Volume 3, Issue 3, March-2016, pp. 152-56 ISSN (O): 2349-7084

International Journal of Computer Engineering In Research Trends

Available online at: www.ijcert.org

Design Issues and Threats in Security Risk Management

Naveen Kumar R[1] , G. Ravindra Babu[2]

[1]Reseaech Scholar, Department of Computer Science and Engineering, Siddhartha Institute of Engineering and Technology, Hyderabad, Telangana, India.
[2]Siddhartha Institute of Engineering and Technology, Hyderabad, Telangana, India.

Abstract Risk management can be thought of as a process, a theory, a procedure, or a methodology. Its primary objective is to identify assets, vulnerabilities, and threats and then to protect those assets. Risk management is crucial to any organization for the simple reason that it is the best available tool that enables them to determine the level of protection required for their many different assets at the lowest possible cost. A few different approaches have been created for the purpose of managing the risks associated with information security. These methodologies incorporate a variety of strategies, procedures, and perspectives in order to analyses and evaluate risks.

Key words — Risk management, Information Security Management, Management Framework

1. INTRODUCTION

Various risk management methods are discussed in brief in terms of design issues, methodologies and attacks.The rapid growth of attacks against computer network systems and the high cost of the available security countermeasures have favored structured high-level methodologies aiming at evaluating the security state of such systems and selecting the most convenient defense measures. Therefore, integrating security issues into the business activity of enterprises has been an important challenge since several decades.

One of the earliest risk management approaches was developed on 1979 by Campbell who developed a structured methodology based on a set of concepts that subsisted in the later approaches such as vulnerability analysis, threat analysis, risk analysis, and control implementation. Summers proposed a similar method based on four steps:

1. Asset analysis: Identifying assets and assigning values to them.
2. Threat and vulnerability identification.
3. Annual Loss Expectancy (ALE) calculation for each threat.

Safeguard selection: Effectively, these approaches came to enhance earlier efforts within this field. Since the mid-1970s, the National Institute of Standards and Technology (NIST, previously the National Bureau of Standards) began to address some aspects of risk management, especially risk assessment. The ALE is a concept that allows assessing and ranking security threats. This pioneering work proposed some mechanisms to evaluate quantitatively security risks.

Throughout the difficulties that were faced when applying these methods, it has been realized that quantitative risk assessment is very hard to conduct. This was the result of two major factors. First, as security models were, at this moment, still in infancy, building a representation of the analyzed environment was a very complex task. The huge amount of data collected from various components of the information systems could not be treated accurately. Second, the lack of automated tools supporting the risk management activity rendered its manual execution more difficult. Consequently, easier subjective methods, called qualitative approaches, have been introduced. The qualitative valuation of the basic risk analysis variables (e.g., asset importance, threat probability, threat impact) consists in a simple scaling. For instance, the probability of occurrence for a specific threat can be either “high”, “medium”, or “low”. Obviously, the collection of knowledge becomes much easier than in the quantitative case. In fact, instead of

relying directly on digital security data such as audit log records, interviews and questionnaires are used in the qualitative case.

In addition, RM methodologies can be divided into two categories: bottom-up approaches and top-down approaches. The bottom-up approach consists in selecting a priori the residual risk, e.g., the degree of protection of the system, and implementing the countermeasures that allow reaching it. Top-down risk estimation defines scheduled tasks that are intended to reduce the identified threats. Examples of these tasks include, but are not limited to, risk identification, risk mitigation, system state monitoring, and risk assessment.

For a long while, RM has been a focal point of interest for both governmental institutions and research laboratories. This resulted in the development of many standards, guidelines, and models that contributed consistently to the advance of this field. The most important RM related works are listed below:

1. Standards
a) The Australian and New Zealand Standard on Risk Management.
b) The ISO 17799 Standard.
2. Guidelines
a) Risk Management Guide for Information Technology Systems, by the Na-tional Institute for Standards and Technology.
b) A Guide to Risk Management and Safeguard Selection for IT Systems, by the Communications Security Establishment, Government of Canada.
c) Information Technology Security Evaluation Manual (ITSEM), by the Com-mission of the European Communities.
d) Information Security Management: Learning from Leading Organizations, by the United States General Accounting Office.
3. Models
a) The OCTAVE approach, by Carnegie Mellon University.
b) The CORAS framework (EU-funded project) .

In the same way, several tools have been developed to automate the RM activity or at least some of its tasks. The most important RM tools are listed below:

1. COBRA, developed by C&A Systems Security Ltd.
2. RiskWatch suite, developed by RiskWatch Inc .

Even though the security objectives slightly differ from an approach to another, a lot of interesting subtleties have been introduced at different levels (e.g., architectural modeling, implementation design, decision-making algorithms). More details about these considerations are given in the following subsections.

2. THE OCTAVE METHOD

The Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) have been developed jointly by the Carnegie Mellon University and the Software Engineering Institute. It is a self-directed method relying on a small team, called analysis team, including both business and information technology personnel. The OCTAVE process has been structured with respect to the various categories of results that could be reached by this team. Mainly, three types of outputs have been considered: organizational data, technological data, and risk analysis and mitigation data.

Unlike the typical technology-focused assessment, OCTAVE focuses on organizational risk and strategic, practice-related issues, balancing operational risk, security practices, and technology. OCTAVE uses a three-phase workshop-based approach enabling organizational personnel to assemble a comprehensive picture of the organizations information security needs. These phases, illustrated by Figure 1, are:

Phase 1: Build Asset-Based Threat Profiles - This is an evaluation of organizational aspects. The analysis team identifies the assets within the organization and what is currently done to protect them. The team then determines the assets that are most important to the organization (critical assets) and defines the corresponding security requirements. Finally, it identifies threats to each critical asset and builds a threat profile for that asset. It is worth mentioning that threat profiles are derived from the event-tree concept.

Phase 2: Identify Infrastructure Vulnerabilities - This is an evaluation of the in-formation infrastructure. The analysis team examines the information technology infrastructure for weaknesses that could be exploited to gain unauthorized access to information or to disrupt information processing. Vulnerabilities include here (1) those that are inherent in the design or specification of the system's hardware or software, (2) those that occur from a flawed software or hardware implementation of a satisfactory design and (3) those that stem from system configuration or administration errors.

Phase 3: Develop Security Strategy and Plans - Risks are analyzed in this phase. The information generated by the organizational and information infrastructure evaluations (Phases 1 and 2) are analyzed to identify risks to the enterprise and to evaluate the risks based on their impact to the organization's mission.

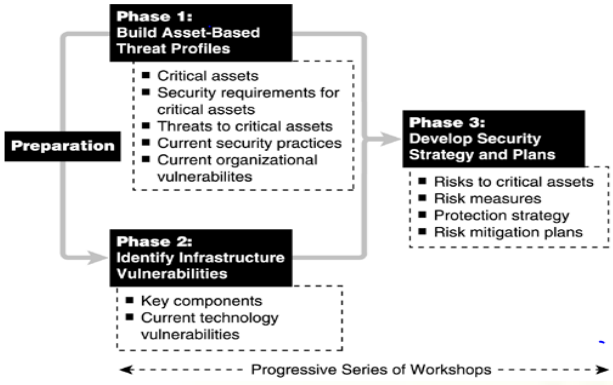

Figure 1. The OCTAVE Method

3. THE CORAS FRAMEWORK

CORAS is a EU-funded RM project that has been conducted between 2001 and 2003. A wide spectrum of risk-related issues has been addressed in the frame of this research. This has resulted, among others, in the definition of a global model-based approach for security risks relying on several aspects that have been borrowed from existing risk assessment techniques. One of the key components of CORAS framework is an automated tool supporting the methodology. It integrates a UML-based specification language that can be used for three major purposes consisting in: (1) representing accurately the system state (from a security point of view) and the interaction between the various entities; (2) standardizing the communication between the risk assessment team members through the use of a uniform language; and (3) documenting the risk management activities. Moreover, this method highlights the library concept. The processes of the CORAS RM are illustrated by Figure2. They are briefly discussed in the following. Context identification aims at determining the security-critical assets as well as the related security requirements. CORAS relies on the initial phase of the CCTA Risk Analysis and Method Management (CRAMM) to address this need. A questionnaire submitted to system users is used to determine the important data groups" (i.e., the relevant assets that should be analyzed).Risk identification consists in identifying the weaknesses of the crucial components and the potential threats that may harm them.

Different methods are used at this level:

– Fault Tree Analysis (FTA) identifies causes for an unwanted outcome event (top-down approach).

– Failure Mode, Effects, and Criticality Analysis (FMECA) focuses on single failures of components (bottom-up approach).

– CRAMM addresses threats/vulnerabilities of an asset by means of predefined questionnaires.

– Goal means task analysis (GMTA) identifies tasks and pre-conditions needed to fulfill an identified security goal.

Risk analysis has the objective to investigate possible consequences of unwanted outcomes, as well as the likelihood of their occurrences. FMECA supports well this activity if the system description is sufficiently detailed. Moreover, Hazards and Operability (HazOp) analysis can be useful when there is no detailed knowledge about the analyzed system. Those methods are complementary in the sense that the former is quantitative while the latter is qualitative. FTA and Markov analysis can be used to support HazOp analysis as they respectively assign probabilities to the basic events in the fault trees (to compute the ones of the top events) and evaluate the likelihood of sequences. Risk evaluation ranks the identified risk events on the basis of the likelihood of occurrence and on the impact. This process is based on FMECA to handle quantitative parameters and on CRAMM to combine qualitative weights. A cause-effect analysis is also performed to determine relationships between risk events. Risk treatment defines the strategy that should be applied to thwart the potential attacks. Multiple options are available at this level such as risk avoidance, reduction of likelihood, reduction of consequences, risk transference, and risk retention.

Communicate and Consult

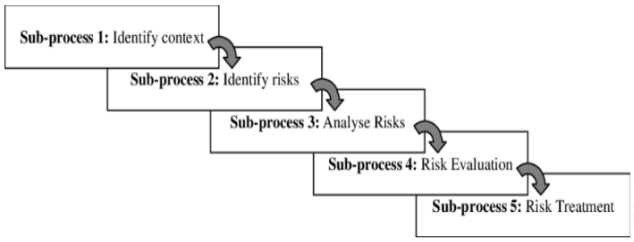

Figure 2. CORAS Risk management Process

3.1 Limits of Existing Methodologies

The historical overview of RM shows that an abundant activity has been directed around RM. Nevertheless, some important issues remained almost uncovered. Since the development of the first risk management method , many attempts to develop new approaches and to strengthen the existing ones have arisen. Nonetheless, the major factor that limits the application of those methods in the real world is the lack of suitable software tools. Effectively, risk analysis is so complicated that performing it manually would be quasi unfeasible. A glance at the existing automated risk management tools shows that they do not keep up with the evolution of the related theoretical models. More importantly, all of these software are incomplete, in the sense that they do not allow their users to perform a complete risk management cycle. Additional obstacles to the widespread application of risk management may stem from several intrinsic weaknesses of the existing approaches. In the following, the most relevant shortcomings of the traditional risk management techniques are highlighted.

• Cost estimation: In view of the fact that security planning itself has a cost to the enterprise that conducts it, the amount of effort related to a risk management project should be conveniently predicted. Naturally, if the cost of security planning overtakes the allowed budget, risk management operations have no reason to start. The absence of techniques estimating the required effort with regard to the characteristics of the analyzed system is one of the key reasons that alter the confidence of IT users, especially high level managers, in risk management.

- Attack modeling: Representing attacks against computer networks is among the most challenging tasks the risk analyst should do. This stems from the intrinsic complexity of these attacks, which can be reduced to the following points:

- Primo, as network attacks are conducted by humans, a corresponding predictive model would be hard to evolve. On the opposite of various events, human thoughts can not be easily translated into a formal representation. Most of the existing methods assume the existence of an attack database without addressing this issue.

- Attacks are seldom performed in a single step. Attackers often follow a structured methodology to achieve malicious acts. Hence, the risk analyst should establish various kinds of links between elementary threats (e.g., causality, time precedence). The major shortcoming of the traditional approaches is that they do not integrate security countermeasures and enterprise resources as a part of the expert knowledge related to attack scenarios.

- Control selection: In traditional risk management approaches, it is customary to map statically security decisions to digital attacks. In other terms, a set of countermeasures correspond to each attack, in the sense that they mitigate it. We found that this static mapping does not satisfy totally the needs of the risk analyst because it can be the source of many problems, especially when evaluating the efficiency of a decision. Consequently, several conditions should be integrated within the security solution selection module to make the choice of the optimal countermeasure alternative depend not only on the possible attacks but also on the environment of the studied system (e.g., network topology). As a result of this reasoning, the security solutions that are optimal for an enterprise are not necessarily the same as for another enterprise even if both of them face the same threats.

- Monitoring and response: This task is often reduced to a continuous control of the system state. However, given that it belongs to a risk management cycle, it

seems inadvisable to confine monitoring to its pure security facets. It should be fully integrated in the business activity of the enterprise. Incident response, for instance, should be enriched by a set of models allowing the accurate measurement of an incident impact, and the selection of the appropriate reactions.

In the following, we focus deeply on the limits of the existing RM methods by classifying them in architectural, technical and environmental limits.

3.2 Architectural Limits

We found that the following aspects constitute the most important short-cuts of the existing RM methods:

There is no separation between preventive and reactive RA - A glance at Figures 1, 2 shows that the related methodologies focus on preventive risk analysis and subordinate reactive risk analysis. As it will be explained later in Section 2, preventive and reactive risk analysis exhibits many differences. The former needs an a priori reasoning about security (before the occurrence of a security incident) while the latter requires a posteriori intervention after a security incident has occurred. Moreover, preventive risk assessment relies on probabilities of occurrence of the identified threats. Reactive risk assessment relies on alerts and therefore introduces probabilities which are principally related to the efficiency of the detection mechanism. As a result, it is advisable to give a bigger importance to reactivity in the RM architecture. Unfortunately, existing RM approaches have not addressed this point. Real-time reactivity is quite absent - When speaking about reactive countermeasures, time is fundamentally important. According to the nature of the attack, security solutions might be implemented immediately (real-time response) or after a collaborative reasoning process involving both stakeholders and security specialists. This differentiation is not made by the methods discussed above. In fact, this depends essentially on how the attack impact varies according to time. In some cases, the decision to block a TCP or UDP port must be made in a very short period of time (a few seconds). Other countermeasures may require the approval of high-level management before being implemented, especially when they cause an important loss to the enterprise. This means that a compromise between security solutions cost and benefit should be ensured. With regard to the rapidity of most of the known attacks, a high proportion of this decision-making framework should be automated. Only crucial decisions should be subjected to a human approval. Exploit libraries are not described as they should be - Putting aside the structure of RM libraries, the security analyst should focus on how to ensure these functions. Due to the large number of vulnerabilities, attacks, alerts (that can be evaluated by thousands), semantic links cannot be built manually (i.e., by the security analyst himself). Suppose that we just want to associate to every attack the vulnerabilities that it exploits. If attacks and vulnerabilities are considered ; and giving that the risk analyst has to inspect, for every attack, the whole set of vulnerabilities to select those that can be exploited by the attack of interest, then the total number of semantic links that should be addressed equals × . This is equivalent to 10000 links if = = 1000. Obviously, one may argue that databases provided by official security organisms (e.g., SANS Institute, CERT, NIST) can be used to cope with this problem. Unfortunately, these databases can be used only in certain simple situations. In fact, they are restricted to system weaknesses and do not consider human or documentation-related vulnerabilities. Consequently, one of the key components of a RM methodology is a language which uniformly treats vulnerabilities, attacks, and alerts. Of course, this language should support several automated inference operations such as mechanically determining which vulnerabilities can be exploited by a given threat, which security alerts correspond to a specific attack, or which countermeasures can thwart a harmful action.

Decisions libraries are not addressed - Another important lack of the existing RM methods consists in the absence of decision libraries. A careful consideration of these approaches shows that an accurate structure is needed to represent

countermeasures. This can substantially facilitate operations such as searching or sorting. Furthermore, an appropriate syntax should be defined to allow the automated association with attacks, assets, and vulnerabilities from the one hand, and security countermeasures, from the other hand. The existence of accurately structured decision repositories would in fact add substantial contribution to the automation of RM processes.

3.3 Technical Limits

Technical limits have also been remarked:

- The available approaches are not homogeneous - One of the most important limitations of OCTAVE and CORAS stems from their lack of uniformity. The techniques that have been proposed to implement RM processes seem disparate and hard to integrate with each other's. For instance, combining the myriad of risk analysis techniques proposed within the frame of the CORAS approach requires an important adaptation effort, even though a uniform language (UML) is used to specify security. Naturally, a RM cycle consists of various activities that cannot be described nor implemented through the use of a single technique. Henceforth, what is required is to take into account the effort that should be made to set up a toolbox that adapts and integrates many techniques. In this context, the use of specification tools that provide an abstraction view of RM processes and tasks may be very useful.

- Sophisticated techniques are rarely integrated in the RM life-cycle - As it has been underlined in [8], many sophisticated tools and theories have been evolved around most of RM tasks, especially vulnerability detection and intrusion detection. Nonetheless, RM methodologies have not kept up with those research advances. Techniques that are proposed by OCTAVE or CORAS are certainly easy enough for being quickly understood and applied but they also present some severe limitations. For instance, HAZOP (used by CORAS to identify security threats) is not compatible with automated vulnerability tools.

Investigation and response to security incidents are absent - The existing RM methodologies are generally ended at the proposal of security countermeasures step. They do not integrate an activity of monitoring the system security state. In addition, security incidents are not handled by these methodologies. Despite the importance of investigation of security incidents and response to them, these activities are missing in OCTAVE and CORAS approaches.

3.4 Environmental Limits

Some other environmental shortcuts have also been detected:

The existing methods are not integrated in the SE process - OCTAVE and CORAS are generally applied in a company independently to any SE process, if it exists. Involving the RM tasks to the SE process enhances the efficiency of the RM methodologies. This allows for example to take the appropriate decisions when the system evolves or when responding to incidents. Hence, the non-integration of the existing RM methods to the SE process keeps these methods isolated from the evolution of the system and of the attack techniques.

Absence of techniques for capitalizing the team knowledge - The existing RM methods does not integrate techniques enabling to capitalize the team's knowledge. For instance, neither OCTAVE nor CORAS is presented to integrate the skills and knowledge of the teams in the computation of several parameters including the updates cost.

The existing methods do not capitalize the acquired experience - The existing RM methods do not provide any technique to capitalize the experiences acquired during all the incident responses performed within a company implementing such method. This shows for example that all the libraries useful for the process of incident response are built internally.

4. RISK ANALYSIS

Risk analysis basically stands for setting up relationships between the main risk at-tributes: assets,

vulnerabilities, threats, and countermeasures [8]. Obviously, the main objective of building this quad is to achieve a precise representation of events that can affect the security of the information system and to deduce the corresponding security countermeasures. In other terms, risk analysis aims at (1) identifying the risk events threatening a particular system, (2) assessing their magnitudes, and (3) maintaining this magnitude below an upper threshold. Risk analysis is among the most complex components of the risk management framework. It generally involves many processes of the risk management cycle. In the following, we present the ingredients of a generic risk analysis activity and we describe several techniques that are commonly used in this context.

4.1 Architecture of the Risk Analysis Process

Generally, risk analysis breaks into three elementary activities which are: Threat analysis, which consists in determining the potential attacks and modeling them accurately. It has been affirmed that threat analysis should involve the examination of three basic elements: (1) the agent, which is the entity that carries out the attack, (2) the motive, which is the reason that causes the agent to act, and (3) the results, which are the outcome of the occurrence of the attack. Another element that can be added to the aforementioned ones is the attack mechanism. This key factor can serve to differentiate between threats that share the same agent, motive, and results. In addition, analyzing the steps to carry out a harmful action is helpful for automating several tasks, such as the selection of appropriate defense solutions.

Business Impact Analysis (BIA), which consists in evaluating the effect of the occurrence of an attack against a given asset (or group of assets). The major outcome of CBA is the ROI. This gives high-level managers the ability to define a strategic view of the efficiency of the potential countermeasures. For instance, balancing the initial cost of a security decision against the value of the assets that it protects might be insufficient. An idea about the period necessary to recoup the cost of the concerned safeguards is often required as some protection equipments depreciate more rapidly than others.

Cost Benefit Analysis (CBA), which consists in comparing the cost and the benefit of the candidate security countermeasures to select the more appropriate ones. It should be remarked that asset identification (or analysis) has been considered as an important issue. In the RM jargon, asset identification and valuation corresponds to BIA. Its goal is to estimate the replacement cost of the security critical assets, as well as the values of their security attributes (i.e., confidentiality, integrity, availability). Depending on whether the risk analysis method is quantitative or qualitative, these values can be expressed in monetary or non-monetary terms. The purpose of a BIA is to identify the impacts of the unavailability of some specific assets. This is indirectly needed to assess the efficiency of the potential countermeasures. Therefore, BIA is strongly linked to CBA as it will appear from the following discussion. Likewise, BIA has an important influence on the Business Continuity Plan (BCP) development, which consists in determining the impact of security incidents on the vital business processes in order to ensure their continuous availability. However, when being directed towards the establishment of a BCP, the BIA should slightly differ from the normal case as some parameters, such as the Maximum Tolerated Downtime (MTD), should be considered during the asset valuation.

4.2 Classifying Risk Analysis Techniques

A first idea to classify risk analysis approaches consists in reasoning about their con-tinuity. In fact, when looking for locating the risk analysis activity in the RM cycle, it turns out that two possibilities are available: (1) prior to the implementation of the security solutions, and (2) during the monitoring (or the incident response) process. The first alternative allows to think of risky events before their occurrence in order to make the infrastructure more convenient with the enterprise context. Therefore, this type of risk analysis approaches is called preventive as its main

concern is to reduce the risk magnitude so that the system state becomes as close as possible to the ideal state (where no attack is possible to carry out). The second possibility is to conduct post facto risk analysis, meaning that the selected countermeasures should limit the effect of malicious acts after they substantially occur. This category is called reactive reactive risk analysis. Although this topic has been plentifully addressed by researchers, it has seldom been put in the frame of RM approaches. It has been rather considered as a functionality of several IDSs. In fact, an interesting point might be highlighted at this level. Even though the goal of a risk analysis activity does not depend upon if it is preventive or reactive, its constituency might be heavily affected by this factor.

The major difference between preventive and reactive risk analysis resides in the uncertainty formalisms. More concretely, when reasoning about preventive counter-measures, the risk analyst has not a full idea about the occurrence of the attacks. Henceforth, a probability of occurrence is assigned to every threat event. On the other hand, reactive incident response relies on the actual occurrence of the harmful event. What is uncertain in this case is whether the attack is effectively related to the detected signs of intrusion. Traditionally, alerts generated by IDSs form the major indicator of the occurrence of an attack. Two factors dictate the introduction of uncertainty formalisms at this level. First, security alerts do not necessarily imply the actual presence of an intrusion a they can correspond to False Positives (FPs). Second, more than one attack can correspond to a single alert, meaning that a weighted link can be built between the potential attacks and the generated alerts. Consequently, it appears that the attack-alert paradigm is at the center of the reactive risk analysis. Moreover, the security attributes differ from the preventive to the reactive case. For instance, the probability of occurrence, which is used to assess attacks prior to their occurrence, cannot be considered for incident response. Even though it can be substituted by the probability of a True Positive (TP), this latter is global as it does not vary from an attack to another.

4.3 Risk Assessment

The risk assessment module is, in some sense, the core of the computational framework in a RM process [8]. It is customary done according to two factors: impact and probability which represent respectively the damage that would result from the occurrence of a potential threat and the chance (likelihood) that the risk becomes actual. As three concepts can be modeled in various ways, a plethora of risk assessment techniques have been developed. This section explains how these techniques can be categorized and explores the most relevant models that associate formal representations to computer security risks. Finally, to complete the decision support framework, methods allowing the selection of the best countermeasures are reviewed.

5. QUANTITATIVE VS QUALITATIVE APPROACHES

As it has been underlined in the foregoing discussion, risk assessment can be performed either quantitatively or qualitatively. In this subsection, we highlight briefly the ad-vantages and the shortcuts that are associated with each of these approaches .

Advantages of quantitative risk assessment

- A rich variety of metrics can be used to represent and evaluate the various risk parameters. This allows a more granular analysis of the risk events.
- The values of the risk parameters are expressed according to their nature (e.g., monetary units for asset importance, threat impact). As no translation to a different scale is necessary, those values are more accurate.
- Sophisticated decision-making techniques can be used as the quantitative assessment provides a credible set of parameters. This aspect is useful to fulfill the cost effectiveness requirement.

The results of the risk analysis process can be expressed in management's language. This makes it

more efficient to help the enterprise in reaching its business objectives.

Limits of Quantitative risk assessment:

The computational methodologies are often complex. Managers may face several difficulties to understand some advanced aspects. An important hardware and software infrastructure is needed to conduct quantitative risk assessment approaches. Automated tools are then required to support the manual effort in order to accelerate the process and to make it more precise by avoiding computation errors. Nonetheless, such software are not always available. Developing an appropriate tool for an enterprise may be so expensive that the managers become reluctant to proceed to a risk management activity. Data collection requires a substantial interest. In fact, the gathered information must be precise enough to ensure the required analysis efficiency. On the other hand, the data collection mechanisms should be themselves secured to prevent the unauthorized access to sensitive data or the violation of privacy policies.

Advantages of qualitative risk assessment:

The process does not involve a complex reasoning and it can be understood by high-level managers. This makes it easier to convince them about the importance of the risk assessment process. Qualitative risk assessment can often be performed manually. Computational steps are restricted to simple arithmetic operations on a reduced set of integer numbers.The data collection process is less complex than in the quantitative case. Questionnaire-based techniques are often used to this end. The use of several parameters that cannot effectively be measured but that can be subjectively evaluated gives more freedom to the risk analyst to choose the assessment parameter basis.

Limits of qualitative risk assessment

The lack of theoretical bases increases the uncertainty rates that affect the risk assessment results, which do not give a precise knowledge about the identified risk events.

The efficiency of the qualitative reasoning relies on the expertise of the risk assessment team (or the population that responded to the questionnaire). This is an important weakness as security specialists are not always available at reasonable costs.

The results of the qualitative risk assessment process cannot be substantially tracked because their subjective nature renders them hard to evaluate.

6. RISK ASSESSMENT FOR PREVENTIVE RISK ANALYSIS

Almost all of the computer security risk assessment methods rely on the ALE, which is derived from the Annualized Rate of Occurrence (ARO) and the Single Loss Expectancy (SLE). Formally, let (resp.) be the value (resp. the exposure) assigned to the studied asset. is the percentage of the asset that would be lost should the attack be realized against it. Then, the Single Loss Expectancy (SLE), is equal to × and:In the following we discuss briefly some relevant aspects related to each of these variables. The ALE is expressed in annualized terms in order to be easily integrated within the business planning activity. The ARO stands for the frequency with which a given threat is expected to occur during a year. This parameter is global in the sense that it is not concern a specific asset.

7. RISK ASSESSMENT FOR REACTIVE RISK ANALYSIS

Our research lab has presented a cost model for reactive countermeasures relying on the following prominent decision criteria:

Detection cost: Prior to the generation of a security alert by an analyzer, the corresponding sensor should have captured appropriately the required parameters (e.g., metric value, packet header fields). Then, the analyzer inspects these data to decide whether an intrusion occurred or not. Both of these operations have a cost that must be taken into account when

computing the total cost of the incident. Hereinafter, this cost is supposed to be intrinsic to the elementary IDS (consisting of an analyzer and a sensor) and to be independent from the nature of the generated alert. A way to express this detection cost is denoted by (γ) $\in$ {1,⋯ , } where is the analyzer.

Cost of reaction: Each of the potential reaction ($\in$ {1, . . . , }, where is the total number of security countermeasures) has a cost γ that depends heavily on the reaction itself. It can be expressed in terms of different attributes such as monetary units or processing resources.

Attack impact: Carrying out a malicious act on an information system causes various kinds of undesirable effects. To this purpose, the impact ι , which rep-resent these effects, will be represented by multiple attributes. Furthermore, a novel concept, called progression factor and denoted λ , is introduced to have a more suitable representation of the benefit of a given reaction. In fact, the system can react before or after the attack occurs. Each of these reactions has a different benefit. The former aims at stopping the complete execution of the attack after receiving signals that reveal it while the objective of the latter is to make the system recover at the right time. To this purpose, the impact is modeled as a function of the progression factor as it will be illustrated below.

IDS Efficiency: Efficiency represents a fundamental topic in IDS research. It should detect a substantial percentage of intrusion into the supervised system while still keeping the false alarm rate at an acceptable level. In our case, it relies on two factors: the alert and the related attack. To this purpose, two random variables are defined. is related to the detection activity and equals 1 if an alarm is generated and 0 if not while states if an attack occurred. Similarly, it is equal to 1 if an attack took place and 0 if not. According to this reasoning, the efficiency of the IDS can be measured using the following conditional probabilities:

(|): represents true positive rate. It can be estimated by submitting a packet flow which is contaminated of ''known'' attacks,

(|¬): represents false positive rate. It corresponds to the case where the detector issues an alarm while the corresponding attack did not occur,

(¬ |): represents false negative rate. It measures the probability of non-generation of an alert in case of occurrence of a security threat,

(¬ |¬): represents true negative rate. It corresponds to the absence of security alerts knowing that no attack has been conducted against the system. Concerning the impact distribution of security threats, numerous models can be developed. Three examples are given below, they can be obviously enriched by other functions.

1. Constant impact: the impact function has zero tolerated downtime.

2. Linear impact: in this case, the Maximum Tolerated Downtime (MTD) is introduced in the impact function. Practically, the MTD corresponds to the maximum time interval in which the service of interest can be stopped. In the following expression, the MTD is denoted 0.

Exponential impact: this model corresponds to attacks that have an impact that increases rapidly across time. These functions model the variation of the impact according to time. More sophisticated approaches can be conducted. For example, the rank of an attack in a scenario (i.e., composite attack) can be significant in some contexts.

Furthermore, the evolution of the attack is considered only between its occurrence and the MTD. In fact, it is supposed that the IDS can react only during this interval. Applying reactions after the MTD should be subjected to a more complex decision framework, which is the Disaster Recovery Plan (DRP). This process is effectively collective and requires the intervention of Incident Response Team

(IRT) members. Therefore, the decision analysis presented here cannot be applied in this context.

In fact, as the benefit results from the fact that the attack is stopped at the progression degree λ , the positive effect of this reaction can be approximated by $\iota(1) - \iota(\lambda)$ where $\iota(1)$ is obviously the maximum damage of the attack . In addition, $\iota(\lambda)$ appears as a cumulative sum of the elementary impacts of the attack in the interval $0, \lambda$.

8. SECURITY POLICY

It is obvious that information systems must be secured against security threats. Identifying which assets should be protected, from what they must be protected and which security counter-measures should be in place require careful planning and attention detail. Thus, it is essential that an organization writes, with regard to its security requirements, a security policy enumerating the security rules to be respected. Notice that accurate specification for security solutions requires that an organization derives its security policy from risk management activity.

SECURITY POLICY DEFINITION

ISO 17799 standard states that information security policy is a document that aims at providing management direction and support for information security. More technically, RFC 2196 defines a security policy as a formal statement of the rules by which people who are given access to an organization's technology and information assets must abide.

Finding a precise meaning to the security policy turns out to be very arduous as it is used to refer to numerous disparate aspects of information systems security [33, 4]. The definition of a security policy remains narrowly related to the context in which it is used. In the following we give some examples highlighting the fact that the definition of a security policy is narrowly related to the context in which it is used

Network security policy: The communication infrastructure is often used to carry out various attacks (e.g., flooding, e-mail spoofing). Therefore, the system should prevent network nodes from forwarding suspicious traffic. Networked as-sets should behave securely to cover the existing weaknesses. The network access policy consists of a set of rules indicating how data should be transmitted across the network. It consists of two components: the preventive network access policy, and the reactive network access policy. The first attempts to cover vulnerabilities so that attacks do not occur while the second aims at limiting the damage resulting from the occurrence of a security incident.

Access control security policy: Due to numerous security threats that exploit weaknesses at the operating system (OS) level, a set of protection mechanisms should be implemented to plug up such vulnerabilities. The set of the protection mechanisms related to OS is called Trusted Computing Base (TCB). They concern the various resources of the computer system (e.g., hardware, software, processes). The most relevant example consists of the access control policy which is enforced by secure OSs to protect the objects they handle. Obviously, for consistency and completeness purposes, those mechanisms should abide by a set of rules, which form the security policy. The reference monitor is an entity that mediates access to objects by subjects. Among those accesses, only those that conform to the security policy are allowed. The reference monitor basically guarantees that the OS respects several pre-defined security principles such as least privilege and continuous protection [33].

Key management security policy: To establish a secure tunnel using the IPSec protocol suite, two end-points should agree upon a set of mutually acceptable cryptographic parameters called Security Association (SA). These security parameters are managed according to local security policies which are set in each end-node. For example, when creating a new SA in order to modify an older one, "deletion of the old SA is dependent on local security policy". Besides, a standard has been recently developed to

administrate IPSec security policies; it defines the concept of IP Security policy (IPSP) [33].

The examples listed above present the security policy seen from different angles. To unify all these views, we define the security policy as "a set of rules that determine how a particular set of assets should be secured" [33]. This definition may appear to be too general, but it has the merit to extend to most communication network contexts.

Notice that security rules are not intuitive. They are derived from the risk management activity. After risk assessment is performed, several security rules are proposed to mitigate the risks. However, we can not retain all the possible security rules because some rules have the same impact than others. It is then clear that we must select the most appropriate one. Appropriateness is delimited by a set of criteria. Examples of these criteria are the acquisition cost of the security countermeasures and their implementation effort.

9. TY POLICY STRUCTURE

According to the RFC 2196, the components of a good security policy include:

1. Computer Technology Purchasing Guidelines which specify required, or preferred, security features. These should supplement existing purchasing policies and guide-lines.

2. A Privacy Policy which defines reasonable expectations of privacy regarding such issues as monitoring of electronic mail, logging of keystrokes, and access to users' files.

3. An Access Policy which defines access rights and privileges to protect assets from loss or disclosure by specifying acceptable use guidelines for users, operations staff, and management. It should provide guidelines for external connections, data communications, connecting devices to a network, and adding new software to systems. It should also specify any required notification messages (e.g., connect messages should provide warnings about authorized usage and line monitoring, and not simply say "Welcome").

4. An Accountability Policy which defines the responsibilities of users, operations staff, and management. It should specify an audit capability, and provide incident handling guidelines (i.e., what to do and who to contact if a possible intrusion is detected).

5. An Authentication Policy which establishes trust through an effective password policy, and by setting guidelines for remote location authentication and the use of authentication devices (e.g., one-time passwords and the devices that generate them).

6. An Availability statement which sets users' expectations for the availability of resources. It should address redundancy and recovery issues, as well as specify operating hours and maintenance down-time periods. It should also include contact information for reporting system and network failures.

7. An Information Technology System & Network Maintenance Policy which de-scribes how both internal and external maintenance people are allowed to handle and access technology. One important topic to be addressed here is whether re-mote maintenance is allowed and how such access is controlled. Another area for consideration here is outsourcing and how it is managed.

8. A Violations Reporting Policy that indicates which types of violations (e.g., privacy and security, internal and external) must be reported and to whom the reports are made. A non-threatening atmosphere and the possibility of anonymous reporting will result in a greater probability that a violation will be reported if it is detected.

9. Supporting Information which provides users, staff, and management with contact information for each type of policy violation; guidelines on how to handle outside queries about a security incident, or information which may be considered confidential or proprietary; and cross-references to security

procedures and related information, such as company policies and governmental laws and regulations.

10. SECURITY POLICY VS SPECIFICATION

Let us highlight that the security policy can be seen as a specification for security solutions. In fact, the security policy forms a precise description of the required behavior of any secured information system or network. It enumerates the security rules stated at the beginning of any securing project with regard to the security requirements de-fined by the organization owning the information system to be secured. Thus, security policies form a conceptual model of the security solutions as the specifications do for software systems.

In addition, and similarly to the software specifications, security policies are modular. In fact, they are split into more than one document, as we have stated earlier in this section, for the same reasons considered in software field, i.e. easy understanding, modification and update.

Despite their similarities, security policies present some differences with software specifications. The main difference consists in the fact that specifications are concerned as a process (following a modeling method such as UML) and a product at the same time, while a security policy is just a product. The remaining differences are summarized in the following issues:

Structure: Specifications define a set of variables (inputs and outputs) and the different methods that will constitute a single software product. However, security policies delimit the boundary of the system to be secured and the actions that are considered in this system. A network analysis is then performed to define the objects and subjects in the information system and the different rules they must abide. It comes without saying that a security policy deals with a set of products. This is due to the heterogeneity of the rules constituting the policy documents.

Nature: Software specifications define methods that can be seen as a set of instructions having in input some variables and providing results or outputs depending on the given inputs. Hence, software programs are considered as a set of serial or parallel instructions. However, in security policies we deal with rules to be applied and security properties to be respected and controlled. Security rules specify what must be done within the objects and roles situated around the delimited boundary. Therefore, security policy can be seen as continuous set of programs since it depends on the state of the system to be secured.

Types of variables, objects and subjects: Variables manipulated in software specifications are essentially predefined types such as integers, reels, floats and strings or constructed types that are composed of the previous ones and types used in graphical interfaces such as buttons, and events. While objects and subjects manipulated in security policies are elements of the information system. They are essentially: network components, locals, documents and roles. These lists are not exhaustive. In fact, equipments list may grow as new equipments arise. Documents and roles may also differ from one organization to and other according to the nature of their activities.

Temporal considerations: Unlike software specifications, security policies involve the concept of time using very special forms often reduced to time stamping and occurrence ordering. In fact, some rules include temporal expressions such that before, after, and periodically. For example, a security rule states that a backup must be performed periodically.

Decisions type during the design phase: Unlike software specifications where a set of instructions are to be implemented, security policies involve complicated decisions. In fact, within information systems, we have to decide about the limit between the secured and non-secured areas. We have also to make the right decisions about where to implement security solutions to fulfill security needs.

SP IMPACT ON THE RISK

The security policy is an important concept that influences risk analysis and that can also be influenced by risk assessment.

SP AND RISK ANALYSIS

As we have seen in the previous section, the security policy is more than a specification for the secured network. Thus, the security policy is not limited to a product. It can be seen as a process having inputs and outputs. Inputs to the security policy are mainly the system assets. Outputs are the security rules to be applied to the system.

In addition, the security policy can be used for analyzing risks as well as it can be influenced by risk analysis. In fact, vulnerability identification, which is one of the activities within risk analysis, is based among others on the security policy. After analyzing the risks, security solutions will be chosen and they will be used to modify the security policy in order to mitigate the identified risks.

Furthermore, the security policy is not static. As we have highlighted earlier in this chapter, a reactive analysis is performed for the secured system to react against security incidents. This means that new security measures are defined to avoid the occurrence of these accidents in the future. Consequently, the security policy is updated to take in consideration these controls.

Moreover, the security policy can be used to deduce the cost of a security project as it forms a full description of the system to be secured. All the assets and the interactions between them are defined in the security policy. In addition, this policy defines the way these assets will be secured. The security policy can then inform us about the complexity of the security tasks that will be performed during the project.

SP BASED ASSESSMENT

A security policy based assessment should follow the policy life cycle [33] depicted by Figure 2.3. The SP life cycle involves six phases explained briefly hereinafter.

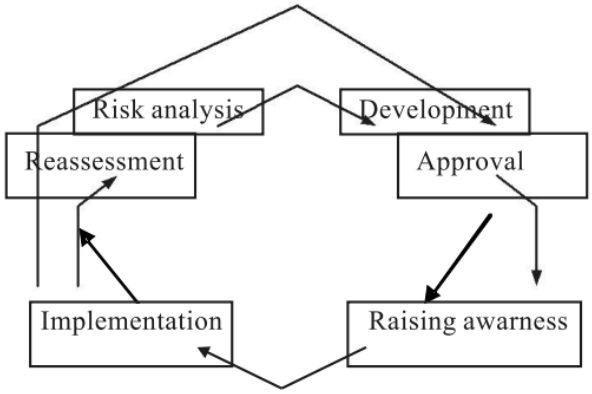

Figure 3.Security Policy life cycle

Risk analysis: It includes essentially a mission statement, asset evaluation, and threat assessment. It is worth mentioning that some parts of the security policy can be written in this step. In fact, the risk analysis needs some rules to assign a security level to each resource, meaning that the data classification policy should have already been constructed at this level.

1. **Development**: This step consists of selecting the security rules that best fit the requirements of the organization. The SP development team must use convenient languages to model and validate the security policy. The main characteristic of this step is that it is performed progressively to move from an abstract represen-tation toward a more concrete one.

2. **Approval**: It relies on a multidisciplinary committee that validates the security policy. At every layer (i.e., abstraction degree) of the development process, the SP should be validated against (a) the upper layer and (b) the security objectives.

3. **Raising awareness**: This ensures that the security policy is accessible to every-one who is authorized to access it. Thus, the SP is published correctly and every user of the secured system must process the skills that are suitable to his or her responsibilities.

4. **Implementation**: It enforces the application of the security policy. During this step, operational and technical controls are put in place. Operational controls are security mechanisms that are essentially

implemented and executed by the users themselves, whereas technical controls include the automated security counter-measures.

5. **Reassessment**: It guarantees a continuous monitoring of the security policy through scheduled revisions and analysis. This process is essential to practically test the efficiency of the SP because new threats occur.

11. CONCLUSION

In this paper, we gave an overview about information risk management. We presented the shortcuts of the existing Risk Management methodologies. We gave also interest to risk analysis approaches, to risk assessment techniques and to security policies and their impact on the risk.

REFERENCESS

1. C. Alberts and A. Dorofee, Managing Information Security Risks: The OCTAVE Approach. Addison Wesley Professional, July 2002.
2. K. Stolen, F. den Braber, T. Dimitrakos, R. Fredriksen, B. A. Gran, S.-H. Houmb, Y. C. Stamatiou, and J. O. Aagedal., Business Component-Based Software En-gineering, ch. Model-based risk assessment in a component-based software engineering process: the CORAS approach to identify security risks, pp. 189–207. Kluwer, 2003.
3. C. S. S. Ltd, "Security risk analysis and ISO 17799 / BS7799 compliance tool." http://www.riskworld.net/.
4. R. Inc. http://www.riskwatch.com.
5. M. Hamdi and N. Boudriga, "Computer security risk management: Theory, challenges, and countermeasures," International Journal of Communication Systems, vol. 18, no. 8, pp. 763–793, 2005.
6. M. Hamdi and N. Boudrgia, "Computer network security risk management: A survey," in Jordan International Conference on Computer Science and Engineer-ing, (Jordan), October 2004.
7. T. R. Peltier, Information Security Risk Analysis. AUERBACH, 1st ed., 2001.
8. B. A. Fessi, M. Hamdi, S. Benabdallah, and N. Boudriga, "A decisional framework system for computer network intrusion detection," European Journal of Operational Research, vol. 177, pp. 1824–1838, 2007.
9. M. Hamdi, N. Boudriga, and M. S. Obaidat, Handbook of Information Security, vol. 3, ch. Security Policy Guidelines, pp. 945–959. John Wiley & Sons, Inc, 2006.
10. E. Verzuh, The Portable MBA in Project Management, ch. Project management is a strategic strength, pp. 5–25. John Wiley & Sons, Inc., 1 ed., 2003.
11. H. Kerzner, Strategic Planning for Project Management using A Project Manage-ment Maturity Model. John Wiley & Sons, Inc., 2001.
12. K. Heldman, PMP: Project Management Professional Study Guide. SYBEX Inc., 2002.
13. P. M. Institute, A Guide to the Project Management Body of Knowledge: PMBOK Guide. Project Management Institute, 3rd ed., 2004.
14. R. L. Kliem and I. S. Ludin, Project Management Practitioner's Handbook. AMA-COM, 1998.
15. Harvard Business School, Project Management Manual, October 1997. 9-697-034.
16. Project management: Guide to project management. No. BS6079-1:2002, British Standards Institute, May 2002.
17. I. O. for Standardization, ISO 10006:2003 Quality management systems - Guide-lines for quality management in projects. June 2003.

18. I. O. for Standardization, ISO 10007:2003 Quality management systems - Guide-lines for configuration management. 2003.
19. V. Temnenco, "Software estimation, enterprise-wide: Reasons and means." IBM developer Works, June 15 2007.
20. H. Leung and Z. Fan, Handbook of Software Engineering and Knowledge Engineering, vol. II, ch. Software Cost Estimation. 2001.
21. B. Boehm, C. Abts, and S. Chulani, "Software development cost estimation approaches - a survey," Tech. Rep. USC-CSE-2000-505, USC Center for Software Engineering, April 10 2000.
22. R. Smith and L. Edwards, "Cocomo- scorm: Interactive courseware project cost modeling," in Proceedings of International Council of Systems Engineering Con-ference, 2006.
23. B. Boehm, R. Valerdi, J. A. Lane, and A. W. Brown, "Cocomo suite methodology and evolution," CROSSTALK The Journal of Defense Software Engineering, pp. 20–25, 2005.
24. C. A. I. (CAI), "Focus on lawrence putnam: A cai state of the practice interview," IT Metrics and Productivity Journal, vol. Special Edition, pp. 1–12, September 2006.
25. O. Marban, E. Menasalvas, and C. Fernandez-Baizan, "A cost model to esti-mate the effort of data mining projects (dmcomo)," Information Systems ournal, vol. 33, pp. 133 – 150, 2008.
26. B. W. Boehm, C. Abts, A. W. Brown, S. Chulani, B. K. Clark, E. Horowitz, R. Madachy, D. Reifer, and B. Steece, Software Cost Estimation with COCOMO II. Prentice Hall, 2000.
27. T. E. Hastings and A. Sajeev, "A vector-based approach to software size measurement and effort estimation," IEEE Transaction on Software Engineering, vol. 27, no. 4, pp. 337–350, 2001.

