ISSN (ONLINE): 2349-7084
GLOBAL IMPACT FACTOR 0.238
ISRA JIF 0.351

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 4, APRIL 2015, PP 280-295

Network Risk Analysis Model for Risk Management

Naveen Kumar R , G. Ravindra Babu

Abstract— In this paper a risk management framework, called NetRAM (Network Risk Analysis Method) has been developed. A key characteristic of NetRAM is that it is heterogeneous, meaning that it integrates different components like software tools, architectural design methodologies and theoretical models.

Index Terms—Management Framework, Network Risk Analysis Method, Computer – aided Risk Analysis

1. INTRODUCTION

1.1 Main Features of NetRam

1. NetRAM is adapted to different modern enterprise structures at different levels. In fact, it is possible to handle various network architectures and topologies. Moreover, NetRAM can be customized so that the business activity of the enterprise is taken into consideration. Therefore, NetRAM instills a big importance to the risk management process at the enterprise level as it would appear as a focal activity.
2. NetRAM includes security project management models helping managers planning and controlling security projects. These models can handle all categories of security projects in organizations owning small or large networks and having low or high technology levels.
3. NetRAM ensures an optimal handling of the risk management triad: analysis, decision, and response. Many sources provide necessary data and execute various operations. For instance, information contained in Intrusion Detection Systems (IDSs) and automated vulnerability scanners can be imported and analyzed as it will be shown in the following sections. Likewise, a group of experts might be involved at some stages of the decision making process and at the incident response process. Consequently, NetRAM can be viewed as a collaborative and a collective framework.
4. NetRAM uses appropriate formalisms allowing to handle appropriately the un-certainly introduced by different factors. Both Computer-Aided Risk Analysis (CARA) and expert-based decision making are performed through the use of structured methodologies that reduce the error-rate and prevent it from propagating across the risk management steps.
5. Security policy definition is a delicate activity as the network security relies closely on security rules defined in this document. NetRAM ensures the efficiency of the defined security policy through mechanisms of validation and test based on a formal representation of the security policy.
6. Monitoring the security of the analyzed Information Technology (IT) infrastructure is a crucial task. A set of modules ensuring a continuous control of the system state has been integrated with the NetRAM framework in [8]. Detect-ing deviations from the normal behavior and conducting convenient reactions are the main

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

objectives of these modules.

7. NetRAM ensures an optimal response to security incidents. In fact, formal systems and security experts are involved in this module to analyze the occurred events, trace-back the actions performed by attackers and select the security counter measures addressing the malicious activities. Moreover, NetRAM ensures the detection and response of new attacks based on the collaboration of experts and the formal tools.
8. NetRAM is highly adapted to the rapid changes of the IT world that might affect the analyzed system. A learning process is thus developed to limit human intervention during the update of several quantitative parameters and semantic links. Effectively, this process is semi-automatic as human experts must continue to participate to it.

2. NETRAM LIFE-CYCLE

NetRAM life cycle is depicted by Figure 1. The interested reader may remark that it consists of two concurrent graphs.

The graph designed with solid lines represents the workflow of the risk management activities. The arrows model the precedence relation between the ten processes of NetRAM , which follows the progress of the risk management activities. Each bullet represents the task that constitutes a part of the activity necessary to conduct the totality of the risk management process. More precisely, if two sub processes are linked by a solid arrow, the output of the first sub process is an input for the second. On the other hand, the second graph, represented by dashed lines, illustrates the retroactivity existing between some processes of NetRAM . It is worth mentioning that this characteristic is specific to NetRAM. In fact, existing risk assessment methods often subordinate the importance of re existing some risk management activities further to the emergence of some vulnerabilities or attacks threatening the information systems.

As far as NetRAM processes are concerned, it is worth to mention here that they are characterized by their completeness. In fact, with NetRAM we consider the totality of the risk management activities. As the reader can see, the life-cycle contains the smallest but important details that must be considered while managing risks. We list here the initialization, monitoring and incident response processes that are generally ignored by the most popular risk assessment methods. Hereinafter we give an overview of the ten processes.

Initialization: This process aims to prepare the risk management project by providing the managers with the cost of the project and the schedule of the different

Asset analysis: The goal of this process is to gather information about the analyzed system.

Vulnerability identification: Weaknesses and security breaches of the analyzed system are identified at this level.

Threat identification: This process aims to identify the attacks that threaten the analyzed system.

Risk analysis: The aim of this process is to identify the risks that may threaten the assets of analyzed system based on the identified vulnerabilities and threats.

Countermeasure proposal: A security strategy mitigating the identified risks is proposed at this level.

Countermeasure selection: This process aims to define the security policy that will be adapted by the analyzed system to mitigate security risks.

Implementation: At this level, selected security countermeasures are implemented according to the security policy.

Monitoring: The aim of this process is to maintain the analyzed system in an acceptable security level. Monitoring activity can result in the re-execution of some processes if needed.

Incident response: This process aims to react to security intrusions according to the incident response plan.

Details about NetRAM processes are given in the next sub-sections.

INITIALIZATION

A prediction step is first performed to estimate the time, the effort, and the budget that would be needed for the whole risk management project. This implies a quantification of the complexity of the target system using input parameters that are defined according to the

estimation method. Complexity estimation is based on the security policy defined for the secured system. The security policy forms a solid basis as it provides a complete specification of this system and the manner it is secured. The resulting data are very useful in the case of an outsourced or an in-house information security risk assessment as they provide a good basis for the mission chiefs to plan the activities and to take decisions such as those concerning the number of persons who will conduct the mission or the amount of money that should be allocated for it.

A security cost evaluation model, called SECOMO [64, 65, 66] (SEcurity COst MOdel), has been developed to assist security analysts and high-level managers during this crucial process.

A second problem that arises at this level is activity scheduling. Knowing the complexity and the time constraints (expressed as deadlines) related to a project, an appropriate distribution of the required amount of effort on the available time periods should be performed. This can be viewed as a multi-objective problem that has been discussed in [67]. Effectively, in addition to the trivial components of the multi-objective utility function, several business-oriented functions that are specific to the studied context can be used to find the optimal time schedule.

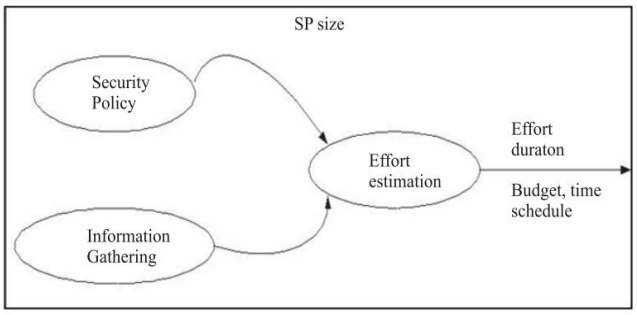

Figure 1. The OCTAVE Method

3. ASSET, VULNERABILITY, THREAT AND RISK ANALYSIS

We discuss here the main modules involved in the risk assessment activity, i.e. asset, vulnerability, threat and risk analysis. Mainly, asset analysis consists in collecting in-formation about the assets within the system to be protected. The interaction between these assets is considered in this module. Then, vulnerabilities are identified based on existing automated tools, questionnaires and expert opinion. After this, potential threats against the assets are identified based on the identified vulnerabilities. After asset, vulnerability and threat analysis, we perform risk analysis to identify the most important attack scenarios and risks.

ASSET ANALYSIS

Asset analysis is designed to gather detailed information about the various assets that make part of the protected system. Indeed, an inventory containing all the resources must be established including some parameters such as the criticality of each asset or the objects that are authorized to access it. Furthermore, knowing that the components of the analyzed system are - in most cases - interrelated, the interaction between the resources given in the inventory is a point of interest. For instance, issues as data or in-formation flow between the different entities may be focused. Also, physical interaction has to be analyzed as the security of an asset can depend on the security of another asset that contains it physically (the security of an equipment put in a room depends on the security level given by the walls, the doors and the windows of the room itself). Thus, dependency trees can be built to show the interrelation between the resources of the system to facilitate the risk analysis process that will be discussed later. It is worth to mention that the documents related to security (security strategy, security policy, etc.) are considered as special assets.

VULNERABILITY ANALYSIS

The purpose of vulnerability analysis is to identify the weaknesses of the system de-scribed in the former

step. The recommended approach is to have a vulnerability library and to check, for every vulnerability, if it is present or not in the studied case. Three detection methods are considered (automated scanners, questionnaires, and ex-pert opinions). Each one is adapted to particular classes as shown in the following:

Automated scanners: Scanners are pieces of software that contain pre-defined test lists that can be automatically (and remotely) executed to evaluate the security of a networked system. In most cases, penetration testing should be conducted to confirm (or negate) the existence of an identified vulnerability. They are particularly efficient in detecting:

– **Operating Systems (OSs) and application bugs:** Most of the software used in networked environment contain security holes that can be exploited by malicious entities,

– **Conceptual vulnerabilities:** The theoretical specifications of several widely used communication and security protocols contain vulnerabilities that have to be focused when analyzing the target system. One of the most famous vulnerabilities of this class is relative to SMTP (Simple Mail Transfer Protocol) which does not authenticate the source of an e-mail,

– **Misconfiguration:** Lack of experience or insufficient training of the staff opens many breaches in the system. For instance, a mistake in the policy of a firewall can allow unauthorized users to gain access to a private network.

Questionnaires: A list of questions is asked to each user so that weaknesses can be inferred from the submitted answers. Questions are structured into sections to confer an adaptive aspect to the questionnaire meaning that it is adapted to various asset types (e.g., web server, firewall, router). This mechanism is essentially used to check for:

– **Behavioral vulnerabilities:** Users may not apply appropriately the security policy resulting in a insecure interaction with the system.

– **Misconfiguration:** Several misconfigurations cannot be detected by auto-mated scanning tools. Therefore, a complementary testing procedure can be performed through the use of questionnaires.

Expert opinions: Inadequate procedures or inappropriate security measures can be exploited to perform malicious acts. An example of this type of vulnerability could be the absence of a backup policy which can result in an unrecoverable loss of data. Hence, the security documentation should be subjected to a deep analysis by a group of security experts. The main problem that may be faced at this level resides in conflicting beliefs (experts may reason differently about a problem). A three-round voting mechanism has been developed to resolve it. It allows the experts to select vulnerabilities from the "vulnerability library" and to update this latter if new kinds of weaknesses appear in the security documents.

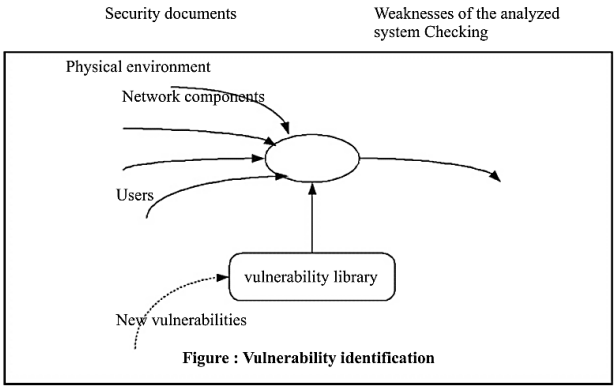

Figure 2. Vulnerability Identification

4. THREAT ANALYSIS

Threats are potential events that can affect the analyzed system. They can result from malicious actions, accidents, natural disasters, etc. Unlike

vulnerabilities, threats are measurable as each of them can be represented by its frequency and its severity. Obviously, threat rates and impacts depend on the environment in which the analyzed system is situated. Factors as geographical position, political stance or activity sector have to be taken in consideration when allocating probabilities of occurrence to threats. Practically, numerical values corresponding to those parameters are assigned by experts. However, due to their prominent importance, the frequency and the severity of a threat are dynamically updated according to the values of several metrics measured during the monitoring process as shown in Figure 4.4 (dashed arrows). A learning mechanism included in the monitoring step allows the metrics (i.e., probability, impact) to be updated according the values taken from the real context. In addition, it adds new threats to the library whenever new types of attacks are identified.

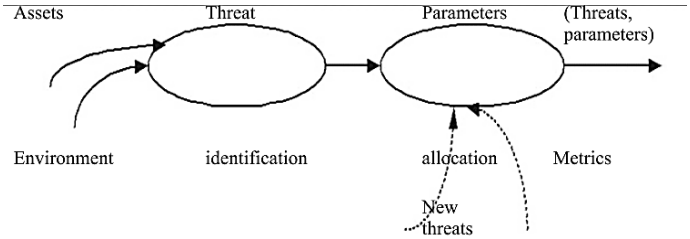

Figure 3. Threat Identification

5. RISK ANALYSIS

We define an attack as a combination of a threat and a set of corresponding vulnerabilities. A risk can then be seen as a weighted attack. Many weights can be allocated to a single attack where each weight corresponds to a criterion. The probability of the attack, its technical difficulty or the amount of money needed to carry it out may constitute a good criterion. Then, as an extension to this reasoning, we introduce the concept of attack scenarios which may be viewed as attack chains where the last link is called the main attack, the first link is an elementary attack while the other links are intermediate attacks. This approach expresses accurately the occurrence of real attacks where a malicious user performs a set of intermediate attacks in order to achieve his major goal: the main attack. Furthermore, as a weight can be assigned to each attack, we can conclude that a global weight may be allocated to an attack scenario by com-bining the elementary weights corresponding to each attack of the scenario. Weighted attacks are then called risk scenarios.

The aim of the risk analysis process is to define the main risks corresponding to each asset and to establish the risk scenarios leading to every main risk. Indeed, this process can be divided into the following steps:

STEP 1) Identify the global (main) attacks corresponding to the asset;

STEP 2) Build the attack scenarios for every main attack;

STEP 3) Determine the risk coefficients (weights) for each scenario: a weight corresponding to a scenario is computed by combining the weights of the elementary attacks belonging to it.

New attack scenarios should be automatically appended to the existing ones whenever the occurrence of a new attack chain is detected during the monitoring step.

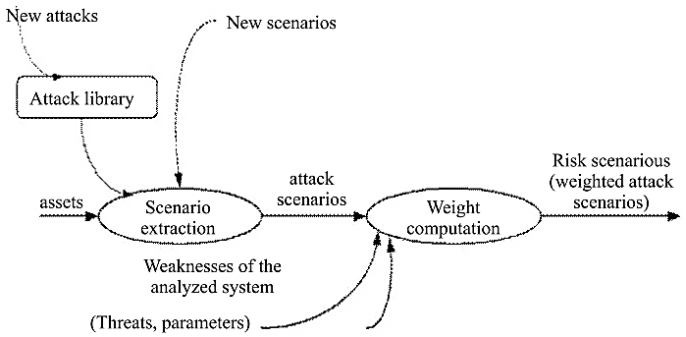

Figure 4. Risk Analysis

6. COUNTERMEASURE MANAGEMENT

Countermeasure management involves solution proposal, selection and update during time. Using the identified risks, we extract the potential

countermeasures from the predefined list. A sub-set of the proposed countermeasures is selected based on several parameters, including their efficiency, cost, implementation effort, etc.

The selected countermeasures can be updated as long as the organization is active upon the detection of security incidents. During the incident response process -which aims at collecting events, identifying the incidents and responding to them- counter-measures are adapted to respond to security incidents.

6.1 Counter-Measure Proposal

Possible risk scenarios issued from the former step can be ranked and prioritized. Then, for every scenario, a set of security rules are defined to minimize its priority. If the scenarios are ranked according to their probability of occurrence, the goal of the rule should be to minimize this probability. These rules must be general and must not include issues related to the effective implementation (e.g., practices to follow, products to acquire, technical standards to comply with). The obtained set of rules constitutes the security strategy of the analyzed system.

Practically, the selection of the security solutions that will be candidate to the optimization process discussed in the next step is done through the use of heuristics that allows limiting the search space of the decision maker. A couple of examples of such heuristics are given in the following:

If a decision mitigates one attribute of an attack (e.g., probability, impact) that has been found to be possible to carry out against the analysed system, then this decision is candidate to the selection process.

If a decision mitigates one attribute of an attack that belongs to a given scenario, then the decision reduces the same parameters as the whole scenario.

In addition, a learning mechanism is implemented to update the link between the scenario and the decision library. Initially, this link is built by a group of experts through a collective decision process.

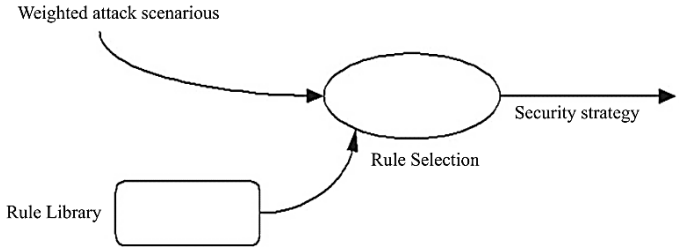

Figure 5. Counter measurement

6.2 Countermeasure Selection

A set of candidate risk control techniques are proposed to implement the rules of the security strategy. Then, according to several criteria, the actions that would be taken to protect the system have to be clearly described in a document that is called the security policy. These criteria may include:

1. The efficiency or the degree of protection given by the technique (i.e., the influence of a decision on an attack can be measured in terms of probability and impact rates),
2. The cost of the security solution,
3. The criticality of the asset concerned by the risk
4. The feasibility of the countermeasure.

The alert reader would have noticed that multiple factors should be taken into account when selecting the appropriate security solutions. Even worse, most of those factors do not have the same nature. For instance, the loss resulting from a given attack can be tangible (e.g., amount of money) or intangible (e.g., impact on enterprise reputation). Therefore, an ordering allowing comparing and ranking security solutions is established on the basis of these criteria. In point of fact, decisions are shown to have a lattice structure which is exploited in the selection process. Several counter-measures can even be omitted if the security level they allow to attain is not proportionate to their costs.

Obviously, comparison criteria can be customized to fit with various system requirements. Hence, context-

specific evaluators that reflect the business objectives of a given enterprise can be easily integrated into the optimization task. Within NetRAM , users (i.e., high-level managers and security administrators) are able to fix their own risk management goals so that the resulting strategy be closer to their context.

Moreover, a plan defining the comparison of the retained actions with respect to the corresponding level of risk and to the allocated budget must be done during this process. This means that an activity schedule has to be fixed to guarantee an optimal benefit to the enterprise.

The resulting security policy is validated and tested to avoid problems of inconsistency and incompatibility with security requirements. Within NetRAM , executable security policies written in a formal language are used to check if the policy does not present conflicts with security requirements. In addition, tests are automatically gen-erated based on the formal representation of the security policy. These tests are used to check if the implemented solutions are conform to the security requirements.

Likewise, an incident response plan is also performed at this level. It defines the actions that might be taken when a security incident occurs. These actions should include procedures for the notification and the documentation of security incidents as well as a description of the recovery mechanisms.

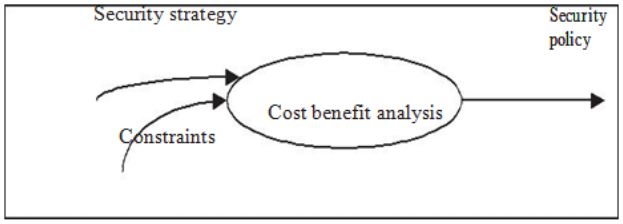

Figure 6. Countermeasure selection

6.3 Incident Response

Whenever an anomaly is detected at the monitoring level, the system should execute appropriate reactions. The incident response plan defines the constituency and the role of the incident response teams as well as the actions they have to take if an anomaly do occur. These actions cover essentially four topics:

Notification: Who should be not notified? Which mean of communication should be used?

Impact attenuation: What should be done in order to reduce the impact of the incident and to stop its propagation through the analyzed system?

Recovery: In case of damage, what should be done to ensure a recovery of the essential functions of the target system? Documentation: Which part(s) of the information related to the incident has to be archived?

With regard to the sensitivity of the decisions that would be selected during incident response, this process cannot be fully automated. For example, stopping a service for few hours can result in a so big amount of loss that convincing justifications should be provided. The particularity of incident response is that it is a collective process, meaning that a group of experts participate to it. NetRAM integrates the concept of Incident Response Probabilistic Cognitive Maps (IRPCMs) to handle this situation. Typically, an IRPCM consists of a set of nodes and a set of weighted edges. The formers represent concepts (e.g., symptom, action, unauthorized result) while the latter are wighted links (e.g., (< , 0.7), (/ , 0.9)). IRPCM-based approaches have been evolved to translate the reasoning and the views of a group of experts about a given problem in order to extract a single opinion that would lead to a more efficient interpretation.

Moreover, NetRAM involves digital investigation to construct the attack scenarios followed by the attackers. A formal logic, referred to as Investigation Temporal Logic of Actions (I-TLA), is used to achieve this goal.

Adapting this concept to our case results in the three following phases:

1. Evidence collection and IRPCM building: Data relative to the incident or evidences are collected from different sources within the network. Based on these evidences, incident response team members build an incident response probabilistic cognitive map.

2. Reasoning and attack scenarios construction: The built IRPCM is used by a for-mal language and its model checker to automatically extract the attack scenarios followed by the attackers. Incompleteness of details regarding the investigated incident and the investigator knowledge may be detected at this level. This is re-solved using the concept of hypotheses. Hypothetical actions are then generated, appended to the scenarios under construction, and efficiently managed during the whole process of construction.

3. Decision making: The final phase consists in selecting the convenient security solutions using the IRPCM and knowing the cost and the benefit of each alter-native. We adopt a multi-attribute quantitative optimization scheme to compare and rank candidate countermeasures. This comparison is based on the cost and the benefit that would result from the application of each alternative.

6.4 Implementation

During this phase, the actions defined in the previous process are effectively implemented. The key operations of this process are:

Acquiring the needed hardware and software specified in the security policy, Defining the teams that will be in charge of securing and monitoring the system, Promote security awareness, En force the application of the security practices,

Implement the technical operations so that the system becomes compliant with the security strategy and with the security policy.

Whereas it is of utmost importance, most of the existing risk management approaches do not discuss deeply the implementation related issues. It may seem that this activity consists in a simple application of the security solutions found at the previous process. However, implementation is much more complicated as two important issues, at least, should be considered:

An appropriate methodology should be followed when applying security counter-measures. In fact, previously implemented security mechanisms should be particularly taken into consideration since their interaction with new ones can lead to unpredicted results. Moreover, installing security solutions would not perturb the normal functioning of the system. In fact, drastic security measures would be a source of denial of service.

An engineering activity has to be developed around the activities of this process to ensure a maximum efficiency. It should be guaranteed that the selected counter-measures are implemented such that they effectively mitigate the risk threatening the target information system. To this purpose, suitable testing strategies should be evolved and applied to check whether a countermeasure is implemented appropriately. Of course, a consistent part of these test are automated to make the process less time-consuming.

In addition, implementation constitutes the core of a learning module that permits to update several parameters of the decision selection process discussed above. For example, the influence of a decision on a given attack can be effectively assessed through the execution of a set of simulated attacks and the use of real traffic. Obviously, refining factors that are used in the decision making process improves considerably the countermeasure proposal and selection processes.

6.5 Monitoring

Monitoring is the central process of NetRAM . It can be seen as the detection of events that change several properties of the analyzed system such as security

incidents, the addition of a new asset or the appearance of a new vulnerability. This module has two main characteristics: it is continuous and retrospective. In fact, monitoring must be done, from the moment of the implementation of the first action of risk control, in a uninterrupted manner to ensure an efficient detection of the interesting events. In addition, it can be seen as a trigger that launches other processes which were already performed such as vulnerability identification in the case of the appearance of a new vulnerability or counter-measure selection if a technological innovation allows the implementation of a solution that were impossible at the time of the initial application of the framework.

A reoccurring theme when speaking about monitoring is intrusion detection, which is naturally the kernel of this process. Many factors can be adopted to evaluate an IDS but efficiency counts for most and reducing the rate of false decisions is the primary objective. To this end, a three-fold intrusion detection activity is being investigated within the frame of NetRAM :

Alert representation: Alerts are notifications of the occurrence of attacks and therefore a strong link should be built between those two fundamental concepts. Especially, the consecutive occurrence of a set of attacks (i.e., attack scenario) should be inferred from the generation of several consecutive alerts.

2. Correlation: The ability NetRAM 's monitoring module has to interact with more than one IDS product enhances the quality of the final decision resulting from the combination of the elementary decisions (originating from elementary IDSs). The main difficulty at this level is to state whether two alerts coming from two different IDSs, possibly considering different kinds of parameters, correspond to the same event. For instance, a change in a system file (detected through TRIPWIRE) can be linked to a malicious packet (detected by SNORT). To cope with this problem, a correlation coefficient between each couple of alerts is computed. If this coefficient exceeds a given threshold value, the decision sys-tem concludes that the alerts are correlated. To reduce the complexity of the correlation coefficient computation, an alert classification can be performed prior to correlation.

Learning: Decision making factors used by the IDS at different levels rely on the measurement and the interpretation of several metrics that translate efficiently the state of the target system. As the values of those metrics change across time, a learning mechanism allowing an appropriate update is developed. The heterogeneity of the monitored parameters requires the use of various learning approaches (e.g., supervised, non-supervised, reinforcement).

Furthermore, an interface with the IDS has been designed so that the alert database can be taken into account in the monitoring activity. For instance, IDS log files can help to estimate the probability of occurrence of each attack.

7. THE INITIALIZATION PROCESS

Before executing a security project, the latter needs to be managed. Management of security projects involves the estimation of its cost and the scheduling of its activities.

8. SECURITY EFFORT ESTIMATION APPROACH AND TECHNIQUE

As effort and cost estimation approaches are considered, we have chosen to use the top-down approach because effort and cost estimation is performed at the starting of the security project where the necessary data may not be all available. In addition, this approach does not require additional effort and is easier to use than the bottom-up approach.

As cost estimation techniques are considered, we have chosen to use a hybrid cost estimation technique combining the model-based, expertise-based and regression-based techniques. The first category of techniques is objective and provides accurate esti-mates. In addition, the estimates can be easily documented with the model-based technique and hence can be

reused in the future. Moreover, it is possible to refine and customize the formulas of the model to fit an appropriate environment. Expertise is required to initialize some parameters at the first stages of the model as at the beginning, we lack of the necessary data relative to security projects. Regression is used to initialize the model and refine it as new security projects are performed and data about them are documented.

To estimate effort relative to security projects, we have developed a Security Cost Model (SECOMO) based on COCOMO II. The choice of COCOMO II is argued by several reasons: (1) it is well documented, available in the public domain and supported by public domain and commercial tools, (2) it has been widely used and evaluated in a range of organizations, and (3) it has a long pedigree from its first instantiation in 1981 [53], to its most recent version, COCOMO II, published in 2000 [51]. In addition, COCOMO II uses the top-down cost estimation approach and a model-based cost estimation technique combined with the expertise-based and the regression based techniques.

9. PARAMETERS AFFCTING THE EFFORT

Generally, effort estimation depends on several parameters. The size of the secured network is the main parameter in estimating security projects effort. Network size estimation is a delicate activity because network size is more than computing its elements (e.g., computers, switches, routers, etc). These components must also be considered when estimating the network size. Network size includes also the interaction between these components as they are linked to each other. Interaction between network components makes it difficult the network size estimation.

In addition, networks generally include security elements such as firewalls, intrusion detection systems, anti-viruses and cryptographic systems. These elements must be considered when estimating the network size because they have a great effect on its complexity.

The second category of factors affecting the effort estimation is those that are related to the security project running. They are called effort multipliers. Within security projects, we consider four categories of these factors. The first category is related to the security solution resulting from the project. For example, the reuse of some results of the solution during the project means that we must document these results before the termination of the project which needs additional effort.

The second category of factors is related to the team involved in the project. For example the experience has a great impact on the effort. Experienced persons will take less time than those having less experience in performing security activities. Experience is a wide term involving the habit of the person in performing the security activity, in using security tools and in dealing with the platform used in the secured system.

The third category of factors affecting the effort is related to the project details. Security activities can be more or less complex according to the project contract. For example, active vulnerability analysis is more complex than passive analysis; it involves more activities and needs more time. Thus effort required for active vulnerability analysis is greater than effort needed for passive vulnerability analysis.

The reader may notice that the three above mentioned categories of effort multipliers are already used in COCOMO II. The difference in considering them for security projects appears in the significance of the parameters they involve. In fact, effort multipliers should be appropriate to security field in our case. Nevertheless, we introduce here a new category of effort multipliers related to the secured information system. The information system is the core of the security project. It appears in its inputs and outputs at the same time. Some characteristics of this system affect the effort of securing it. For example, the attacks that have targeted this system in the near past may inform about the current security of this system. Systems poorly secured need greater effort for running the current project.

In addition to the effort multipliers, some other parameters called scale factors affect the effert estimation. These factors are related to the scale effect. An economy or diseconomy of scale may characterize the model according to the values of these parameters. Economy of scale occurs when the cost per output unit decreases with the output number, while diseconomy of scale occurs when this cost increases.

10. SCHEDULING

Within NetRAM , the problem of scheduling involves the planning of the activities and the allocation of resources under several constraints and in an optimal way. The classic scheduling techniques are not appropriate to this problematic. Especially, the Gantt chart is not applicable as security projects fall in the category of complex projects. In fact, security projects involve numerous tasks that have complex relationships and numerous resources including human and non-human ones.

At first blush, PERT and CPM techniques seem to be applicable to our case as they are created to deal with complex projects. However, these techniques are also inadequate to our problem because the aim is not reduced at calculating a schedule. In fact, the schedule must be the optimal one. Optimality is determined through the use of multiple criteria which must be maximized or minimized according to their nature. Examples of these criteria are the time spent idle by the resources, the cost of the human resources. The criteria may also concern the client for whom we are performing the security project. One example of such criteria is the financial losses due to the stopping of some services at the client site.

The scheduling and allocation problem of security projects can then be seen as a multi-objective optimization problem. This category of problems is generally resolved using the genetic algorithms. The latter have shown a great effectiveness to solve non linear and combinatorial problems where several objectives have to be reached. They are also known for scheduling large since they operate on a population of solutions rather than a single solution. This makes them the most appropriate for solving scheduling problems where the space of possible solutions is very large. Genetic algorithms from a group of methods that apply the principles of natural selection and evolution to solve hard optimization problems [68, 69].

Over many years of research, several multi-objective optimization techniques using evolutionary algorithms were suggested. These algorithms present, however, some disadvantages including high computational complexity of non-dominated sorting, lack of elitism, and need for specifying the sharing parameter [70]. The elitist non-dominated sorting genetic algorithm called NSGA-II has been designed to palliate these disadvantages. The details of NSGA-II appear in [70].

For scheduling security projects, we have chosen to use the NSGA-II algorithm for the reasons stated above. However we have found that this algorithm needs some modifications to be appropriate to security projects scheduling problems. NSGA-II shortcuts with regard to security projects and the modifications that we have added to it are presented in details in Chapter 6.

11. SECURITY POLICY EFFCTIVENESS

Traditionally, security policies are hard-coded [71]. This can cause major problems including the existence of errors in the policy causing security problems to the secured system (for instance the policy may contain two conflicting rules resulting in the creation of a breach that can be exploited by attackers), the difficulty in implementing the policy (for instance, if a policy states that “passwords must be at least 6-character long”, there must exist a snippet of code in the system implementation that checks the length of passwords) and the difficulty in managing it (hard-coded policies do not separate the policy specification from the system implementation). In addition, hard-coded policies are expansive to implement and change. It is then important to find a representation of security policies facilitating

their validation, management and implementation and enabling cost effectiveness of these policies.

12. SECURITY POLICY EFFCIENCY

Security policies may contain errors such as human errors or errors in expressing the defined needs. An example of errors always encountered is the existence of conflicting rules. To avoid the occurrence of such errors, security policies have to be proved. Verification, validation and test are the tools used for proving systems efficiency. Generally, Verification and Validation (V&V) is the process of checking that a product, service, or system meets specifications and that it fulfils its intended purpose. We define here the V&V process with regard to Security policies.

Security policy verification is a process that is used to evaluate whether or not the SP complies with the security properties stated at the beginning of the security project. Formally speaking, verification is the act of proving or disproving the correctness of in-tended SPs securing a system with respect to a certain formal specification or property, using formal methods of mathematics. There are roughly two approaches to formal verification, namely model checking and logical inference. Within NetRAM , we use the model checking approach, which consists of a systematically exhaustive exploration of the mathematical model (this is possible for finite models, but also for some infinite models where infinite sets of states can be effectively represented). Usually this consists of exploring all states and transitions in the model, by using smart and domain-specific abstraction techniques to consider whole groups of states in a single operation and reduce computing time. Implementation techniques include state space enumeration, symbolic state space enumeration, abstract interpretation, symbolic simulation, abstraction refinement.

Security policy validation is the process of establishing documented evidence that provides a high degree of assurance that the security policy accomplishes its intended requirements. In other words, the security policy is validated by checking if the policy matches the security needs.It is sometimes said that validation ensures that “we are building the right policy” and verification ensures that “we are building it right”. “Building the right policy” refers back to the organizations needs in terms of security; while “building it right” checks that the specification was followed.

Unfortunately, validation is not sufficient to state that a SP implementation is correct with regard to the basic security requirements; another key consideration is to test the conformance of each specification to the one it was derived from. This feature is useful to check whether or not the implementation of a SP fulfils the criteria defined by its specification (which is supposed to be valid). Generating test sets constitutes an important step in SP engineering because it permits to detect the security properties that are not fulfilled by the developed solution. In addition, the testing processing allows to compare a set of candidate SPs with respect to their conformance to the target ideal solution.

13. IMPLEMENTATION COST OF SECURITY POLICIES

The amount of money spent on security should match the risks associated with a potential breach of security (e.g., a financial firm has a higher risk profile than a paper supply company). However, companies must both assess their risk and decide on a reasonable level of protection. A company can spend a lot of money on security, but at some point its ROSI diminishes because the company is outspending her risk.

Before implementing security policies, the IT Engineering team should be able to determine how to implement each countermeasure or control and to provide reasonably accurate estimates on how much acquiring, implementing, and maintaining each one would cost. When the team creates these estimates, it should consider all of the following direct and indirect expenditures that might be associated with a control [72].

Acquisition costs: These costs comprise the software, hardware, and services related to a proposed new control. Some controls may have no acquisition costs. For instance, implementing a new control may merely involve enabling a previously unused feature on a piece of network hardware that the organization is already using. Other controls may require the purchase with application layer filtering capabilities. Some controls may not require the purchase of anything but rather the hiring of a third-party organization. For example, an organization might hire another firm to provide it with a block list of known spammers that is updated daily so that it can tie the list into its spam filters already installed on mail servers in the organization. There may be other controls that the organization chooses to develop itself; all of the costs relating to designing, developing, and testing the controls would be part of an organization's acquisition costs.

Set-up and/or configuration costs: These expenditures relate to start or consultants who will install and configure the proposed new control. Some controls may require a large team to specify, design, test, and deploy properly. Alternatively, a knowledgeable systems administrator could disable a few unused system services on all desktop and mobile computers in only a few minutes if the organization already has enterprise management tools deployed.

Ongoing Costs: These costs relate to continuing activities associated with the new control, such as management, monitoring, and maintenance. They may seem particularly hard to estimate. A simple way consists in estimating them in terms of how many people will need to be involved and how much time each week (or month or year) will need to be spent on these tasks.

Communication Costs: This expenditure is related to communicating new policies or procedures to users. For an organization with a few hundred employees that is installing electronic locks for its server room, a few e-mails sent to the IT staff and senior managers might be sufficient. But any organization deploying smart cards, for example, will require a lot of communication before, during, and after the distribution of smart cards and readers, because users will have to learn a whole new way of logging on to their computers and will undoubtedly encounter a wide range of new or unexpected situations.

Training Costs for IT State: These costs are associated with the IT state that would need to implement, manage, monitor, and maintain the new control. Consider the previous example of an organization that has decided to deploy smart cards. Various teams within the IT organization will have different responsibilities and, therefore, require different types of training. Help desk state will have to know how to help end users overcome common problems such as damaged cards or readers and forgotten PINs. Desktop support state will have to know how to install, troubleshoot, diagnose, and replace the smart card readers. A team within the IT organization will have to be responsible for provisioning new and replacement cards and retrieving cards from departing employees.

Training Costs for Users: This expenditure is related to users who would have to incorporate new behaviour in order to work with the new control. In the smart card scenario referenced previously, all users will have to understand how to use the smart cards and readers.

Costs to Productivity and Convenience: These expenditures are associated with users whose work would be impacted by the new control. In the smart card scenario, if critical business applications, like tools used to manage confidential employee information, or the customer relationship management software used throughout the organization to track important data for all customers, are not compatible with smart cards and are configured to require user authentication, the organization may be faced with some difficult choices. It could upgrade the software, which would require even more costs in terms of new licenses, deployment, and training. Or it could disable the authentication features, but that would lower security significantly. It could, alternatively, require users to enter user names and

passwords when accessing these applications, but then users would once again have to remember passwords, undermining one of the key benefits of smart cards.

Costs for Auditing and Verifying Effectiveness: An organization would incur these expenditures after implementing the proposed new control. Examples of questions that can be asked to further define these costs include:

How will it ensure that the control is actually doing what it was supposed to do? Will some members of the IT organization perform penetration testing?

Will they try running samples of malicious code against the asset that the control is supposed to protect? After the effectiveness of the control has been validated, how will the organization verify that the control is still in place, on an ongoing basis?

14. ALGEBRAIC SPECIFICATION OF SECURITY POLICIES

Let us highlight that the term security policy refers to numerous aspects of information systems security such as "a set of rules that determine how a particular set of assets should be secured". Furthermore, the SP should be split into multiple components as it should address all the security requirements of the enterprise. RFC 2196 defined a list including the major components of a SP. We found that all these policy components can be specified similarly, even though they use distinguished security techniques. Abstracting away from its context, a security policy representation should contain the protected entities, the operation modeling their interaction, and the security rules that must be followed. Developing a framework that allows to model generic security policies should necessarily consider these three components. Entities refer to the elements constituting the system of interest. For instance, an entity can be, depending on the application, a human user, a computer, or a process. Operations are used to represent the actions that can be performed by the different entities; they range from physical access (for a human being) to network connections (for networked computers). Operations slightly differ from actions in the sense that they are more abstract with regard to the system entities. For example, if (., ., ., .) is an operation representing the connection establishment between two machines, then (1, 2, 1, 2) is an action that represents the connection establishment between two specific hosts (i.e., , 2) using source and destination ports (i.e., 1, 2). Security rules define the constraints that allow differentiating between legitimate and prohibited actions.

Many-sorted signatures [73] can be used to manipulate the previous sets as they are commonly used to handle 'typed' data values [74]. It is noteworthy that a rule can be either positive or negative, meaning it can allow the action or deny it.

To illustrate what has been stated, we consider a network consisting of some PCs connected to the public network. The access to objects on every PC is secured using the Bell-LaPadula (BLP) multi-level access control model. The BLP model restricts access to objects based on the sensitivity of the information contained in the objects and the formal authorization of subjects to access information of such sensitivity. It also supports discretionary access control by checking access rights from an access matrix. In this model, an access request (subj, obj, acc) is granted if and only if all of the following properties are satisfied:

simple security property (no read up): if acc is read, then level(subj) should dominate level(obj).

{*}-property (no write down): if acc = append, then level(obj) should dominate level(subj); if acc = write, then level(obj) should be equal to level(subj).

Disretionary security property: the (subj, obj) cell in the matrix contains acc.

All the operations defined in this signature correspond to constants except , which is an application assigning a clearance to a subject or an object. The predicate represents an access request for a to an according to an

mode. Axioms Φ1−3 formally model the properties (1) and (2) discussed above.

15. CONCLUSION

In this paper, responses to security incidents are selected using IRPCMs and based on their impact on the unauthorized results caused by the incidents. Enhancing this approach by developing a complete sys-tem for selecting security responses would be of utmost interest. The aim is to develop a system for finding the correct decisions in a finite time. The approach followed by this system must guarantee an adaptive response to incidents through the use of criteria to be defined and analyzed.

REFERENCES

[1]. S. Snedaker, "IT Security Project Management Handbook". Syngress, 2006.

[2]. J. Davis, "Information Security Management Handbook", ch. Measuring ROI on Security, pp. 1056–1060. CRC Press LLC, 5th ed., 2004.

[3]. R. Richardson, "2007 csi computer crime and security survey," tech. rep., Computer Security Institute, 2007.

[4]. N. Damianou, N. Dulay, E. Lupu, and M. Sloman, "The ponder policy specification language," in Proceedings of Policy 2001: Workshop on Policies for Distributed Systems and Networks, pp. 18–39, 2001.

[5]. N. Dulay, E. Lupu, M. Sloman, and N. Damianou, "A policy deployment model for the ponder language," in Proceedings of IEEE/IFIP International Symposium on Integrated Network Management, (Seattle, USA), 2001.

[6]. J. Dai and J. Alves-Foss, "Logic based authorization policy engineering," in Proc. 6th World Multi conference on Systemics, Cybernetics, and Informatics, pp. 230– 238, July 2002.

[7]. S. Jajodia, P. Samarati, and V. S. Subrahmanian, "A logical language for express-ing authorizations," in Proceedings of IEEE Symposium on Security and Privacy, (Oakland, CA, USA), 1997.

[8]. M. Hamdi, "Mathematical Aspects of Network Security Risk Analysis". PhD thesis, SUP'COM, July 2005.

[9]. N. Satoh and N. Komoda, "A labor time estimation model for the information security audit by quantitative analysis i and regression analysis," in Proceedings of the 4th WSEAS International Conference on E-ACTIVITIES, (Miami, Florida, USA), pp. 136–141, November 17-19 2005.

[10]. T. Akin, "Information Security Management Handbook", ch. Managing the Re-sponse to a Computer Security Incident, pp. 2977–2986. CRC Press LLC, 5th ed., 2004.

[11]. C. C. Center, "Csirt faq." http://www.cert.org/csirts/csirt faq.html, April 2008. Last visited: April 2008.

[12]. M. J. West-Brown, D. Stikvoort, K.-P. Kossakowski, G. Killcrece, R. Ruefle, and M. Zajicek, "Handbook for computer security incident response teams (csirts)," Handbook CMU/SEI-2003-HB-002, CMU/SEI, April 2003. 2nd Edition.

[13]. C. Hare, Information Security Management Handbook, ch. CIRT: Responding to Attack. CRC Press LLC, 2004.

[14]. K. M. Shaurette and T. J. Schleppenbach, Information Security Management Handbook, ch. Incident Response Exercises. CRC Press LLC, 2004.

[15]. R. Campbell, "A modular approach to computer security risk management," in

[16]. Proceedings of the AFIPS Conference, 1979.

[17]. R. Summers, Secure Computing. McGraw Hill, 1997.

[18]. "Risk management," in AS/NZS 4360:1999, Standards Australia and Standards New Zealand, 1999.

[19]. "Iso/iec 1799:2000 (part 1), information technology-code of practice for information security management," 2000.

[20]. "Bs 7799-2:2002 (part 2), information security management systems," 2002.

[21]. G. Stonebumer, A. Grogen, and A. Fering, Risk Management Guide for Information Technology

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 4, APRIL 2015, PP 280-295

Systems. National Institute fro Standards and Technology. special publication 800-30.

[22]. Government of Canada, Communications Security Establishment, A Guide to Risk Management and Safeguard Selection for IT Systems, January 1996.