition of a methodology and related supports for the systematic investigation and quantitative assessment of insider threats. Investigation of insider threats and mitigation is a well-known, recent topic which highlights the growing need of solutions as systems became more open, dynamics and with non-fixed boundaries, and profiles of potential users multiply.

This paper extends our previous paper [23] presenting a detailed methodology for the assessment of insider threats and an extensive case study. All steps of the methodology have been reviewed and extended, especially the first step for the identification of use cases that guide the whole process. Moreover, the application of the methodology to the Secure! case study was carried out through a deeper investigation of the potential threats and a new insider (System Expert) was considered, along with the System Administrator. These improvements allowed us to formalize the methodology and to re-apply it within the Secure! project, leading us to produce new considerations on the system vulnerabilities related to the insider threats and suggest to the designer new and appropriate countermeasures.

As future works, we are currently working on developing a generic insider threats library, both with insiders and countermeasures, to guide the application of the methodology in different systems. For this purpose, we are continuing to apply the methodology to the Secure! project and we are planning to apply it also to a very different system, namely the Smart Grid, which introduces i) new services for the control and the grid management, and at the same time ii) several cyber vulnerabilities that have to be addressed. Further exploration that is currently not addressed is in term of performance and usability analysis of the systems after the implementation of the identified countermeasures, in order to establish whether they worsen the characteristics of the system.

7. ACKNOWLEDGMENTS

This work has been partially supported by the European Project FP7-2012-324334-CECRIS (CErtification of CRItical Systems), the Regional Project POR-CREO 2007-2013 Secure!, and the TENACE PRIN Project (n. 20103P34XC) funded by the Italian Ministry of Education, University and Research.

8. REFERENCES

[1] M. Schonlau, W. Dumouchel, W. Ju, A. Karr, M. Theus, and Y. Vardi, "Computer intrusion: Detecting masquerades," *Statistical Science*, 16(1):58–74, 2001.

[2] S. Hoyer, H. Zakhariya, T. Sandner, and M.H. Breitner, "Fraud Prediction and the Human Factor: An Approach to Include Human Behavior in an Automated Fraud Audit," In *Proc. of the 45th Hawaii Int. Conf. on System Sciences* (HICSS '12). IEEE Computer Society, Washington, DC, USA, pp. 2382,2391, 4-7 Jan. 2012.

[3] A. Kamra, E. Terzi, and E. Bertino, "Detecting anomalous access patterns in relational databases," *The VLDB Journal* 17, 5, pp. 1063-1077, 2008.

[4] S. Mathew, M. Petropoulos, H. Q. Ngo, and S. Upadhyaya., "A data-centric approach to insider attack detection in database systems," In *Proc. of the 13th Int. Conf. on Recent advances in intrusion detection* (RAID'10), S. Jha, R. Sommer, and C. Kreibich (Eds.). Springer-Verlag, Berlin, Heidelberg, pp. 382-401, 2010.

[5] C. Ramkumar. A. Iyer, H.Q. Ngo, and S. Upadhyaya, "Towards a theory of insider threat assessment," *Proc. of the Int. Conf. on Dependable Systems and Networks* pp. 108-117, 2005.

[6] G. Doss, and G. Tejay, "Developing insider attack detection model: a grounded approach," *IEEE Int. Conf. on Intelligence and Security Informatics, 2009*, pp. 107,112.

[7] J. Hunker and C. W. Probst, "Insiders and Insider Threats - An Overview of Definitions and Mitigation Techniques," in *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, vol. 2, n. 1, pp. 4-27, 2011.

[8] T. Casey, "Threat Agent Library Helps Identify Information Security Risks," Intel White Paper, September 2007, Retrieved April 24, 2013 from http://communities.intel.com/docs/DOC-1151.

[9] G. Silowash, D. Cappelli, A. Moore, R. Trzeciak, T. Shimeall, and L. Flynn, "*Common Sense Guide to Mitigating Insider Threats,*" 4th Edition (CMU/SEI-2012-TR-012), 2012. Retrieved April 24, 2013.

[10] M. Bishop. "Insider is relative," In *Proc. of 2005 Workshop on New Security Paradigms* (NSPW). ACM, New York, NY, USA, pp. 77-78 Lake Arrowhead, CA, October 20-23, 2005.

[11] M. Bishop, S. Engle, S. Peisert, S. Whalen, and C. Gates, "We have met the enemy and he is us," In *Proc. of the 2008 Workshop on New Security Paradigms* (NSPW). ACM, New York, NY, USA, pp. 1-12, 2008.

[12] NIST, "Guide for conducting risk assessment," Sept. 2012.

[13] Secure! project, "D1.1 Requirements specification," May 2013, http://secure.eng.it.

[14] Q. Althebyan, "Design and analysis of knowledge-base centric insider threat models," Ph.D. Dissertation. University of Arkansas, Fayetteville, AR, USA, 2008.

[15] J. Rushby, "Runtime certification," In Runtime Verification, Martin Leucker (Ed.). Lecture Notes In Computer Science, Vol. 5289. Springer-Verlag, Berlin, Heidelberg 21-35.

[16] M. Keeney and E. Kowalski. "Insider threat study: Computer system sabotage in critical infrastructure sectors." May 2005.

[17] E. LeMay, M. Ford, K. Keefe, W.H. Sanders, C. Muehrcke, "Model-based Security Metrics Using ADversary VIew Security Evaluation (ADVISE)", 8th Int. Conf. on Quantitative Evaluation of Systems (QEST), 2011, 191-200.

[18] B. Schneier, Secrets and Lies: Digital Security in a Networked World. John Wiley & Sons, 2004.

[19] O. M. Sheyner, "Scenario graphs and attack graphs," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, 2004.

[20] M. Dacier and Y. Deswarte, "Privilege graph: An extension to the typed access matrix model," *Proc. of the 3rd European Symposium on Research in Computer Security (ESORICS '94)*. London, UK: Springer-Verlag, 1994, pp. 319-334.

[21] M. Maybury, et al. "Analysis and detection of malicious insiders." MITRE CORP BEDFORD MA, 2005.

[22] A. Moore, D. Cappelli, R. Trzeciak, "The big picture of insider IT sabotage across u.s. critical infrastructures," in: S. Stolfo, S. Bellovin, A. Keromytis, S. Hershkop, S. Smith, S. Sinclair (Eds.), Insider Attack and Cyber Security, Vol. 39 of Advances in Information Security, Springer US, 2008, pp. 17-52.

[23] N. Nostro, A. Ceccarelli, A. Bondavalli and F. Brancati. "A methodology and supporting techniques for the quantitative assessment of insider threats," in *Proceedings of the 2nd International Workshop on Dependability Issues in Cloud Computing*. M. Correia and N. Mittal eds. 2013. pp. 1-6.

[24] Secure! Consortium. Deliverable 4.1. "Modelli di gestione dei contenuti e delle decisioni," December 2013.

[25] A. Ceccarelli, M. Vieira, and A. Bondavalli. "A testing service for lifelong validation of dynamic SOA," in *2011 IEEE 13th International Symposium on High-Assurance Systems Engineering (HASE)*, Nov. 2011, pp. 1-8.

[26] A. Ceccarelli, M. Vieira, and A. Bondavalli. "A service discovery approach for testing dynamic SOAs," *in 14th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW)*, 2011, pp. 133-142.

[27] The Smart Grid Interoperability Panel–Cyber Security Working Group - NISTIR7628 Guidelines for Smart Grid Cyber Security: Vol. 3, Supportive Analyses and References, 2010.

[28] CVE- Common Vulnerabilities and Exposures, http://cve.mitre.org/index.html [last accessed 12 December 2013].

[29] Secure! Consortium. Deliverable 2.2 – Architettura della infrastruttura di gestione dell'affidabilità, sicurezza, fiducia e privacy, July 2013.

[30] M.Afzaal, C. Di Sarno, L. Coppolino, S. D'Antonio, L. Romano, "A Resilient Architecture for Forensic Storage of Events in Critical Infrastructures," High-Assurance Systems Engineering (HASE), 2012 IEEE 14th International Symposium on, pp.48-55, 2012.

[31] A. Ceccarelli, M. Vieira, A. Bondavalli. "A service discovery approach for testing dynamic SOAs," *in ISORCW 2011 - IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops*, pp. 133-142, Newport Beach, USA, 28-31 March 2011. IEEE Computer Society 2011, Washington DC, USA.

[32] M. Cinque, D. Cotroneo, R. Natella, A. Pecchia, "Assessing and Improving the Effectiveness of Logs for the Analysis of Software Faults," in *Proc. of the 40th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2010),* pp. 457-466, 2010.

[33] A. Ceccarelli, A. Bondavalli, F. Brancati, E. La Mattina, "Improving Security of Internet Services through Continuous and Transparent User Identity Verification," In 2012 IEEE 31st Symposium on Reliable Distributed Systems (SRDS), pp.201,206, 2012.

[34] S. Stolfo, S. M. Bellovin, A. D. Keromytis, S. Sinclair, S. W. Smith, and S. Hershkop. 2008. "Insider Attack and Cyber Security: Beyond the Hacker" *(Advances in Information Security)* (1 ed.). TELOS, Santa Clara, CA, USA.

[35] L. Montecchi, P. Lollini, A. Bondavalli, and E. La Mattina. "Quantitative Security Evaluation of a Multi-Biometric Authentication System." In Computer Safety, Reliability, and Security (Proceedings of DESEC4LCCI workshop - SAFECOMP 2012, Magdeburg, Germany, September 25-28, 2012), Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Volume 7613, pp. 209-221, 2012.