



CNAS REPORT

Printed January 2008

CNAS-2008-006
Public

Supersedes CNAS-2007-005
Dated Dec. 2007

Algebraic Specification of Network Security Risk Management

Mohamed Hamdi
Noureddine Boudriga

Prepared by
CN&S Research Lab.

The Communication Network and Security (CN&S) research Laboratory,
(Created in 1999, 02/UR/11-08) is located at the Communication School of Engineering
(University of 7th of November at Carthage, Tunisia).

Approved for public release.

Copyright © 2007 by the Communication Networks and Security Research Lab. All rights reserved.

NOTICE: No part of this publication may be reproduced, stored in a retrieval system, or transmitted without written authorization from the CN&S research lab.

Available from

CN&S research lab.
Engineering school of communications.
Techno-parc El Ghazala, Route de Raoued.
Ariana, 2083, Tunisia.

Telephone: (+216) 71857000 (ext. 2104)
Facsimile: (+216) 71856829
E-Mail: cnas@laposte.net

Approved for public release

Professor Nouredine Boudriga
Head of CN&S research lab.

Algebraic Specification of Network Security Risk Management

Mohamed Hamdi, Nouredine Boudriga
CNAS Research Lab., Telecommunication School of Engineering (SUP'COM)
University of 7th of November, Carthage, Tunisia
mmh@certification.tn, nab@supcom.rnu.tn

ABSTRACT

Existing risk analysis techniques are often hard to handle in real world contexts without the use of appropriate software because of their computational complexity. This makes managers and security analysts use simplified methods to evaluate security investments. However, these methods have been shown to be inefficient in most cases. Therefore, an automated tool for risk management would be of great interest, provided that it allows reasoning on attacks and helps building security decisions. This paper provides an algebraic specification of network security risk management activities. It constitutes a helpful mean to reason about automating the risk assessment process without taking into consideration implementations issues.

Keywords

KEYWORDS. Risk management, formal specifications, algebraic modeling.

1. INTRODUCTION

Many enterprises have been involved in building the digital world during the last few years. This is mainly due to the fast progress in communications and computer science fields. Information systems have become a crucial concern in modern production systems. However, these systems are highly vulnerable to malicious attacks and abuses because of the intrinsic properties of the communication infrastructure and protocols (poor authentication, no secrecy, etc.). Statistics show that attacks are getting more sophisticated and more harmful. For instance, the financial loss caused by the attacks performed against the enterprises that participated to the annual CSI/FBI Computer Crime and Security Survey [1] had reached \$201,797,340 in 2002. An intriguing note about this study is that it revealed that most of the attacked sites were protected by several security mechanisms. Therefore, managers have to be aware that not all of the security products available in the market can be efficient in the proper contexts of their organizations. Thus, a rational

and coherent reasoning has to be provided in order to make the appropriate security decisions under security risks.

Recently, many attempts to develop quantitative methods to handle and mitigate computer security risks were performed. Furthermore, the interest of governmental entities in this issue appeared through the publication of several guides that are intended to help organizations in limiting the impact of potential attacks [2, 3, 4, 5]. In spite of this consistent effort, risk management is still not applied in many organizations. In our sense, this is caused by the complexity of risk mitigation approaches, especially in large enterprises where the application of the aforementioned methods without use of automated tools is quasi unfeasible.

This paper addresses the problem of automating the risk management process. Starting from the notion of risk management specifications, it is shown that algebraic specifications can be helpful in our context. This technique, which has been widely applied in software engineering [6, 7], allows the definitions of the operations that have to be done to select the convenient security decisions without a complete description of how these operations are performed. It merely dictates the required properties of the risk-based decision process. Moreover, as an extension to this reasoning, we demonstrate that concrete risk management projects can be represented by algebras belonging to a specific class. This modeling constitutes a useful mean for proving the correctness of the proposed risk management method (with respect to the specified requirements) by checking its formal description against inconsistencies.

The algebraic approach we are developing in this paper differs from the traditional algebraic specifications in different manners. Not only it allows reasoning and proving properties about scenarios of attacks, but also it provides the following main features:

- It defines a uniform environment for the description of assets, vulnerabilities, attacks, and security decisions, and makes it feasible to check for proofs using inter-related order relations.
- It defines a natural algebra of attacks where security decisions appears to work as virtual inverses, which can be constructed gradually and efficiently.

The remaining of this paper is organized as follows. The

next section presents risk management fundamentals. Section 3 introduces an algebraic specification of the risk management process. Section 4 proposes an algebra representing a risk management method that is proved to be correct with respect to the defined specification. Section 5 gives the conclusion and perspectives of the paper.

2. RISK MANAGEMENT

Risk management approaches fall into two categories: qualitative risk management [8] and quantitative risk management. The former consists in prioritizing the various risk elements in subjective terms. The latter is based on quantifying the magnitude of risk created by the exposure of the target system to negative events. Techniques belonging to the first class are more used than the others as they are more easy to implement. Nonetheless, they are by far less efficient because they are essentially based on the expertise of the analysis team. The rest of the discussion therefore concerns quantitative methods. In the following, we outline the main principles of these methods and discuss briefly several existing approaches. Finally, we present a scenario-based view of quantitative risk management showing the interest of our approach.

2.1 Basic steps

In this subsection, the key concepts of risk management are presented. The reader should be familiar with several basic terms such as asset, vulnerability, threat and risk. For an extensive presentation of these notions, a good and complete risk management taxonomy can be found in [9].

Abstracting away from the implementation approach, most of security risks management processes are five-fold:

- Assess the various assets of the organization,
- Identify the weaknesses of the system,
- Assess the potential attacks that threaten the analyzed system,
- Assess the security risks,
- Select the appropriate security decisions.

Asset assessment aims at gathering detailed information about the various assets. Indeed, it often consists in establishing an inventory containing all the resources including different attributes such as criticality or value.

The purpose of the second step (vulnerability identification) is to detect the weaknesses of the described assets. This can be performed through the use of various detection mechanisms that depend on the nature of the vulnerability. For instance, application-level breaches can be detected through the use of automated scanning tools (e.g., Nessus, Nmap, etc.), while questionnaires are the most efficient mean to identify human vulnerabilities. In addition, expert opinions combining techniques are often useful when trying to find procedural weaknesses in the security documents (e.g., strategy, policy, practices, etc.).

Threat assessment consists in identifying (on the basis of the detected vulnerabilities) harmful events that may cause damage to the target system. Quantitative measures (such as probability, frequency, and severity) are allocated to each identified threat. These values are often hard to determine as they vary according to many factors such as geographical position, political stance, or activity sector.

The objective of risk assessment is to prioritize the risks with respect to a set of quantitative parameters. In most cases, these parameters include an estimation of the impact of the risk relatively to the target asset, an estimation of the probability of the corresponding threat and an estimation of the likelihood of the related vulnerabilities.

Finally, to select the optimal countermeasures, a set of candidate risk mitigation techniques is proposed. Then, with regard to several criteria, a subset representing the actions that would be effectively taken to protect the system is chosen. The basic attributes of an alternative countermeasure might include the efficiency of the protection technique, its cost and its feasibility.

2.2 Common approaches

This discussion focuses on two methods: OCTAVE and GAO [10, 11]. Both of them are hybrid in the sense that they include qualitative and quantitative issues.

OCTAVE is a three phase approach that was developed by the Software Engineering Institute (SEI) and Carnegie Mellon University (CMU). It addresses both organizational and technical issues. According to this method, a security risk is defined by four elements: asset, threat, vulnerability and impact. The evaluation process is performed on a per-asset basis. It relies on qualitative criteria to evaluate risks. One of the key features of OCTAVE is the use of threat profiles which are tree structures based on inductive logic used to represent a range of threats against a specific asset.

GAO/AIMD method, that was developed by the US General Accounting Office (GAO), relies on five principles:

- Assess risk and determine needs,
- Establish a central management focal point,
- Implement appropriate policies and related controls,
- Promote awareness,
- Monitor and evaluate policy and control effectiveness.

Prioritizing risks is based on a risk assessment matrix that represents risk level according to the severity and the probability of occurrence of harmful events. These two factors are scaled in a qualitative way. The main difference between the two methods cited above is the monitoring step that is included in GAO method. This monitoring activity confers to this approach the possibility to perform a continuous evaluation of the security state of the system.

2.3 Scenario-based security evaluation

Scenario-based risk management relies on the attack scenario concept. An attack scenario is a chain where the last link is called the main attack, the first link is an elementary attack while the other links are intermediate attacks. The success of each attack in this chain allows the attacker to perform the attack represented by the next link in the chain. This approach expresses accurately the occurrence of real attacks where a malicious user performs a set of successive attacks in order to achieve his major goal: the main attack.

Rather than considering attacks as independent events, the risk analyst should establish a causal relation between the basic attacks. It is noteworthy that the introduction of this notion does not induce a big complexity to the risk management process as the quantification of attack scenarios can be easily performed by the use of the basic attacks weights.

An important work has been done to model efficiently attack scenarios. Most of the proposed approaches rely on hierarchical data structures (trees) where roots constitute the most important attacks. Event-trees, that are widely used in other fields such as software engineering, have been adapted and included in the OCTAVE method as threat profiles. More recently, B. Schneier [12] introduced the concept of *attack trees*, which are particularly efficient to model the attacker behavior. Since, they have been revisited by some researchers (see for example [13]) and they are continuing to give rise to an important interest of the security community.

Previous sections have argued that there exists many approaches and tools to perform computer security risk management. The development and the evaluation of these methods is often based on qualitative reasoning. Such subjective assessment can not be used to check the correctness of a decision making technique or to detect its paradoxes and inconsistencies. To overcome this lack, formal specifications and proofs can be used as they have a number of benefits. First, they allow the definition of the elementary variables and functions of a risk management model without considering complex implementation issues. In addition, a set of requirements used to state about the correctness of an implementation of the formal specification can be introduced by using logical axioms. These axioms can be used to prove various properties the specification by building an appropriate deduction system.

3. A FORMAL SPECIFICATION OF THE RISK MANAGEMENT PROBLEM

In this section, we construct gradually a formal specification scheme for risk management starting from the relation between assets and vulnerabilities. Then we show how proving tools can be established based on deductive systems.

3.1 The risk management signature

It appears from the previous section that a common risk management method handles a set of entities that can be classified independently from the adopted approach. In fact, risk evaluation is always done depending on assets, vulnerabilities and threats. Furthermore, it can be remarked that, from a global point of view, the operations that guide the

security analyst in the selection of the optimal decisions are similar for most of the approaches. Then, many-sorted signatures, that are commonly used to handle 'typed' data values, can be used in our context to manipulate the family of sets mentioned above. A many sorted signature [14] is typically constituted of a set S of sort names, an $S^* \times S$ -sorted set of operation names and an S^* -sorted set of predicate symbols. For instance, the following signature represents the two first steps of the risk management process.

sig	$\rho_1 =$	
sorts		$asset, vuln$
preds		$ispresent : asset \times vuln$

Two sorts are essentially related to risk evaluation, *asset* and *vuln* that represent respectively assets and vulnerabilities. In addition, a predicate symbol (*ispresent*) is contained in ρ_1 to indicate whether a particular asset exhibits a given vulnerability or not.

Signatures can be extended through the use of the operator **enrich** that allows the addition of new sorts and operations. For example, the signature ρ_1 can be enriched by several sorts, operations and predicates that represent the threat assessment and risk analysis steps. Thus, the obtained signature ρ_2 would model the four first steps of the security risk evaluation process.

sig	$\rho_2 =$	enrich ρ_1 by
sorts		$attack$
opns		$-\star - : attack \times attack \rightarrow attack$ $1_a : \rightarrow attack$
preds		$exploits : attack \times vuln$ $impossible : attack \times asset$ $isminimal : attack$ $-\leq_a - : attack \times attack$

A new sort (*attack*) has been added to represent computer network attacks. The operation \star is a composition law that is used to build attack scenarios. It is noteworthy that the composition of two attacks is also an attack. In other terms, the sort *attack* represents both simple and composite attacks, which are called attack scenarios.

The predicate *exploits* indicates if an attack exploits a vulnerability. On the other hand, *impossible* states whether it is possible to carry out a given attack against an asset. In addition, a unary predicate (its arity is equal to 1) denoted *isminimal* is introduced. It permits to check if a composite attack is **minimal**. An attack scenario is said to be minimal if, and only if, it is no longer a scenario when one of its constituents is removed. The last predicate is an order defined on the set of attacks and will be discussed with further details in the sequel.

To ensure the ultimate step (countermeasure selection), the signature is extended again by adding up a sort called *decision* and the following operations and predicates.

The operation \bullet is a composition law meaning that combining the application of two security decisions to two assets

sig	$\rho_3 =$	enrich ρ_2 by
	sorts	<i>decision</i>
	opns	$(-, -) \bullet (-, -) : decision \times asset \times decision \times asset \rightarrow decision \times asset$ $a^* : \rightarrow asset$ $d^* : \rightarrow decision$
	preds	<i>mitigates</i> : <i>decision</i> \times <i>attack</i> $(-, -) \succ_c (-, -) : decision \times asset \times decision \times asset$

results in the application of a composite decision to a composite asset. It can be remarked that a security countermeasure is represented by a decision d and an asset a meaning that d is applied to a .

The predicate symbol \succ_c represents the most important concept in the signature as it will serve to compare countermeasures. Consequently, the overall objective of the risk management process is to find the decision d^* and the asset a^* such that for every decision d and asset a :

$$(d^*, a^*) \succ_c (d, a) \quad (1)$$

Predicate symbol \succ_c and the construction process of d^* will be addressed in the next section. Thus, ρ_3 is a first-order signature that can be denoted $(S_{\rho_3}, \Omega_{\rho_3}, \Pi_{\rho_3})$ where S_{ρ_3} is the set of sorts, Ω_{ρ_3} is the set of operations and Π_{ρ_3} is the set of predicate symbols. It just introduces the basic ingredients of the algebraic structure without specifying any requirements. To express the properties of the signature ρ_3 , we have to introduce a deduction system defined by a logic and a set of inference rules.

3.2 Proving properties in the risk management signature

In order to give an axiomatic description to signature ρ_3 , we add up a set of equations denoted Φ_{ρ_3} . Each of these equations has the form

$$\forall X. t_1 = t'_1 \wedge \dots \wedge t_n = t'_n \Rightarrow t = t', \quad (2)$$

where X is a S_{ρ_3} -sorted set of variables and

$$\left\{ \begin{array}{l} t, t' \in |T_{\rho_3}(X)|_s \\ t_i, t'_i \in |T_{\rho_3}(X)|_{s_i}, i = 1, \dots, n \end{array} \right.$$

$|T_{\rho_3}(X)|$ is the term algebra corresponding to ρ_3 and generated by applying the operations of Ω_{ρ_3} to variables belonging to the sorts of S_{ρ_3} .

These equations constrain the permitted behavior of the operations that belong to Ω_{ρ_3} . The reader would have noticed that Φ_{ρ_3} axioms are conditional equations. In fact, equational conditional logic is more expressive in our context as we introduced a set of predicate symbols in ρ_3 .

Thus, by adding the following axioms, this signature becomes a presentation denoted ρ_4 .

Here is an informal lecture of the nine defined axioms:

- Axiom (φ_1) states that if an attack at exploits a vulnerability v which is present in an asset as ; then, the attack at is possible to carry out against as .
- Axiom (φ_2) guarantees that the operation \star is associative.
- Axioms (φ_3) and (φ_4) gives that, for two attacks at_1, at_2 , $at_1 \leq_a at_2$ if, and only if, there exist two attacks at_3, at_4 such that $at_3 \star at_1 \star at_4 = at_2$.
- Axiom (φ_5) states 1_a is neutral element for the operation \star .
- Axiom (φ_6) ensures that the only composite attacks that are equal to the neutral element are compositions of 1_a by \star .
- Axiom (φ_7) gives that the composition of 0_a with any attack leads to 0_a ,
- Axiom (φ_8) ensures The neutral element 1_a is unique.
- Axioms (φ_9) and (φ_{10}) guarantee that the predicate \succ_c is antisymmetric and transitive.
- Axiom (φ_{11}) gives that the operation \bullet is commutative.

These axioms have a double interest. They are useful to check if a practical implementation of Ω_{ρ_4} -operations is correct. Furthermore, they form a key component of a deduction system that automates proofs in ρ_4 . The remaining part of this section is related to logical calculus over this presentation.

Consider an S_{ρ_4} -sorted set denoted X . Define the respective sets of atoms and conditional axioms as follows:

$$At(\rho_4, X) = \{t = t' | t, t' \in |T_{\rho_4}(X)|_s\} \cup$$

$$\left\{ p(t_1, \dots, t_n) | p : s_1 \times \dots \times s_n \in \Pi \text{ and } t_i \in |T_{\rho_4}(X)|_{s_i}, i = 1, \dots, n \right\}, \quad (3)$$

$$Cond(\rho_4, X) = \{\forall X. \epsilon_1 \wedge \dots \wedge \epsilon_n \Rightarrow \epsilon_{n+1} | \epsilon_i \in At(\rho_4, X), i = 1, \dots, n\}. \quad (4)$$

These sets are respectively the set of basic logic formulas (directly obtained from axioms or predicates) and the set of derived first-order formulas (obtained from the atoms).

Then, we introduce an inference system denoted \vdash to derive conditional equations from Φ_{ρ_4} . \vdash satisfies the axioms and

pres	$\rho_4 =$	enrich ρ_3 by
axioms		$(\varphi_1) \forall as : asset. \forall v : vuln. \forall at : attack.$ $exploits(at, v) \wedge ispresent(v, as) \Rightarrow impossible(at, as)$ $(\varphi_2) \forall at_1, at_2, at_3 : attack. ((at_1 \star at_2) \star at_3) = at_1 \star (at_2 \star at_3)$ $(\varphi_3) \forall at_1, at_2 : attack. at_1 \leq_a at_2 \Rightarrow \exists at_3, at_4 : attack. at_3 \star at_1 \star at_4 = at_2$ $(\varphi_4) \forall at_1, at_2, at_3, at_4 : attack. (at_1 \star at_2 \star at_3 = at_4) \Rightarrow (at_2 \leq_a at_4)$ $(\varphi_5) \forall at : attack. (at \star 1_a = at) \wedge (1_a \star at = at)$ $(\varphi_6) \forall at_1, at_2 : attack. at_1 \star at_2 = 1_a \Rightarrow (at_1 = 1_a) \wedge (at_2 = 1_a)$ $(\varphi_7) \forall at : attack. (at \star 0_a = 0_a) \wedge (0_a \star at = 0_a)$ $(\varphi_8) \forall at_1, at_2 : attack. \neg((at_1 = 0_a) \wedge \neg(at_3 = 0_a) \wedge (at_1 \star at_2 \star at_3 = at_2))$ $\Rightarrow (at_1 = 1_a) \wedge (at_3 = 1_a)$ $(\varphi_9) \forall d_1, d_2 : decision. \forall as_1, as_2 : asset.$ $(d_1, as_1) \succ_c (d_2, as_2) \wedge ((d_2, as_2) \succ_c (d_1, as_1)) \Rightarrow (d_1 = d_2) \wedge (as_1 = as_2)$ $(\varphi_{10}) \forall d_1, d_2, d_3 : decision. \forall as_1, as_2, as_3 : asset.$ $((d_1, as_1) \succ_c (d_2, as_2)) \wedge ((d_2, as_2) \succ_c (d_3, as_3)) \Rightarrow (d_1, as_1) \succ_c (d_3, as_3)$ $(\varphi_{11}) \forall d_1, d_2 : decision. \forall as_1, as_2 : asset. (d_1, as_1) \bullet (d_2, as_2) = (d_2, as_2) \bullet (d_1, as_1)$

Figure 1:

the inference rules of a first-order predicate logic proof system that are listed below.

Congruence Axioms

- $\Phi \vdash \forall X. t = t$ (reflexivity)
- $\Phi \vdash \forall X. t = t' \Rightarrow t' = t$ (symmetry)
- $\Phi \vdash \forall X. t = t' \wedge t' = t'' \Rightarrow t = t''$ (transitivity)
- $\Phi \vdash \forall X. t_1 = t'_1 \wedge \dots \wedge t_n = t'_n \Rightarrow f(t_1, \dots, t_n) = f(t'_1, \dots, t'_n)$
- $\Phi \vdash \forall X. t_1 = t'_1 \wedge \dots \wedge t_n = t'_n \wedge p(t_1, \dots, t_n) \Rightarrow p(t'_1, \dots, t'_n)$

Proper Axiom

$$\Phi \vdash \forall X. \varphi \text{ for } \forall X. \varphi \in \Phi$$

Substitution

$$\frac{\Phi \vdash \forall X. \varphi}{\Phi \vdash \forall Y. \varphi[\theta]} \text{ for } \theta : X \rightarrow |T_{\rho_4}(Y)|_s$$

Cut Rule

$$\frac{\Phi \vdash \forall X. \epsilon_1 \wedge \dots \wedge \epsilon_n \Rightarrow \eta_i \quad \Phi \vdash \forall Y. \eta_1 \wedge \dots \wedge \eta_k \Rightarrow \epsilon}{\Phi \vdash \forall X \cup Y. \eta_1 \wedge \dots \wedge \eta_{i-1} \wedge \epsilon_1 \wedge \dots \wedge \epsilon_n \wedge \eta_{i+1} \wedge \dots \wedge \eta_k \Rightarrow \epsilon}$$

To illustrate the use of this deductive system, we propose to prove that the predicate \leq_{attack} is an order relation ; in other terms, that it is reflexive, antisymmetric and transitive.

1. Reflexivity:

$$\frac{(at_1 \star at_2) \star at_3 = at_1 \star (at_2 \star at_3)}{(1_a \star at) \star 1_a = 1_a \star (at \star 1_a)} \quad (\varphi_2)$$

$$\frac{(1_a \star at) \star 1_a = 1_a \star at}{(1_a \star at) \star 1_a = at} \quad (\varphi_5)$$

$$\frac{(1_a \star at) \star 1_a = at}{at \leq_{attack} at} \quad (\varphi_4)$$

2. Antisymmetry:

$$\frac{(at_1 \leq_a at_2) = (at_2 \leq_a at_1)}{\exists at_3, at_4 : attack. (at_3 \star at_1 \star at_4 = at_2)} \quad \text{Hypothesis } (\varphi_3)$$

$$\frac{\exists at_5, at_6 : attack. (at_5 \star at_2 \star at_6 = at_1)}{at_5 \star at_3 \star at_1 \star at_4 \star at_6 = at_1} \quad \text{Substitution } (\varphi_2)$$

$$\frac{at_5 \star at_3 = 1_a}{at_4 \star at_6 = 1_a} \quad (\varphi_8)$$

$$\frac{at_5=1_a}{at_3=1_a} \quad (\varphi_6)$$

$$\frac{at_4=1_a}{at_6=1_a} \quad (\varphi_6)$$

$$\frac{at_6=1_a}{at_1 = at_2} \quad (\varphi_5)$$

3. Transitivity:

$$\frac{(at_1 \leq_a at_2) = (at_2 \leq_a at_3)}{\exists at_3, at_4 : attack. (at_3 \star at_1 \star at_4 = at_2)} \quad \text{Hypothesis } (\varphi_3)$$

$$\frac{\exists at_5, at_6 : attack. (at_5 \star at_2 \star at_6 = at_3)}{at_5 \star at_3 \star at_1 \star at_4 \star at_6 = at_3} \quad \text{Substitution } (\varphi_2)$$

$$\frac{at_5 \star at_3 = at_3}{at_1 \leq_a at_3} \quad (\varphi_4)$$

The presentation ρ_4 and the inference system \vdash constitute a specification denoted ρ that induces a class of algebras $Alg(\rho)$. To prove that the application of a risk management

method in a particular organization fulfills the requirements specified in ρ , we have to model the concrete decision making process by an algebra R and prove that $R \in Alg(\rho)$. This will be the objective of Section ??.

3.3 Features of the proposed presentation

In this subsection, we underline some of the interesting characteristics of the proposed specification. First, it can be remarked that a major benefit of using algebraic specifications in our context is that they allow an efficient representation of the environment. For example, simple and composite attacks can be represented through the use of a single sort. Similarly, combinations of security countermeasures can be built using a simple composition law. Furthermore, the operations that have been defined are easy to be implemented as most of them consist in composition laws and order relations.

The composition law \star is not commutative. it has a neutral element and is associative. moreover, axiom (φ_6) shows that no attack can be considered as the inverse element of another attack for composition \star . One way to embed the above algebra in a minimal set where elements are inversible is to consider that a security decision is made to recover from an attack (i.e. performed, it gets back the system state targeted by the attack).

For evident reasons, a security decision d is nothing else but a right inverse of an attack a , say

$$a \star d = 1_a,$$

that exists (at least theoretically).

To achieve constructing a security decision, we consider available the following:

- a set of preliminary attacks, i.e. attacks for which simple decisions are available,
- a mechanism to write any attack in its minimal expression,
- for any attack targeting a set of resources, say $\{r_1, \dots, r_n\}$, there is one decision, say d , that can recover one among the resources.

Starting from these assumptions, one can build for each attack a recovering plan, which implements iteratively the following procedure:

1. find minimal form of attack a ,
2. define all resources targeted by a ,
3. choose a security decision d to recover one resource
4. state $a := a \star d$

The recovering process terminates, since the set of resources targeted by the initial attack a is finite.

Some deep study can be performed to characterize the property of the deduction system and even to enhance it by a rewriting system. In fact, considerations such as completeness, termination and confluence might be addressed in the future. Within the frame of this work, we propose to prove the consistency of the presentation by the existence of an algebraic model that validates it.

4. ALGEBRAIC MODELING OF RISK MANAGEMENT PROJECTS

In the previous section, we have pointed out that requirements on risk management activities can be expressed by the use of an algebraic specification and a first-order inference system. The objective of the following discussion is to illustrate how the conformance of a practical risk management method to these requirements can be concluded. To this end, we present an approach based on set theory and relational algebra to evaluate security risks. The advantage of this approach is that it can easily be implemented by translating it to relational databases. We will put the stress on the practical issues to emphasize the ease of applicability of the proposed method in real situations.

4.1 From the specification to the algebra

The preliminary constituents of our approach are four sets, that can be implemented through databases, representing the basic knowledge of the decision making system. These sets are:

- A set of resources, denoted R , describing the constituency of the system under analysis,
- A set of vulnerabilities, denoted V , representing the vulnerabilities that can be present in an asset belonging to R ,
- A set of attacks, denoted A , representing the potential malicious actions that can be performed on the elements of R ,
- A set of decisions, denoted D , containing the eventual decisions that a security analyst could make to enhance the security level of the system.

To build a multi-sorted algebra α on the basis of the specification described in section 3, the first step is to define the carrier set related to each sort. In our case, this task can be achieved using the sets mentioned above.

- $|\alpha|_{asset} = R^*$,
- $|\alpha|_{vuln} = V^*$,
- $|\alpha|_{attack} = \bigcup_{n \in \{1, \dots, card(A)\}} \mathcal{S}_n$,
- $|\alpha|_{decision} = D^*$.

where $(.)^*$ denotes the set of partitions of a given set and \mathcal{S}_n is the set of permutations on $\{1, \dots, n\}$.

It is worth to mention that the structures of $|\alpha|_{asset}$, $|\alpha|_{vuln}$, $|\alpha|_{attack}$ and $|\alpha|_{decision}$ allow the representation of elementary and composite entities. Nonetheless, these sets have different natures as the order of the elements of a composite entity is significant only for attacks.

The use of composite assets, vulnerabilities and decisions, that was first introduced in NetRAM, is particularly effective to represent several situations that can not be easily handled with existing asset-driven approaches. For instance, if r_1 and r_2 are two assets exhibiting a vulnerability that can be remedied using a firewall, the decision $d = \text{"Implementing a firewall"}$ will be selected two times (one for each resource). Our representation permits to avoid this redundancy as the combination $\{r_1, r_2\}$ is also considered as a resource. Thus, we can directly say that d should be applied to $\{r_1, r_2\}$. Furthermore, composite assets allow a good description of the network topology as networks and subnetworks can be seen as elements of R^* . Likewise, V^* can be used if a given attack exploits more than one vulnerability.

To build attack scenarios, the composition law \star has to be used, as the presentation ρ states. In the algebra α , \star is defined as follows:

$$(a_1, \dots, a_p) \star_{\alpha} (a'_1, \dots, a'_q) = (a_1, \dots, a_p, a'_1, \dots, a'_q),$$

for all $a_1, \dots, a_p, a'_1, \dots, a'_q \in A$. (5)

However, from a security point of view, there is no semantic control over the elements of $|\alpha|_{attack}$. For example, if $A = \{\text{port scan, gaining root access}\}$ then the attack scenario (gaining root access, port scan) belongs to $|\alpha|_{attack}$ even it is semantically wrong. In fact, an attacker would not access to a machine as root in order to perform a port scan on it.

The predicate *ispresent* can be modeled by a binary relation defined on $|\alpha|_{vuln} \times |\alpha|_{asset}$. The table of this relation can be filled by the combination of three vulnerability detection mechanisms: automated scanning tools, questionnaires and expert opinions to detect respectively application-level, user-related and document vulnerabilities. This ensures the identification of various kinds of weaknesses.

The predicate *exploits* is also represented by a binary relation on $|\alpha|_{attack} \times |\alpha|_{vuln}$. It can be built using public attack databases such as Mitre's CVE [15] (Common Vulnerabilities and Exposures) or the NIST I-cat project [16]. Similarly, *ispossible* is a binary relation on $A \times R$. According to the axiom (φ_1) :

$$\begin{aligned} |ispossible|_{\alpha} &= |exploits|_{\alpha} \circ |ispresent|_{\alpha} \\ &= \{(a, r) \in |\alpha|_{attack} \times |\alpha|_{vuln}, \exists v \in |\alpha|_{vuln} : \\ &\quad (a |exploits|_{\alpha} v) \text{ and } (v |ispresent|_{\alpha} r)\} \end{aligned} \quad (6)$$

In addition, we define two morphisms on $|\alpha|_{attack}$ representing respectively the probability and the impact of attacks:

$$\begin{aligned} \pi_r &: (|\alpha|_{attack}, \star_{\alpha}) \rightarrow ([0, 1], \times), \\ \iota_r &: (|\alpha|_{attack}, \star_{\alpha}) \rightarrow (\mathbb{R}, +). \end{aligned} \quad (7)$$

These morphisms are resource-indexed as the probability and the impact of an attack depend not only on the nature of this attack but also on the sensitivity of the target asset.

An informal reading of these functions, showing the relation between them and the real world cases, is done for all attack a and asset r by:

- $\pi_r(a)$ is the probability that the attack a against the asset r succeeds,
- $\iota_r(a)$ is the damage caused to the system by the success of the attack a on the asset r .

We also introduce two functions I_{π} and I_{ι} representing the influence of the application of a decision $d \in |\alpha|_{decision}$ to thwart an attack $a \in |\alpha|_{attack}$ as illustrated in Equation 8.

$$\begin{aligned} I_{\pi} &: |\alpha|_{decision} \times |\alpha|_{attack} \rightarrow [0, 1], \\ I_{\iota} &: |\alpha|_{decision} \times |\alpha|_{attack} \rightarrow [0, 1]. \end{aligned} \quad (8)$$

This means that the impact of carrying out an attack a on an asset r if the decision d is implemented on this resource would be equal to $I_{\iota}(d, a) \times \iota_r(a)$.

π_r , ι_r , I_{π} and I_{ι} will serve to rank the candidate countermeasures in order to select the optimal one. However, an appropriate representation of the countermeasure must be performed prior to this selection process. In , the authors introduced a model of the risk analysis problem that relies on a binary matrix representation of security countermeasures. A $card(D) \times card(R)$ -size countermeasures matrix, denoted C , is built according to the rule given in Equation 9.

$$\forall (i, j) \in \{1, \dots, card(D)\} \times \{1, \dots, card(R)\}$$

$$\begin{cases} C_{ij} = 1 & \text{if the decision } d_i \text{ is applied to the asset } r_j, \\ C_{ij} = 0 & \text{if not.} \end{cases} \quad (9)$$

The major advantage of this representation is that it allows a global view of the problem (on the contrary of asset driven approaches) that facilitates, as it will be shown further, the optimization process. Another interesting feature of using the matrix structure is that binary relations themselves can be specified as binary relation-matrices instead of ordered pairs. For instance, to the relation $|ispresent|_{\alpha}$ corresponds a $card(V) \times card(R)$ matrix. Each row is labeled by a different element of V and the same is true for columns. An entry in row v and column r (for any $v \in V$ and $r \in R$) is 1 if, and only if, the pair (v, r) is in $|ispresent|_{\alpha}$.

The composition law \odot is implemented in α as follows.

4.2 Solving the risk analysis problem

The main goal of the risk management process is to make the optimal decisions in the sense that:

- The chosen decisions must have an optimal implementation cost,

- The chosen decisions must have an optimal influence on the impact of the possible minimal attack scenarios,
- The chosen decisions must have an optimal influence on the probability of the possible minimal attack scenarios.

Thus, decision making under security risks can be modeled by a typical multi-objective optimization problem because more than one challenge must be taken into consideration. As some of these objectives are conflicting (e.g. the influence of decisions on potential attacks should be maximized while their cost has to be minimal), the complexity of the risk analyst's task gets higher. To represent appropriately this situation, it is necessary to model each of the criteria mentioned above by a mathematical function.

Let Π , \mathbf{I} , Γ , \mathbf{I}_π and \mathbf{I}_l be the matrices representing the respective graphs of π_r , ι_r , γ_r , I_π and I_l . Rows and columns are labeled by elements of $|\alpha|_{asset}$, $|\alpha|_{attack}$ and $|\alpha|_{decision}$ depending on the represented function. The entry at a given position in a matrix is equal to the value of that function at the corresponding point. For example, for every $i \in \{1, \dots, card(|\alpha|_{decision})\}$ and $j \in \{1, \dots, card(|\alpha|_{asset})\}$, if d_i and r_j correspond respectively to the indexes i and j then

$$\Pi_{ij} = \pi_{r_j}(d_i). \quad (10)$$

These matrices constitute the basis of the evaluation process. In fact, objectives are mathematically defined as follows.

$$\begin{aligned} f_1(C) &= \|\mathbf{I}_\pi \boxtimes_C (\Pi \otimes \mathbf{M})\|, \\ f_2(C) &= \|\mathbf{I}_l \boxtimes_C (\mathbf{I} \otimes \mathbf{M})\|, \\ f_3(C) &= \|C \otimes \Gamma\|, \end{aligned}$$

where

- \mathbf{M} is the relation-matrix of $|ispossible|_\alpha$,
- \otimes denotes term-by-term multiplication of matrices,
- \boxtimes_C is an operator such that for every $card(|\alpha|_{decision}) \times card(|\alpha|_{attack})$ and $card(|\alpha|_{attack}) \times card(|\alpha|_{asset})$ matrices \mathbf{M}_1 and \mathbf{M}_2 ,
$$\forall (i, j) \in \{1, \dots, card(|\alpha|_{decision})\} \times \{1, \dots, card(|\alpha|_{asset})\}$$

$$\begin{cases} (\mathbf{M}_1 \boxtimes_C \mathbf{M}_2)_{ij} = \sum_{k=1}^{card(|\alpha|_{attack})} (\mathbf{M}_1)_{ik} (\mathbf{M}_2)_{kj} & \text{if } C_{ij} = 1, \\ (\mathbf{M}_1 \boxtimes_C \mathbf{M}_2)_{ij} = \sum_{k=1}^{card(|\alpha|_{attack})} (\mathbf{M}_2)_{kj} & \text{if } C_{ij} = 0. \end{cases}$$
- $\|\cdot\|$ is a norm on the $card(|\alpha|_{decision}) \times card(|\alpha|_{asset})$ matrices space.

To this purpose we define the following functions where C is the countermeasure matrix:

$$f_1(C) = \sum_{i=1}^{i=card(D)} \sum_{j=1}^{j=card(R)} C_{ij} \times \gamma_{r_j}(d_i),$$

$$f_2(C) = \sum_{i=1}^{i=card(D)} \sum_{j=1}^{j=card(R)} C_{ij} \times I_l(d_i, r_j, a) \times \iota_{r_i}(a),$$

$$f_3(C) = \sum_{i=1}^{i=card(D)} \sum_{j=1}^{j=card(R)} C_{ij} \times I_\pi(d_i, r_j, a) \times \pi_{r_i}(a),$$

$$f_4(C) = \sum_{i=1}^{i=card(D)} \sum_{j=1}^{j=card(R)} C_{ij} \times \gamma_{r_i}(a).$$

A preference relation (strict order) can be built on the basis of these functions. In fact, we can state that:

$$C \succ C' \text{ iff } \begin{cases} \forall i \in \{1, 2, 3, 4\}, f_i(C) \leq f_i(C'), \\ \exists i \in \{1, 2, 3, 4\}, f_i(C) < f_i(C'). \end{cases} \quad (11)$$

Consequently, the optimal combination of decisions and assets $(a^*)_\alpha$ and $(d^*)_\alpha$ correspond to the matrix C^* such that $C^* \succ C$ for every C . In fact, the strict order $(\succ_c)_\alpha$ can be derived from \succ as a countermeasure matrix can be derived from an element of $|\alpha|_{decision} \times |\alpha|_{asset}$. If (d, r) and (d', r') belong to $|\alpha|_{decision} \times |\alpha|_{asset}$, and if C and C' are the corresponding countermeasure matrices, then:

$$(d, r) (\succ_c)_\alpha (d', r') \text{ iff } C \succ C' \quad (12)$$

This problem can be solved by using heuristic methods such that genetic algorithms. However, the discussion of this issue is beyond the scope of this paper.

5. CONCLUSION

In this paper, a method for constructing progressively an algebraic presentation which represents the risk management process has been presented. The basic feature of the work is that attacks are irreversible actions that can not be reversed. Therefore, security decisions can be viewed as pseudo-inverses for attacks. The risk management decision making, from an algebraic point of view, is performed by finding the pseudo-inverses of the attacks that are likely to be carried out against the protected system. In the last section, our reasoning has been illustrated by a concrete risk analysis method that validates the algebraic specification and the deduction system.

6. REFERENCES

- [1] R. Power, "2002 CSI/FBI Computer Crime and Security Survey," Computer Security Institute, 2002.
- [2] C.J. Alberts, A.J. Dorofee, "Managing Information Security Risks: the OCTAVE Approach," Addison Wesley Professional, ISBN: 0321118863, July 2002.
- [3] G. Stonebumer, A. Grogen, A. Fering, "Risk Management Guide for Information Technology Systems," National Institute for Standards and Technology, Special Publication 800-30.
- [4] "A Guide to Risk Management and Safeguard Selection for IT Systems," Government of Canada, Communications Security Establishment, January 1996.

- [5] "Information Technology Security Evaluation Manual (ITSEM)," COMMISSION OF THE EUROPEAN COMMUNITIES, DIRECTORATE GENERAL XIII: Telecommunications, Information Market and Exploitation of Research, DIRECTORATE B: Advanced Communications Technologies and Services, 1993.
- [6] I. ClaSSen, H. Ehrig, D. Wolz, "Algebraic Specification Techniques and Tools for Software Development: the ACT Approach," AMAST Series in Computing - Vol. 1, ISBN: 981-02-1227-5.
- [7] J. A. Goguen, G. Malcolm, "Software Engineering with OBJ: Algebraic Specification in Action," Kluwer Academic Publishers, Boston, April 2000, ISBN 0-7923-7757-5.
- [8] T.P. Peltier, "Information Security Risk Analysis," Auerbach Publications, ISBN: 0-8493-0880-1, 2001.
- [9] A. Holmes, "Risk Management," Capstone Publishing, ISBN: 1-84112-341-2, 2002.
- [10] "Information Security Risk Assessment: Practices of Leading Organizations," United States General Accounting Office, GAO/AIMD-00-33, November 1999.
- [11] "Information Security Management: Learning from Leading Organizations," United States General Accounting Office, GAO/AIMD-98-68, May 1998.
- [12] B. Schneier, "*Secrets and Lies: Digital Security in a Networked World*," John Wiley & Sons, ISBN: 0471253111, 2001.
- [13] T. Tidwell, R. Larson, K. Fitch, J. Hale, "Modeling Internet Attacks," Proceedings of the 2001 IEEE Workshop on Information Assurance and Security, West Point, NY, June, 2001.
- [14] J. Loeckx, H-D. Ehrich, M. Wolf, "Specification of Abstract Data Types," Wiley&Teubner, 1996, ISBN: 0-471-95067-X.
- [15] FedCIRC, U.S. General Services Administration, "Common Vulnerabilities and Exposures," <http://www.cve.mitre.org>.
- [16] National Institute for Standards and Technology, Computer Security Division "The ICAT Project," <http://icat.nist.gov>.