Optimization of cost-based threat response for Security Information and Event Management (SIEM) systems

Gustavo Daniel Gonzalez Granadillo

THÈSE DE DOCTORAT CONJOINT TELECOM SUDPARIS et L'UNIVERSITE PIERRE ET MARIE CURIE

Spécialité : Informatique et Réseaux

École doctorale : Informatique, Télécommunications et Électronique de Paris

Présentée par

Gustavo Daniel GONZALEZ GRANADILLO

Pour obtenir le grade de
DOCTEUR DE TELECOM SUDPARIS

Optimisation de la Réponse aux menaces basée sur les coûts dans des systèmes pour la Sécurité de l'Information et la Gestion des Evénements (SIEMs)

Soutenue le 12 Décembre 2013 devant le jury composé de :

Paulo VERISSIMO
Professeur, Université de Lisbonne FFCUL / *Rapporteur*

Vania CONAN
HDR, Thales / *Rapporteur*

Sébastien TIXEUIL
Professeur, UPMC / *Examinateur*

Ludovic MÉ
Professeur, Supélec, Campus Rennes / *Examinateur*

Hervé DEBAR
Professeur, Télécom SudParis / *Directeur de thèse*

Grégoire JACOB
Ingénieur-Docteur, Télécom SudParis / *Co-directeur de thèse*

Thèse No : 2013TELE0033

PHD THESIS TELECOM SUDPARIS IN PARTNERSHIP WITH PIERRE ET MARIE CURIE UNIVERSITY

Speciality : Informatics and Networks

Doctoral School : Informatique, Télécommunications et Électronique de Paris

Presented by

Gustavo Daniel GONZALEZ GRANADILLO

To obtain the degree of
DOCTOR OF TELECOM SUDPARIS

Optimization of Cost-based Threat Response for Security Information and Event Management (SIEM) Systems

---

Presented on December 12^{th}, 2013 with the Jury composed by :

Paulo VERISSIMO
Professeur, Université de Lisbonne FFCUL / *Reporter*

Vania CONAN
Chercheur, Thales / *Reporter*

Ludovic MÉ
Professeur, Supélec, Campus Rennes / *Examiner*

Sébastien TIXEUIL
Professeur, UPMC / *Examiner*

Hervé DEBAR
Professeur, Télécom SudParis / *Thesis Director*

Grégoire JACOB
Professeur, Télécom SudParis / *Thesis Co-director*

Thèse No : 2013TELE0033

To all those who have been part of this experience, especially to José Ernesto, who has taught me to see life from a different perspective, and whose support and company throughout these 3 years have really made a difference in my life.

Abstract

Current Security Information and Event Management systems (SIEMs) constitute the central platform of modern security operating centers. They gather events from various sensors (intrusion detection systems, anti-virus, firewalls, etc.), correlate these events, and deliver synthetic views for threat handling and security reporting.

Research in SIEM technologies has traditionally focused on providing a comprehensive interpretation of threats, in particular to evaluate their importance and prioritize responses accordingly. However, in many cases, threat responses still require humans to carry out the analysis and decision tasks e.g., understanding the threats, defining the appropriate countermeasures and deploying them. This is a slow and costly process, requiring a high level of expertise, and remaining error-prone nonetheless. Thus, recent research in SIEM technology has focused on the ability to automate the process of selecting and deploying countermeasures.

Several authors have proposed automatic response mechanisms, such as the adaptation of security policies, to overcome the limitations of static or manual response. Although these approaches improve the reaction process (making it faster and/or more efficient), they remain limited since these solutions do not analyze the impact of the countermeasures selected to mitigate the attacks. In this thesis, we propose a novel and systematic process to select the optimal countermeasure from a pool of candidates, by ranking them based on a trade-off between their efficiency in stopping the attack and their ability to preserve, at the same time, the best service to normal users. In addition, we propose a model to represent graphically attacks and countermeasures, so as to determine the volume of each element in a scenario of multiple attacks.

The coordinates of each element are derived from a URI. This latter is mainly composed of three axes : user, channel, and resource. We use the CARVER methodology to give an appropriate weight to each element composing the axes in our coordinate system. This approach allows us to connect the volumes with the risks (i.e. big volumes are equivalent to high risk, whereas small volumes are equivalent to low risk). Two concepts are considered while comparing two or more risk volumes : Residual risk, which results when the risk volume is higher than the countermeasure volume ; and Collateral damage, which results when the countermeasure volume is higher than the risk volume.

As a result, we are able to evaluate countermeasures for single and multiple attack scenarios, making it possible to select the countermeasure or group of countermeasures that provides the highest benefit to the organization.

Résumé

Les SIEMs (systèmes pour la Sécurité de l'Information et la Gestion des Evénements) sont le coeur des centres opérationnels de sécurité actuels. Les SIEMs corrèlent les événements en provenance de différents capteurs (anti-virus, pare-feux, systèmes de détection d'intrusion, etc), et offrent des vues synthétiques pour la gestion des menaces ainsi que des rapports de sécurité.

La recherche dans les technologies SIEM a toujours mis l'accent sur la fourniture d'une interprétation complète des menaces, en particulier pour évaluer leur importance et hiérarchiser les réponses. Toutefois, dans de nombreux cas, la réponse des menaces a encore besoin de l'homme pour mener l'analyse et aboutir à la prise de décisions, p.ex. compréhension des menaces, définition des contremesures appropriées ainsi que leur déploiement. Il s'agit d'un processus lent et coûteux, nécessitant un haut niveau d'expertise, qui reste néanmoins sujet à erreurs. Ainsi, des recherches récentes sur les SIEMs ont mis l'accent sur l'importance et la capacité d'automatiser le processus de sélection et le déploiement des contremesures.

Certains auteurs ont proposé des mécanismes automatiques de réponse, comme l'adaptation des politiques de sécurité pour dépasser les limites de réponses statiques ou manuelles. Bien que ces approches améliorent le processus de réaction (en le rendant plus rapide et/ou plus efficace), ils restent limités car ces solutions n'analysent pas l'impact des contremesures choisies pour atténuer les attaques.

Dans cette thèse, nous proposons une nouvelle approche systématique qui sélectionne la contremesure optimale au travers d'un ensemble de candidats, classés sur la base d'une comparaison entre leur efficacité à arrêter l'attaque et leur capacité à préserver, simultanément, le meilleur service aux utilisateurs légitimes. Nous proposons également un modèle pour représenter graphiquement les attaques et les contre-mesures, afin de déterminer le volume de chaque élément dans un scénario de multiples attaques.

Les coordonnées de chaque élément sont dérivés d'un URI. Ce dernier est composé principalement de trois axes : l'utilisateur, le canal et le ressource. Nous utilisons la méthodologie CARVER pour donner un poids approprié à chaque élément composant les axes de notre système de coordonnées. Cette approche nous permet de connecter les volumes avec les risques (p.ex. des grands volumes sont équivalents à des risques élevés, tandis que des petits volumes sont équivalents à des risques faibles). Deux concepts sont considérés en comparant deux ou plusieurs volumes de risques : le risque résiduel, qui résulte lorsque le volume du risque est plus élevé que le volume de la contre-mesure, et le dommage collatéral, qui en résulte lorsque le volume de la contre-mesure est supérieur au volume du risque.

En conséquence, nous sommes en mesure d'évaluer les contre-mesures pour des scénarios d'attaques individuelles et multiples, ce qui permet de sélectionner la contre-mesure ou groupe de

contre-mesures qui fournit le plus grand bénéfice à l'organisation.

Acknowledgement

There are quite a few people that have contributed in one way or another to the accomplishment of this work. Some of these people even come unexpectedly to our lives to give us a word of courage or just to listen to us when we are down, or when we do not find an answer to our multiple questions. I would like to thank all of you from very deep inside.

Thanks to **Hervé Debar**, my thesis director, for his support, his trust and his advices throughout the three years of my thesis. It is absolutely difficult to succeed in the process of finding and developing an idea without the help of an specialist in the domain. I found in my director not only the source of wonderful ideas to develop, but also the support that a PhD student needs to develop and publish them all. Without any doubt, the influence of Hervé in my life has largely contribute to what I have accomplished today.

I am very much thankful to **Grégoire Jacob**, the co-director of my thesis, who came at the beginning of the second year to help me improve the ideas and to dedicate part of his time to discuss and find an interesting solution to most of the problems I faced. Thanks Grégoire for your time, your dedication, your remarks -always appropriate and direct to the point-, thanks for your advice and for the uncounted hours we spent through Skype discussing the ideas for my thesis.

A big thank to **Sophie Gastellier-Prévost**, the coordinator of my Master internship. I have always believed that a first experience really matters in the development of a professional career. My experience in the Security domain started with Sophie. Her dedication, her advice, her way of work, and the attention she put to my work, made me love the research environment and woke up my desire and willingness to continue with a PhD. I am sure that if I had not met you, Sophie, I would not be writing this thesis today.

Thanks to **Marilyne Laurent**, the director of my Master program. Her lectures on Security and all the practical work performed in the MSc CCN at Telecom SudParis made myself very much interested in this domain. I am very thankful for your advice and the opportunity to perform the internship at Telecom SudParis.

A special thank to the European Project **"MASSIF"** (MAnagement of Security information and events in Service InFrastructures) and all the use case providers, that contributed with this dissertation in providing a variety of real case studies. Thanks to **Malek Belhaouane** for his implementation of PyOrBAC and its integration in all the MASSIF use cases, and above all, for being always willing to help in searching the best solution to the different technical problems faced during the integration.

Thanks to **Prof. Paulo Veríssimo** and **Mr. Vania Conan** who, as reporters and members of the jury, had the hard task of reading my thesis and giving their advice in order to improve its

content. I would also like to thank **Prof. Ludovic Mé**, and **Prof. Sébastien Tixeuil** for being part of the jury of my thesis.

I would also like to thank **Nizar Kheir** for being always available to answer any question or clarify any doubt I had regarding his work, specially at the beginning of this project. Your ideas, Nizar, have been the starting point of this dissertation.

Thanks to my colleagues from the Debar's team : **Yosra, Nabil, Malek, Ghada, Gregory, Samer, Francois**, for all the time spent together, the team meetings, and discussions, but above all, thanks for the croissants ;). It has been a great experience having you all around.

A big big thank to all my friends from Venezuela that still stay in Evry : **Ernesto, Jorge, Yelimer, Samer, Francis, Glenda, Ender, Vanessa** ; and all those who have moved away : **Oscar, Alberto, Hamlet, David, Veronica, Rafael, Luis, Carina, Franklin**, as well as my foreign classmates from the CCN Master program : **Himani, Andrei, and Darius**. You guys have all been my family in this country, thanks for all the soirées and for being part of this adventure.

Thanks a lot to my family, especially to my mother **Elisabhet Granadillo**, my aunt **Dilcia Granadillo**, my brother **Daniel Gonzalez** my Grandmother **Tarcisa Granadillo**, and my Godmother **Elizabeth Varela**, for all their love, support, and words of courage, and for always showing the pride in their faces while referring to me and my achievements.

I would also like to thank all the people that I have met and whose ideas, words and philosophy have motivated me to take one choice over another. I will never stop from thanking you **Pradeep Ghosh** for all you taught me, for having such a great influence in my life and contributing to what I am today. I know that in heaven you will be very happy for this achievement. Thanks **Eric Hale**, for your words of courage and for being a role model to follow.

I can not conclude this acknowledgement without thanking **Fundayacucho**, the Venezuelan organization that granted me a scholarship to come to France in 2008, **Sonia Salicetti** for believing in me and giving me the opportunity to study abroad, **Melanie Blanchard** for all your advice and your willingness to help us when we just came to France. A big thanks to **Francoise Abad** for her effort in making sure that our missions were treated properly, and for always helping us in finding a solution to our flight and hotel problems. Thanks a lot to **Valérie Le Page** for her music and French lessons, and to the Choral group at Telecom SudParis : **SingINT**, for all the precious moments we shared together.

Thanks a lot, Merci beaucoup, Muchas Gracias to everybody that contributed directly and indirectly to the realization of this dissertation.

Enjoy your reading!!!

Contents

Chapter 1

Introduction

> There is nothing more difficult to take in hand, more perilous to conduct, or more uncertain in its success, than to take the lead in the introduction of a new order of things
>
> *The Prince*
> N. Machiavelli - 1513

Attacks in Information Security are techniques used by intruders, script kiddies or crackers to exploit existing misconfiguration or vulnerabilities in systems, networks and applications [Aki10]. These attacks have multiple motivations, such as stealing an organization's intellectual property, leaking on-line bank accounts and confidential business information, creating and distributing viruses, worms and other malware on target computers, or simply breaking the organization's infrastructure.

Back in the early 80's, hacking was only considered as a simple attempt to gain government or enterprise networks access. The image of a regular hacker was typified by a young teenager willing to obtain more knowledge than financial gain. It was not until the dot com boom that the threat landscape idea changed drastically, due to the fact that worms and viruses scaled and became the order of the day [CW08].

Over the years, attacks have moved up on the architectural layer to become content-based. Most of the new infections come either through social engineering[1] or zero-day exploit[2] [CW08]. The Phishing attack [GPGL11b,GPGL11a] is a typical example of infection that uses social engineering techniques. An example of 0-day attacks is the recent exploit that affects the latest version of Oracle Java (version 7 update 10), which exposes all Windows computers running Java to malware infections [Seg13].

Attacks against Information Systems have grown in sophistication and complexity, making the detection and reaction process a challenging task for security administrators. In addition, network and system devices are designed to support heterogeneous environments, with different characteristics and functionalities that increase the difficulty of this task. The definition of security policies to protect these systems is a process that requires a great expertise and knowledge. Inappropriate

[1] Social engineering lures legitimate users into revealing their confidential information or downloading and installing viruses and malware

[2] Zero-day attacks are targeted at systems or applications with unknown vulnerabilities that are undetectable from security applications

security policies may result in disastrous consequences for the organization. Security Information and Event Management (SIEM) systems have been developed in response to help administrators to design security policies and manage events from different sources.

SIEM platforms provide real time analysis of security events generated by network devices and applications [MHH+10]. These systems acquire high volumes of information from heterogeneous sources and process them on the fly. Their deployment thus focuses, firstly, on writing ad-hoc collectors and translators to acquire information and normalize it, and secondly, on writing correlation rules to aggregate the information and reduce the amount of data. This operational focus leads SIEM implementers to prioritize syntax over semantics, and to use correlation languages poor in features. However, as the number of attacks, and thus the diversity of alerts received by SIEMs increases, the need for appropriate treatment of these alerts has become essential.

In addition, even though the new generation of SIEMs provides response ability to automate the process of selecting and deploying countermeasures, current response systems select and deploy security countermeasures without performing a comprehensive impact analysis of attacks and response scenarios.

This research on automated reaction considers two main aspects: on the one hand, the large volume of data that represents the events in a SIEM environment, and on the other hand, the definition of security policies in heterogeneous enforcement points. In order to cope with the aforementioned limitations, we propose a decision support and reaction model to be integrated into a SIEM platform for the evaluation, ranking and selection of security countermeasures to mitigate a given attack. The model evaluates the impact of a given security incident versus the implementation of security countermeasures.

1.1 Basic Definitions

Research on attack reaction is a domain where terminology is the subject of ongoing discussions. For a better understanding, it is important to use a consistent terminology along the dissertation. The introduction begins therefore with basic definitions, most of which have been taken from [Kis11]. The rest of the definitions are found in the Glossary of Terms of this dissertation.

Information security is the protection of information based on three key factors: Confidentiality, Integrity and Availability. *Confidentiality* ensures that the information is accessible only to authorized users; *integrity* ensures the accuracy and completeness of the information; and *availability* assures that the information is accessible by authorized users whenever it is required.

In Information Security, a *threat* is a circumstance or event with the potential to adversely impact organizational operations (e.g., mission, functions, reputation), as well as organizational assets, individuals, other organizations, or even the Nation through unauthorized access, destruction, disclosure, modification of information, and/or denial of service. A *vulnerability* is a weakness in an information system, system security procedures, internal controls, or implementation that could be exploited by a threat source. A *risk* is the level of impact on organizational operations, assets, or individuals resulting from tampering with the operation of the information system. The notion of risk covers two aspects: the potential impact of a threat and the likelihood of such a threat occurring.

An *attack* is the instance or the realization of a potential threat. An *attack* is defined as the attempt to gain unauthorized access to system services, resources, or information, or an attempt to compromise the integrity of a system. The *Attack Surface* refers to the subset of resources used to attack a system (i.e. methods such as get, read, write, print, etc.; channels such as TCP, SSL, Unix sockets, etc.; and untrusted data items such as files, cookies, database records, registry

entries, etc.). The *Attack Surface Area* refers to the system's total surface exposed to a given attack. This surface includes tangible assets (e.g. PCs, mobile phones, network components, etc.), as well as intangible assets (e.g. confidential information, business reputation, etc.).

To mitigate the effects of a given attack, we need to implement security measures. *Countermeasures* are security actions required to oppose an attack, either by eliminating or preventing it, by minimizing the harm it can cause, or by discovering and reporting it so that corrective action can be taken [Kis11]. A *combined countermeasure* results from the simultaneous implementation of two or more countermeasures to mitigate a given attack. A combined countermeasure is therefore analysed as a single solution with a combined cost and a combined effectiveness.

In this research we assume that the implementation of a countermeasure always results into a security policy. *A Security Policy* defines and constrains the activities of components processing data in order to maintain security properties for systems and data. Security policies are enforced by *Policy Enforcement Points* (PEP) [YPG00, WSS+00]. A PEP is a logical entity or place on a server that enforces policies for admission control and decisions in response to a request from a user wanting to access a resource or a service on a computer or network server.

1.2 Problem Statement, Objectives and Contributions

This thesis proposes a simple and well-structured approach to select optimal countermeasures by maximizing its cost-effectiveness ratio. In addition, we compare previous mathematical models and study their limitations, which lead to the creation of a new model that evaluates, ranks and selects optimal countermeasures. This model represents the decision support framework that allows a Security Information and Event Management (SIEM) system to evaluate and select the optimal reaction strategy for a particular attack scenario.

Problem statement:

Current Security Information and Event Management (SIEM) systems select and deploy security measures without performing a comprehensive impact analysis of attacks and response scenarios.

Thesis statement:

The selection of countermeasures to mitigate the effects of a given attack requires the use of cost-sensitive metrics. Defining a quantitative model that maximizes the cost-effectiveness ratio of countermeasures contributes in the selection of the appropriate alternative. In addition, the model allows the evaluation of combined countermeasures in a scenario of multiple attacks.

In order to solve the problem and validate the thesis statement, we have set-up the following objectives:

- **Objective 1.1**: defining a quantitative model based on the costs and benefits of applying particular countermeasures for a given attack.
- **Objective 1.2**: selecting optimal countermeasures based on their individual and combined evaluation.
- **Objective 1.3**: suggesting a complete methodology to evaluate, rank and select optimal countermeasures (individuals and/or combined) to mitigate individual and multiple attacks.
- **Objective 1.4**: providing several real world case studies to show the applicability of the

model over different attack scenarios.

- **Objective 1.5**: proposing an implementation of the cost-sensitive model using a standard and widely deployed formalism.

Contributions: The proposed model relies on the optimization of cost sensitive metrics based on the Return On Response Investment (RORI) index. The optimized metric compares the expected impact of the attacks when no response is enacted against the expected impact after applying countermeasures. Moreover, since a single countermeasure is not generally sufficient to mitigate the impact of current attacks, we propose a methodology that assesses each countermeasure individually and evaluates their union and intersection with the objective of determining the overlapping area among multiple solutions. We verify all possible combinations of countermeasures and determine the solution (individual or combined) that provides the highest return to the organization. Finally, for complex attacks, we study the attack surface in order to define an approach that evaluates and selects countermeasures in a scenario of multiple attacks, while managing conflicts that may originate from the implementation of the selected security policies.

The contributions on this dissertation are summarized as follows:

- A RORI-based model for evaluating and ranking individual countermeasures in a single attack scenario *(Objective 1.1)*,
- A combination approach to evaluate and select combined countermeasures to mitigate the effects of individual attacks *(Objective 1.2)*,
- A process that selects optimal countermeasures. The process evaluates all possible combinations of security measures to select the one that provides the highest benefit *(Objective 1.3)*,
- The use of the attack surface notion to evaluate multiple attacks and multiple countermeasures *(Objective 1.3)*,
- The deployment of our model over real case studies provided by several telecommunication companies in Europe *(Objective 1.4)*,
- The evaluation of our cost sensitive model in a multiple attack scenario *(Objective 1.4)*,
- The implementation of our model for reaction, using the OrBAC formalism to deploy new security policies *(Objective 1.5)*.

1.3 Organization

This dissertation is organized as follows. Basic definitions regarding attack reaction were presented in Chapter 1. Chapter 2 describes the state of the art in intrusion and threat response as well as cost sensitive models. It proposes a countermeasure taxonomy to fulfil the lack of structured information available in this field. In addition, this section presents a comparison of different cost-sensitive models and provides the advantages and disadvantages for each of them. The limitations of the aforementioned models motivated our choice to improve the Return On Response Investment (RORI) index. From this model, the dissertation is divided into 2 parts described below.

Part I - Countermeasure Selection for Individual Attack Scenarios introduces the evaluation, ranking and selection of optimal countermeasures in a scenario of individual attacks. We describe the process to select individual and combined countermeasures and propose axioms to calculate approximations of the costs and benefits of combined solutions. We demonstrate the applicability of our proposed model by deploying two case studies: a Mobile Money Transfer Service and a Critical Infrastructure Process Control. Part I comprise three chapters:

Chapter 3 - Individual Countermeasure Selection explores the advantages and disadvantages of current cost-sensitive models to propose an improved model that overcomes the limitations of existing solutions. This model fulfils *Objective 1.1*. A sensitivity analysis is performed on the proposed model to evaluate the influence of each variable on the selection of security measures. As a result, it is possible to identify which parameters have the highest impact on the selection of countermeasures. In addition, a countermeasure selection process is proposed to evaluate, rank and select individual countermeasures to react over single attacks [GDJ+12, Con11c].

Chapter 4 - Combined Countermeasure Selection addresses the limitations of current solutions that only implement single countermeasures as a strategy to mitigate complex attacks. Based on these limitations and to fulfil *Objective 1.2*, an approach , adapted from Chapter 3, that performs all possible combinations of countermeasures is proposed as a way to evaluate the cost-effectiveness ratio of these combinations. A countermeasure selection process for the combined approach is suggested as a methodology to compute each parameter of the model and to calculate the index for each combination [GDJC12, Con11c].

Chapter 5 - Application of the Countermeasure Selection Model in Single Attack Scenarios deploys the aforementioned approaches in two real case studies defined in partnership with two European Telecommunication Companies. The first case study discusses an account takeover attack scenario in a Mobile Money Transfer Service (MMTS). The second use case discusses a control station hacking attack over a critical infrastructure process control (specifically a scenario of a dam). Both case studies perform the complete process for the evaluation, ranking and selection of individual and combined countermeasures as introduced in Chapters 3 and 4. This chapter contributes to solving *Objective 1.4* and shows the applicability of our improved cost-sensitive model. Results demonstrate that in most cases, combined countermeasures provide more interesting results than single security measures in the mitigation of individual attacks [GDJ+12, GDJC12, Con11c].

Part II - Countermeasure Selection for Multiple Attack Scenarios proposes an approach based on the attack surface to evaluate and select optimal countermeasures in a multiple attack scenario. Unlike Part I of the dissertation, we do not use approximations to calculate the cost and mitigation levels of multiple countermeasures. Instead, we compute the union and intersection of the different volumes (i.e. system, attacks, and countermeasures) by using geometrical operations. We deploy a use case on malware with two attack scenarios in order to show the applicability of our model. Part II comprises three chapters:

Chapter 6 - Attack Volume formalizes the measurement as a volume of the impact that a given attack has over a system. In relation with the notion of attack surface, we provide a geometric approach to identify several attack dimensions in a scenario of multiple attacks; making it possible to calculate the intersection and union of two or more attacks that occur simultaneously. As a result, we are able to identify the volume of each attack and the coverage of the selected security solutions. These results are useful in the evaluation of our quantitative model for multiple-attack scenarios. Chapter 6 contributes to *Objective 1.3*

Chapter 7 - Multiple Attacks Evaluation provides a comprehensive approach to evaluate and select appropriate countermeasures in a scenario of two or more attacks. The approach discusses the procedure to evaluate the impact of multiple attacks over a given target and the countermeasures to implement. The approach details the process based on three categories: independent attack surface, totally-covered attack surface and partially-covered attack surface. Chapter 7 fulfils

Objective 1.3. In addition, we present a case study that describes two famous malware infection scenarios (Conficker, and Zeus). This example deploys the process for evaluating and analysing countermeasures as described in previous chapters, and contributes to *Objective 1.4*.

Chapter 8 - Conclusions and Perspectives concludes the dissertation with a summary of contributions and presents the perspectives for future work.

From the Appendixes, Appendix A provides a French summary of the dissertation. Appendix B presents examples of inference rules used in one of the use case scenarios. Appendix C details the URI General Structure. Appendix D fulfils *Objective 1.5* by presenting a Python implementation of the Mobile Money Transfer Service (MMTS) use case (described in Chapter 5) using the Organization-Based Access Control (OrBAC) formalism to model concrete entities (e.g., subjects, actions, objects), as well as abstract entities (e.g., roles, activities, views) in order to allow the MMTS organization to define its own security policies.

Chapter 2

Intrusion and Threat Response Models and Techniques: a state of the art analysis

> Whenever a theory appears to you as the only possible one, take this as a sign that you have neither understood the theory nor the problem which it was intended to solve
>
> *Objective Knowledge: An Evolutionary Approach*
> Karl Popper - 1972

Contents

Information Technology Risks can be optimally mitigated by the appropriate selection of countermeasures. Cost-sensitive metrics help in assessing and selecting optimal countermeasures for a given attack. Several approaches (e.g. Net Present Value (NPV) [Pua09], Return on Investment (ROI) and all its variants [Sch11, Sch04]) have been used over the last two decades as financial metrics for quantifying the costs and benefits of security investments.

Security metrics (e.g. ROA [CM05], ROSI [SAS06], RORI [Khe10], etc.) have been proposed to assess risks and select countermeasures accordingly. However, current approaches are limited and evaluate countermeasures individually (no attempt has been made to evaluate multiple countermeasures simultaneously), a great level of subjectivity is necessary while estimating parameters composing the metric.

This chapter presents a state of the art in intrusion and threat response models and techniques. Section 2.1 proposes a countermeasure taxonomy based on the attacks, services, layers and techniques used to perform the attack. Section 2.2 describes financial and information security approaches in cost sensitive models and compares their benefits and constrains. Section 2.3 discusses the different methodologies proposed to date on the evaluation and selection of security measures. Section 2.4 concludes by analysing the current state of the art in intrusion and threat response models and techniques.

2.1 Countermeasure Taxonomy

The appropriate mitigation of a given attack depends on the optimal selection of security measures. In order to select a countermeasure, it is important to identify its attributes and properties as well as the consequences of its application. A great effort has been done to define and classify threats and attacks. While some authors have just listed categories of attacks [Kum95, LJ97], others have formally developed taxonomies [Lou01, HH04, MR04]. However, countermeasure taxonomies are less well developed than attack taxonomies [TAAA12].

Current intrusion response taxonomies do not include response strategies, timing response, and other factors related to the actual relevance and efficiency of chosen countermeasures [Tho07].

In addition, none of the existing response taxonomies to date distinguish between temporary responses (countermeasures that only deal temporarily with threat e.g. server port filtering), and permanent response (countermeasures that stays active for long periods of time e.g. application security patches); nor they classify security measures according to their deployment time-frame (i.e. immediate action, deferred action).

This section presents a countermeasure taxonomy based on four dimensions: the countermeasure's strategy (i.e. prevention, reaction); the type of service they protect (i.e. confidentiality, integrity, availability), the response time (i.e. duration, deployability), and the countermeasure's efficiency (i.e. high, medium, low).

2.1.1 Background on Countermeasure Taxonomy

The following authors have proposed countermeasure's taxonomies as a strategy to analyze and evaluate security mechanisms to mitigate intrusions and attacks.

Irvine and Lewin [IL99] provide a taxonomy based on security goals (e.g. confidentiality, integrity, availability) that is used as a framework to define the costs associated to the network security services. However, there are some inconsistencies in the proposed taxonomy. For instance, data confidentiality as well as audit and intrusion detection are considered in the same group of criteria. The former is a security service, whereas the latter are security technologies.

Venter and Eloff [VE03] propose a taxonomy for information security technologies that is used to secure information at application, host and network levels. The taxonomy is divided into two main sections: proactive, which groups measures that are used as a preventive strategy (before the security breach occurrence); and reactive, which groups measures used as a response strategy (as soon as the security breach is detected). However, some concepts are ambiguous (e.g. access control and passwords are considered as reactive measures).

Wang and Wang [WW03] present a countermeasure taxonomy based on the attack target (e.g. application layer, platform layer) and it is categorized along 4 dimensions: standards and policies, library and tools, administration and system management, and physical tools. The authors present an evaluation of each security technology and its effectiveness in dealing with the applicable threats and risks. However, the taxonomy lacks of important concepts such as encryption, and some general concepts like biometrics are mixed with products such as Tripwire and SQLnet.

Schumacher [Sch03], Kim et al. [KLK05], and Talib et al. [TAAA12] have proposed the definition of a countermeasure taxonomy through the use of security ontologies, in order to maintain a knowledge base of security patterns. However, most of the existing works lack of some concepts or do not clearly express the link between threats, assets and countermeasures. In addition, the definition of some concepts such as threat and attack remain ambiguous in some of the ontologies while in others, such concepts are not even developed.

According to Thomas [Tho07], there is not a response taxonomy that includes response strategy, timing response, and other factors related to actual relevance and efficiency of chosen countermeasures. In addition, it is suggested to distinguish between short-term response (countermeasures that only deal temporarily with a threat e.g. server port filtering), and long-term response (countermeasures that stays active for long periods of time e.g. application security patches).

2.1.2 Proposed Countermeasure Taxonomy

Considering the limitations of current security response taxonomies, we propose in this section a countermeasure taxonomy based on 4 dimensions: the countermeasure's strategy, the type of

service they protect, the time at which the countermeasure is applied and remains active in the system, and the countermeasure's efficiency, as depicted in Figure 2.1.

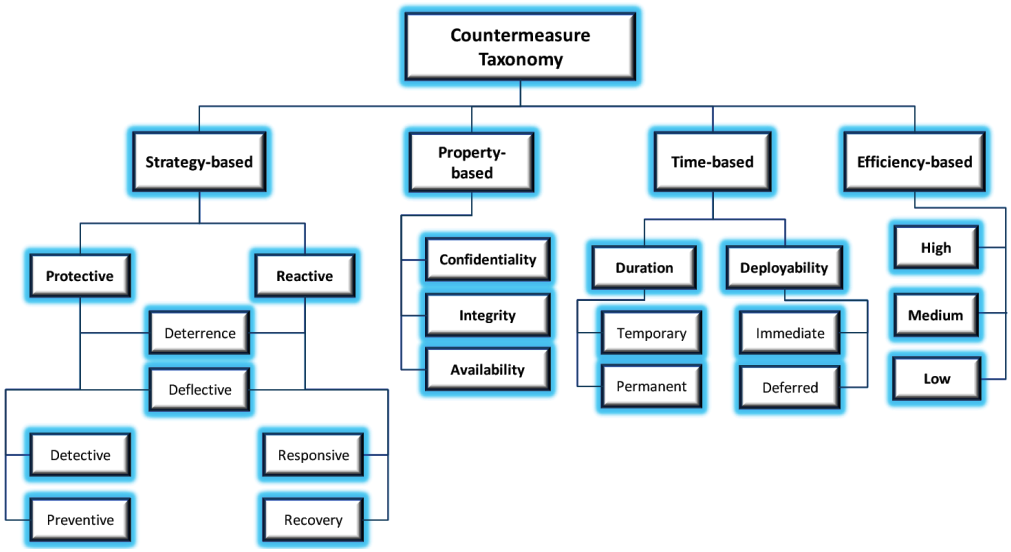

FIGURE 2.1 - Countermeasure Taxonomy

2.1.2.1 Strategy-based Classification

Based on the response strategy or the objective to which the security measure has been designed, countermeasures in our taxonomy are classified as proactive or reactive.

2.1.2.1.1 Proactive Countermeasures They cover all security controls that are employed to protect the infrastructure (assets, information, reputation, etc), from intrusions or attacks originated internally and/or externally, in order to provide detailed information of the incidents to the security analysts. The main objective of a proactive countermeasure is to secure the infrastructure to make it more robust. Examples of proactive countermeasures are the use of Policy Enforcement Points such as: IDS, Network Monitoring, Anti-virus, etc., and also, the activation of security policies and procedures, e.g. keeping traffic for longer periods, monitoring configuration changes within a predefined threshold, etc.

According to the countermeasure's objective, proactive countermeasures are further classified as detective, preventive, deterrence and deflective. Since deterrence and deflective countermeasures can be either proactive or reactive, they will be define at the end of this section.

Detective Countermeasures are security controls that detect and report unauthorized or undesired events. They include equipments or techniques whose mission is the surveillance of the target system to analyze data by comparing against signatures of known attacks, anomalous behavior, or specific outcomes of interest. Examples of these countermeasures are log monitoring and review, system audit, IDS, motion detection, etc.

Preventive Countermeasures are security measures designed to be implemented before an intrusion or attack occurs. Preventive countermeasures seek to thwart intruders from succeeding in their objectives, making it possible to reduce the likelihood of success of a particular intrusion.

Halme and Bauer [HB95] consider that preventive countermeasures ensure that a system is well conceived, designed, implemented, configured, and operated that the opportunity for intrusion is minimal. Examples of preventive countermeasures are access control mechanisms, encryption, applicative firewall, etc. [TK00].

2.1.2.1.2 Reactive Countermeasures As opposed to preventive countermeasures, reactive countermeasures are security controls that aim at stopping or delaying attackers in reaching their objectives. They are further classified as responsive and recovery countermeasure according to their final goal. These controls require the activation or deactivation of security policies such as: deny access, block users/ports/IP addresses, disable services.

Responsive Countermeasures are security controls used to respond to an attack and correct the incident. This strategy is deployed after the attack has been detected and aim at mitigating the effects and consequences of a given attack by delaying it or reducing further damage. Responsive countermeasures include procedures to eliminate a virus from an infected system, updating firewall rules to block a given IP address, disconnect the network from the outside, etc.

Recovery Countermeasures are security measures that re-stablish the system after an incident or a disaster has occurred. Examples of this strategy include Disaster Recovery Systems, backup procedures, restoring files, and systems, etc.

Due to their nature, some countermeasures are both proactive and reactive. Two types of countermeasures fall into this definition: deterrence and deflective.

Deterrence Countermeasures are less strong than preventive countermeasures since they do not attempt to preclude an intrusion, instead, they are designed to discourage attackers by making them move to a different system with a more interesting reward. For instance, hiding the system target through camouflage may deter some intruders; displaying warnings e.g. stating that access to servers is monitored, could deter internal unauthorized users; establishing obstacles to increase the effort of unauthorized users from succeeding may discourage attackers.

Deflective Countermeasures aim at luring intruders by making them believe they have succeeding in accessing system resources whereas they have been attracted to a controlled environment for observation. Examples of this technique include quarantined faux systems, controlled faux accounts, honey-pot, etc [HB95].

2.1.2.2 Property-based Classification

Countermeasures in our taxonomy are further classified based on the compromised service property (i.e. confidentiality, integrity, and availability).

2.1.2.2.1 Confidentiality Confidentiality is the protection of information from unauthorized users in a given system [TK00]. An important aspect of this property is user authentication, therefore, appropriate identification of users is essential to ensuring the effectiveness of policies that specify who access what. Countermeasures to protect or increase confidentiality in a system include the use of privacy and unpredictable passwords, multiple factor authentication, biometrics, data encryption, etc.

2.1.2.2.2 Integrity Integrity is a service property that protects the system data from intentional or accidental unauthorized modifications. Data integrity covers data in storage, during processing, and while in transit. As with confidentiality policy, authentication of users is a key element of information integrity. Countermeasures to protect information integrity include encryption procedures, firewall, password policies, etc [ASM06].

2.1.2.2.3 Availability Availability is the property of being accessible and usable upon demand by an authorized entity [Kis11]. Two facets of availability are general discusses: Denial of service, and loss of data processing capabilities [TK00]. The former refers to actions that renders computing services unusable to authorized users, the latter originates generally as a result of natural disasters such as fire, flood, earthquakes; or human actions (e.g. terrorism, strikes, etc). Countermeasures to protect system's availability include the allocation of limited access to data for unauthorized users, backup emergency process, and connection management.

2.1.2.3 Time-based Classification

In order to properly mitigate a given attack, countermeasures should be implemented as quickly as possible. Thus time is a key factor in determining the appropriate mitigation strategy. However, few of the existing security response taxonomies consider time in their design. Stakhanova et al. [SBW07b] for instance, consider the time instance of response as proactive, which allows to foresee the incoming intrusion before the attack has occurred; and delayed, which delays the response action until the attack has been confirmed.

None of the existing works have dealt with the period of time at which a countermeasure is intended to be active, nor the time they can be deployed on the system. Countermeasures in our taxonomy are classified based on their duration and deployability.

2.1.2.3.1 Countermeasure Duration: Based on the time they are active on the system, countermeasures are classified as either temporary or permanent responses.

Temporary Responses are countermeasures that are active for a limited period of time (generally some days or some hours). They protect temporarily the system from threats or attacks and get deactivated once the danger is completely mitigated. Examples of this kind of countermeasures are temporarily activation/deactivation of user accounts, quarantining hosts/accounts, blocking suspected IP addresses or Ports, etc.

Permanent Responses are countermeasures that are active for long or an unlimited period of time. Once they get activated, there is no need to deactivate them. Examples of permanent countermeasures are OS hardening, application security patches, etc.

2.1.2.3.2 Countermeasure Deployability: Based on the their capability to be implemented on the system, countermeasures in our taxonomy are classified as immediate or deferred.

Immediate Activation includes all security controls that are activated immediately after the detection of a security incident. This classification covers independent countermeasures, or those dependent solutions whose pre-requisite for their activation -installation of Policy Enforcement Points e.g. Firewall, IDS, Database, etc - has been already met. Examples of this countermeasures are; update firewall rules, block suspicious incoming/outgoing network connections, block ports/ IP addresses/ user accounts, deny/allow transactions, and all other actions that are possible to be executed at the moment they are required.

Deferred Activation requires a given period of time (e.g. hours, days or system restart) to be activated. This classification includes countermeasures that generally demand substantial modifications in the system, changes in the topology design of the network, purchase of new hardware, software, and other related actions. Examples of deferred countermeasures are: hardened OS, anti-virus installation, implementation of new biometric system, etc.

2.1.2.4 Efficiency-based Property

The efficiency of a given countermeasure is measured as the level of reduction of the total risk, or the ability to mitigate the attack. According to Norman [Nor10], the risk (R) is determined as the product of:

- The degree of vulnerability 'V' (given by the probability of a given set of vulnerabilities existing),
- The level of threat 'P' (given by the probability of a given set of attacks being able to hit the former vulnerabilities), and
- The impact or consequence 'C' (given by intrusions caused by the successful combination of the two clauses above)

As a result, the risk is calculated as $R = V \times P \times C$, and the efficiency level (E) is computed as the risk reduction percentage that results from the application of a given countermeasure. For instance, if the risk of a given attack before countermeasure is $R_1 = 10 \times 7 \times 7 = 490$ and the resulting risk after the application of a particular countermeasure is $R_2 = 7 \times 6 \times 6 = 252$, then the efficiency level is $E = 100 - (\frac{R_2 \times 100}{R_1}) = 51.43\%$ As a result, countermeasures in our taxonomy are classified as high, medium and low level efficiency.

2.1.2.4.1 High-level Efficiency: These countermeasures are more aggressive than the rest of the solutions since their mission is to stop the service degradation as soon as possible at the expense of possible side-effects. Examples of high level efficiency countermeasures are: blocking user accounts/IP addresses/Port numbers, stopping compromised processes, implementing IDS/IPS, updating firewall rules, etc.

2.1.2.4.2 Medium-level Efficiency: They are generally less aggressive and therefore less expensive than high-level impact countermeasures since their objective is to prevent the propagation of a given attack. Medim-level efficiency countermeasures considers not only the benefit, but also the cost of the countermeasure in mitigation a security incident. Examples of medium-level impact countermeasures are: monitoring systems, trace back network ID and lock all associated accounts, slow system response, etc.

2.1.2.4.3 Low-level Efficiency They are the least aggressive countermeasures from the list, since their main focus is to secure the infrastructure, and observe the event(s) in order to analyse the danger and provide statistical results to help security administrators in the appropriate reaction. The application of low-level efficiency countermeasures general results into no disruption or other side-effects. Examples of low-level impact countermeasures are: camouflage, quarantined faux systems, library control systems, honey-pot, historical traffic flow analysis, etc.

Without a working classification of security measures, identifying the possible candidates to mitigate a given attack and estimating their costs and benefits in a particular scenario becomes unrealistic, making the countermeasure selection process a very complicated task. The design of a countermeasure taxonomy helps in the decision process of analyzing and selecting candidate solutions based on cost-sensitive metrics. The next section discusses some examples of usage of a countermeasure taxonomy.

2.1.3 Countermeasure Taxonomy Usage

The proposed taxonomy is ideal to be used in the design of security ontologies. We develop an ontological knowledge base that defines the classes and properties of Security Information and Event Management (SIEM) systems [GMHD11, GMHD12], as depicted in Figure 2.2.

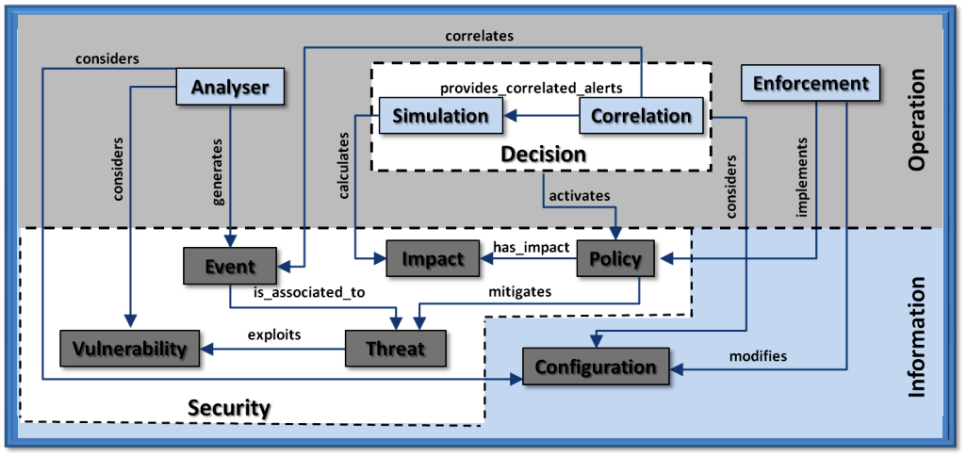

FIGURE 2.2 - Security Ontology Model

The Ontology is composed of two main classes: Information and Operations. The Information Class models all the data related to the system and network configuration, as well as security related data (e.g. vulnerabilities, threats, events, impacts, policies). The Operation Class models the treatments (correlation, simulation, enforcement) needed to provide security policies in order to mitigate the impact of intrusions or attacks. Our security taxonomy can be included in the *Policy* subclass that belongs to the Information Security class, making it easier the identification and selection of countermeasures.

Intrusion Response Systems may also benefit from this taxonomy to handle malicious activities by applying effective countermeasures to protect the system. Following the 4 dimensions of our proposed taxonomy, we provide examples of countermeasures for each category. The classification considers firstly the time during which a countermeasure is active (i.e. temporary, permanent). Secondly, it proposes a classification based on their response strategy (i.e. proactive, reactive). Thirdly, countermeasures are classified according to the compromised service (i.e. confidentiality, integrity, availability); and they are further classified based on the reduction impact (i.e. low, medium, high). Tables 2.1 and 2.2 summarizes these examples.

2.1.4 Other Approaches

It is also important to consider other approaches in order to react against a given attack. One approach that has slowly emerged during the past decade is the Intrusion Tolerance [VNC03, SBC+07, DP06, SND09, KRZ+98, VNC+06]. Verissimo et. al [VNC03, VNC+06, SBC+07] define the Intrusion Tolerance as the notion of handling (react, counteract, recover, mask) a wide set of faults encompassing intentional and malicious faults (generally called intrusions), which may lead to failure of the system security properties if nothing is done to counter their effect on the system state. Instead of trying to prevent every single intrusion, these are allowed, but tolerated due to the fact that the system has the means to trigger mechanisms to prevent the intrusion from generating a system failure.

Deswarte et al. [DP06] state that intrusion tolerance aims to organize and manage a system such that an intrusion in one part of the system has no consequence on its overall security. To do that, techniques developed in the traditional field of fault tolerance (e.g. error handling -detection and recovery-, and fault handling -diagnosis, isolation, repair, reconfiguration-) can be used. However,

there are two main problems: on the one hand, it should be made very difficult for the same type of attack to succeed in different parts of the system; and on the other hand, an intrusion into a part of the system should not allow the attacker to obtain confidential data.

The intrusion tolerance approach assumes that systems remain to a certain extent vulnerable; that attacks on components or sub-systems can happen and some will be successful; and ensures that the overall system nevertheless remains secure and operational, with a quantifiable probability [VNC03]. In other words, an AVI composite fault model should take place. The following elements composed the AVI model:

- Vulnerability: fault in a computing or communication system that can be exploited with malicious intention.
- Attack: malicious intentional fault attempted at a computing or communication system, with the intent of exploiting a vulnerability in that system, which then leads to an intrusion.
- Intrusion: a malicious operational fault resulting from a successful attack on a vulnerability.

Intrusion tolerance strategies derive from a confluence of classical fault tolerance and security strategies. These strategies should consider the following [VNC03]:

- Fault Avoidance vs. Fault Tolerance
- Confidential Operation
- Perfect Non-stop Operation
- Reconfigurable Operation
- Recoverable Operation
- Fail-Safe

Besides the several advantages associated to the intrusion tolerance model, the approach has some limitations, for instance, strategies are conditioned by several factors, such as: type of operation, classes of failures (i.e., power of intruder); cost of failure (i.e., limits to the accepted risk); performance; cost; available technology. However, the intrusion tolerance remains an interesting approach that can be combined with other approaches in the intrusion/attack detection and reaction process. Our approach, for instance, concentrates in the domains of "Reconfigurable Operation" and "Recoverable Operation" by proposing tools to improve reaction strategies.

2.1.5 Discussion

Some of the countermeasures in our taxonomy belong to all the security services (i.e. Confidentiality, Integrity, Availability), due to the fact that the degree of granularity given in the example does not allow to specify which type of service is affected by the given countermeasure. For instance, enabling additional firewall rules, implementing abnormal behavior rules, and cleaning a virus from an infected file, are countermeasures that can be used to mitigate intrusions or attacks that affects all the services from the system.

It is important to note that due to the objective at which the security measure has been designed, proactive countermeasures are typically thought of permanent solutions, whereas reactive countermeasures are conceived as temporary solutions. This is why, in our taxonomy there are more examples of long-term proactive countermeasures than long-term reactive countermeasures, whereas for short-term countermeasures there are more examples on the reactive category than what we can find on the proactive category.

TABLE 2.1 - Temporary Countermeasure Examples

		Low	Medium	High
Proactive	**Confidentiality**	audit trials, camouflage, quarantined faux systems, controlled faux systems	monitoring systems, authorization policy servers, honey-pots	one-time password, proper authentication methods, expiring password and sessions
	Integrity	audit trials, quarantined faux systems, controlled faux systems	monitoring systems	activation/ deactivation of anti-spoofing rules at the network boundary, reinforce encryption procedures
	Availability	audit trials, quarantined faux systems, controlled faux systems, enable intrusion activity tools	monitoring systems, enable local/remote network activity logging, activate/ deactivate alert mode, activate/deactivate account surveillance	add obstacles in the response, trace connection for information gathering, monitoring configuration changes
Reactive	**Confidentiality**	restrict user activity, restart/terminate suspicious process, re-authenticate user/remote system, activate/deactivate multiple transaction requests, increase/decrease number of transactions per user per day	repeat challenge response, lock local host account, trace back network ID and lock all associated accounts, update firewall rules to block an attacking IP address, deny full/selective access to files, enable/disable user account, increase/decrease amount of transactions	lock entire host system, procedure to clean a virus from an infected file, block suspicious incoming/outgoing network connections, block ports/IP addresses, deactivation of user accounts, deny/allow transactions, disable URL rewriting
	Integrity	restart/terminate suspicious process	trace back network ID and lock out all associated accounts, enable/disable additional firewall rules	procedure to clean a virus from an infected file, disable URL rewriting, allow SSL/TLS session identifier, generate new session identifier SID
	Availability	swallow offending packets, allow to operate on fake files, restart/terminate suspicious process, delay suspicious system calls, restart/terminate targeted system	restrict connection to the system, trace back network ID and lock out all associated accounts, enable/disable additional firewall rules, slow system response, deny/redirect requests	disconnect from network, disconnect network from all outside access, procedure to clean a virus from an infected file, shut down compromised service host, disable URL rewriting

TABLE 2.2 - Permanent Countermeasure Examples

		Low	Medium	High
Proactive	**Confidentiality**	library control systems, review high profile target system	privacy and unpredictable passwords, sensors and alarms, security policies and procedures, digital signatures, digital certificates, vulnerability scanners, security protocols, authentication tokens, update system patches	encryption, proper authentication methods, anti-virus protection, IDS/IPS, access control software, cryptographic cards, database security access controls, anti-spoofing rules at network boundary
	Integrity	library control systems, review boundary security policies to ensure outbound packets are restricted appropriately	password audits, encoded passwords, sensors and alarms, security policies and procedures, vulnerability scanners, security protocols, message authentication capabilities, update system patches	hashed-program files, biometric data encryption, public key certificate, anti-virus protection, IDS/IPS, database encrypted data storage, anti-spoofing rules at network boundary
	Availability	library control systems, network segmentation, preallocate resources to respond to security incidents, perform historical traffic flow analysis	back-up information, security policies and procedures, vulnerability scanners, security protocols, hardened OS, port security mechanisms, automatically expiring URLs, update system patches	anti-virus protection, IDS/IPS, keep traffic for longer period of times, Network Access Control at local hosts, system behavior monitoring
Reactive	**Confidentiality**	data restorage, delete tampered with files, allow to operate on fake files, allow only national transactions, egress/ingress filtering	enforce access controls, trace connection to perform attacker isolation/quarantine, create remote decoy, abnormal behavior rules, enforce privilege separation	firewall control, activate/deactivate multiple-factor authentication, limit interaction between system and the Internet
	Integrity	data restorage	implement abnormal behavior rules	firewall controls, limit interaction between system and the Internet
	Availability	contingency/emergency plans, data restorage, restore tampered with files from backup, remote accessing	limit access to the system, implement firewall control, disaster recovery, abnormal behavior rules, black-holing the victim host/network	limit interaction between system and the Internet

Most of the countermeasures that fall into the temporary and reactive category (e.g., restrict user activity, disable URL rewriting, deny/redirect requests, etc), also fall into the immediate category, since they are meant to be implemented as soon as the security incident is detected. Similarly, most of the countermeasures that fall into the permanent and proactive category (e.g. vulnerability scanners, IDS/IPS, anti-virus protection, etc) also fall into the deferred category if they are not deployed on the system at the time they are required.

Since countermeasures in our taxonomy are designed to be selected based on cost-sensitive metrics (e.g. RORI index), there is no need to add a selection mechanism category (e.g. static, dynamic, cost-sensitive) as proposed by Stakhanova et al. [SBW07b]. However, categories such as ability to adjust, and cooperation ability are suitable to be used as an extension of our approach.

2.2 Cost Sensitive Models

Cost sensitive metrics are widely proposed as a viable approach to find an optimal balance between intrusion damages and response costs, and to guarantee the choice of the most appropriate response without sacrificing the system functionalities. Measurements are either absolute or relative. Absolute measurements use precise values that scale with a given unit (e.g. hundreds, thousands, millions, etc); whereas relative measurements are methods for deriving ratio scales from paired comparisons represented by absolute numbers [Saa93]. Relative measurements are useful in obtaining an overall ratio scale ranking of the alternatives. If the ratio produces repeatable and consistent results, the model can be used to compare security solutions based on relative values [SAS06]. In this dissertation, we use the terms: metric, model, ratio, and index interchangeably. The reminder of this section presents absolute metrics (e.g. Net Present Value-NPV- [Pua09]) and relative metrics (e.g. Return On Investmenet -ROI- [Jef04]) classified into two main groups: Financial models and Information Security models.

2.2.1 Financial Models

A Financial model is a measurement of two selected numerical values taken from a company's financial statement. The measure provides a valuable information to compare an enterprise progress against pre-established goals, competitors, and the overall industry [BW00]. Examples of these models include the Net Present Value (NPV), the Internal Rate of Return (IRR), and the Return On Investment (ROI). This section introduces the aforementioned models since they provide valuable information related to our contribution.

2.2.1.1 Net Present Value (NPV)

The Net Present Value (NPV) is an absolute measure that compares costs and benefits over a period of time, making it possible to analyze long-term investments [Pua09]. The NPV allows to discount all expected costs and benefits from an investment to its present value, taking into account the time value of money. The NPV is computed as the sum of the total present value of benefits and (operating) costs for each period of time minus the initially required (configuration) costs, as depicted in Equation 2.1.

$$NPV = \sum_{t=0}^{T} \frac{B_t - C_t}{(1+r)^t} \tag{2.1}$$

Where:

B_t refers to all benefits during period t,

C_t refers to all costs during period t,

r is the internal rate of discount,

T is the total number of periods for the project.

The resulting NPV is used in the decision making process. A project is profitable, if the NPV exceeds zero (i.e. the investment is at least as profitable as an alternative investment). A positive NPV indicates that the investment generates profit and therefore should be accepted. A negative NPV, on the contrary, means that the initial investment is greater than the present value of the expected cash flows. Investments in projects with negative NPVs should not be made, because they do not add value to the firm and generate a loss [Jef04] .

2.2.1.2 Internal Rate of Return (IRR)

The Internal Rate of Return (IRR) is a relative measure of the anticipated performance of a project. It considers the compounded annual rate of return the project is expected to generate [Cru08]. This metric is related to the NPV of the project, since the NPV is the discount rate at which the NPV of the project is zero. In other words, the IRR is the average discount rate where the cash benefits and costs exactly cancel [Jef04, Pua09]. Equation 2.2 depicts the formula to calculate the IRR.

$$NPV = \sum_{t=0}^{T} \frac{B_t - C_t}{(1 + IRR)^t} = 0 \tag{2.2}$$

Where:

B_t refers to all benefits during period t,

C_t refers to all costs during period t,

T is the total number of periods for the project.

IRR is the internal rate of return

As a result, an investment is accepted for projects offering a rate of return greater than their discount rate (equivalent to a positive NPV). An investment is discarded when the IRR is lower than the project discount rate, since investing in the project will reduce the value of the organization. The IRR is often used for long-term investments, hence this is only one factor to consider in a technology investment decision.

2.2.1.3 Return On Investment (ROI)

The simplest and most used approach for evaluating financial consequences of business investments, decisions and/or actions is the Return On Investment (ROI) metric. The ROI index is a relative measure that compares the benefits versus the costs obtained for a given investment [Jef04, Sch11, Pua09, Sch04].

ROI basically shows how much a company earns from invested money. This metric supports decision makers to select the option(s) that have the highest return. ROI is calculated as the present value of accumulated net benefits over a certain time period minus the initial costs of investment, then divided by the initial costs of investment, as shown in Equation 2.3.

$$ROI = \frac{B_t - C_t}{C_t} \times 100 \tag{2.3}$$

Where:

B_t refers to all benefits during period t,

C_t refers to all costs during period t

The decision rule is that the higher the ROI value, the more interesting the investment. However, Jeffery and Schechter [Jef04, Sch04] agree that the major problem with ROI is that the metric does not include the time value of money, i.e. a 100% ROI realized 1 year from today is more valuable than a 100% ROI realized in 5 years. Furthermore, the costs and benefits of the project may vary over time, meaning that the cash flows are different in each time period. As a result, ROI is not a convenient way to compare projects when the costs and benefits vary with time, and it is also not useful for comparing projects that will run over different periods of time.

2.2.2 Information Security Models

The economic approach for information security is closely related to the concepts of investment and return. This latter is commonly referred to as the amount of losses that are avoided due to a security investment; losses that were expected to occur had these investments not been applied [Khe10]. In order to evaluate information security investments, several models have been proposed, i.e. Return On Attack [CM05], Return On Security Investment [SAS06], Return On Response Investment [KCBCD10]. This section details each of the aforementioned models as they are closely related to our contribution.

2.2.2.1 Return On Attack (ROA)

The Return On Attack (ROA) is a relative measure that evaluates the gain the attacker expects from a successful attack over the losses that he sustains due to the adoption of countermeasures by his target [CM05]. It is important to highlight that the ROA ratio is the evaluation an organization does about the effectiveness of a countermeasure in preventing or discouraging certain class of intrusion attempts assuming some profiles of potential attackers. Equation 2.4 shows this metric.

$$ROI = \frac{GI}{CA} \times (1 - EFF) \tag{2.4}$$

Where:

GI refers to the expected gain from the incident,

CA refers to the perceived cost sustained by the attacker to succeed,

EFF is the efficiency of the attacker to violate countermeasures

As a result, the lower the ROA value, the more interesting the security investment. By using ROA, it is possible to select the most appropriate security investment, especially in situations where different technologies are combined or where the possible degradation of a security solution's efficiency over time should be taken into account.

2.2.2.2 Return On Security Investment (ROSI)

The Return On Security Investment (ROSI) is a relative metric that compares the differences between the damages originated by attacks (with and without countermeasures) against the cost of the countermeasure [Con04, Loc05, SAS06, BSB07, Pua09, Kos11]. To calculate ROSI, a formula adapted from the ROI metric is presented in Equation 2.5.

$$ROSI = \frac{(ALE_b - ALE_a) - Cost_{CM}}{Cost_{CM}} \times 100 \tag{2.5}$$

Where:

ALE_b refers to the annual loss expectancy before countermeasure,

ALE_a refers to the annual loss expectancy after countermeasure,

$Cost_{CM}$ is the cost of the countermeasure

The calculation of each parameter composing the ROSI equation has been widely discussed by Lockstep Consulting [Con04], and Kosutic [Kos11]. The former proposes a methodology that considers different levels of likelihood and severity, which are then, respectively transformed into frequency and direct cost; the latter considers on the one hand, parameters associated to the incident (e.g. financial losses, costs, frequency, etc.), and on the other hand, parameters associated to the protection (e.g. cost, benefits, life expectancy of the security measure, etc).

Similar to the ROI metric, the decision rule is that the higher the ROSI value, the more interesting the investment.

ROSI variants: In the process of finding a metric that evaluates different security investments and provides accurate and meaningful results to be used in the selection process, the Return On Security Investment (ROSI) has evolved to introduce other parameters in its calculation.

Kim et al. [KLI08] introduce the concept of a higher ROSI (maximizing the ROSI value) for an effective security investment by investigating the correlation between a particular threat and a damage type. The process concentrates on finding the factors that contribute to efficient control of the security threats instead of focusing on the process of return on security investment. By prioritizing security measures for selection, the ROSI metric is estimated more accurately, and therefore the value of a security investment is maximized.

Mizzi [Miz05,Miz10] proposes an adaptation to ROSI called the Return On Information Security Investment (ROISI), which introduces the concept of motivation to an attack, successfulness of an attack, and viability of expenditure, making it possible to quantify the total cost of the portion of information assets that may be lost due to intrusions or attacks. However, according to Locher [Loc05], ROSI would lead to proper results, if the risk mitigation effects are calculated properly with scenario analysis and expected values. It has been matured in various models trying to consider qualitative variables, but it is doubtful, if variables like criticality, exposure, and vulnerability, help to improve the ROSI concept.

2.2.2.3 Return On Response Investment (RORI)

The Return On Response Investement (RORI) is a service dependency model for cost sensitive response based on a financial comparison of the response alternatives [KCBCD10, Khe10]. RORI is an adaptation of the ROSI index that provides a qualitative comparison of response candidates against an intrusion. The parameters that constitute this index are derived from the ROSI

parameters by drawing an analogy between costs for intrusion prevention and response.

The RORI index considers not only response collateral damages but also response effects on intrusions, as depicted in Equation 2.6.

$$RORI = \frac{[IC_b - RC] - OC}{CD + OC} \times 100 \tag{2.6}$$

Where:

IC_b represents intrusion impacts when no response is enforced,

RC refers to the combined impact of intrusion and response,

OC are operational costs that cover low level investments such as response setup and deployment costs

CD refers to collateral damages, which are costs that are added by a new response, and are not related to intrusion costs

In contrast with the OC parameter, both ICb, RC and CD parameters are directly associated with the intrusion and response impacts on the security objectives of the target system. The values of these parameters, for the same intrusion and response combinations, depend on the current service configuration. They must be evaluated during system runtime, as soon as new intrusions are detected and new candidate responses are proposed. The ultimate goal is to select the candidate response set that provides a maximal positive RORI index.

2.2.3 Discussion

Each of the aforementioned cost-sensitive metrics provides its own benefits and limitations, which are summarized in Table 2.3. We discuss in this section the characteristics of the different cost-sensitive models presented in this chapter based on 5 factors: accuracy, time value of money, payback period, collateral damage, and corner cases such as the strategy of applying no operation (NOOP).

Accuracy: A main limitation in most of the studied metrics is accuracy. There are many assumptions taken while analyzing the investment. It is therefore practically impossible that all assumptions will be exactly correct. Results are as accurate as the forecasts of loss event frequencies on which they rely, and today these forecasts use best guesses rather than quantitative models. Estimating soft benefits parameters (e.g., fewer errors, reduced processing time, improved customer satisfaction, etc) to calculate the Net Present Value or the Internal Rate of Return of a given project is extremely difficult to accurately quantify [Jef04].

In addition, a great level of subjectivity is considered while estimating parameters such as benefits and importance of the investment in the Return On Investment (ROI) model. Furthermore, it is very difficult to be accurate in predicting the attackers's behavior, an important parameter to calculate the Return On Attack (ROA). Similarly, the ROSI and RORI metrics rely on parameters such as the costs and benefits of a security solution. In general, the costs of countermeasures are rather easily defined, in contrast with their benefits, since it requires predictions of an event that has not yet occurred. However, this issue is of less importance when the ratio is used for relative comparisons.

Time value of money: While the Return On Investment (ROI) presents a percentage of return of an investment over a defined period of time, the Internal Rate of Return (IRR) does not inform

about the absolute value of such investment. The Net Present Value (NPV) is the only approach informing about the absolute value of a project. The ROA, ROI, ROSI and RORI models face a problem in the case of long-term investments because they do not consider the time value of money. A return of 100% of the investment realized 1 year from today is more valuable than the same return realized in 5 years; therefore a decision maker would need the NPV as well to justify investment opportunities.

Although NPV covers the time value of money, this latter has limited information, since it only considers the interest or inflation over a given period of time. IRR has a doubtful assumption because it assumes the same rate of return for the whole period of time. NPV has advantages within pre-investment analysis while all variants of the ROI metric are best for the ongoing assessment of investment profitability [Loc05].

Payback period: The payback period is calculated by cumulatively summing the net cash flows of a project. When the sign of the cumulative sum of the net cash flows changes from negative to positive the project has paid back the initial investment. In making investment decisions, projects with high IRR and short payback periods are more often selected. [Jef04]

The NPV and IRR models depend directly on the time period of the evaluation (results change if the time period of the analysis is reduced or extended). For instance, the Internal Rate of Return may increase significantly if the time period of the project is extended to 5 years instead of 2 years. The payback period for these metrics is directly affected by the selected period of time of the whole project.

Since the time value of money is not considered in metrics such as ROA, ROI, ROSI and RORI, they are not useful for comparing projects that run over different periods of time. Gordon and Loeb [GL02] support the use of the IRR, since this latter incorporates discounted cash flows for investments that have different costs and benefits in different years. However, Gordon and Loeb state that rates of return should not be used when comparing two investments, as an investment may have a greater net benefit but lesser rate of return. If enough cash is available to invest in either of the options, but an organization can only invest in one of them, it would be less profitable to choose the investment with the higher IRR over that with the greater NPV.

Collateral damage: None of the cost-sensitive metrics described in this section, except for the Return On Response Investment, consider collateral damage as a parameter in their calculation. Collateral damages depend on both response mechanism and the current state of the target system. By modifying configurations, the response affects users of the target system, and thus resulting in collateral damages. Although they provoke mostly negative costs, response collateral damages are unlikely to be avoided when reacting against an intrusion attempt [Khe10].

The collateral damage parameter proposed to calculate the RORI index [KCBCD10, Khe10] is directly associated with the intrusion and response impacts on the security objectives of the target system. However, due to difficulty in its calculation, collateral damage is combined with parameters that consider intrusion impacts which are not contained by the selected response.

No operation (NOOP): A reasonable strategy when the mitigation cost is greater than the benefits it provides to the system, is to accept the risk. Acceptance is the choice of executing no operation (-NOOP- for short) against a security incident. From the previously described cost sensitive metrics, none of them are useful in evaluating NOOP.

All the financial metrics defined in this chapter (i.e. NPV, IRR, ROI) consider the fact that at least one security measure is chosen to be evaluated. The Return On Attack (ROA) focuses on the attack behavior and therefore assumes that some gain and costs are associated to the incident. The

Return On Security Investment considers the cost of the countermeasure in both, the numerator and denominator of the equation, leading to an indetermination in case of NOOP.

Although the RORI index proposes a parameter that represents intrusion impacts when no response is enforced, the formula execution leads to a result of -100% in all of the cases, meaning that regardless of the studied attack, we should expect 100% of the losses while evaluating NOOP. As a result, it is useless to compare NOOP against all other security options since it is not possible to find a countermeasure with a Return On Response Investment inferior to -100%.

TABLE 2.3 - Summary of Cost Sensitive Models

Models	Main Focus	Optimal Solution	Advantages	Disadvantages
NPV	Benefits, Costs, Rate of discount, Length of investment	$NPV \geqslant 0$	Good for long-term projects	Useful if combined with other metrics; Unable to evaluate collateral damage and NOOP
IRR	Benefits, Costs, Length of investment, Rate of return	IRR > Project discount rate	Incorporate discounted cash flows for investment that have costs and benefits in different years	Same rate of return for the whole investment period; Unable to evaluate collateral damage and NOOP
ROI	Security Solution Cost, Effectiveness	Highest ROI value	Good for ongoing assessment of investment profitability	Useless for long-term investments; Time value of money is not considered; Unable to evaluate collateral damage and NOOP
ROA	Attack Gain , Attack Cost , Losses due to Security	Lowest ROA value	Evaluate the impact of security solutions based on the attacker's behaviour	Not accurate while predicting attacker's behaviour; Useless for long-term investments; Unable to evaluate collateral damage and NOOP
ROSI	Security Investment , Benefits and Cost	Highest ROSI index	Compare the difference between damages of IT incidents (with and without countermeasures) against the solution cost	Unable to evaluate collateral damage and NOOP; Useless for long-term investments; Time value of money is not considered
RORI	Collateral Damage, Operational Costs and Response Costs	Highest RORI index	Determine the benefit obtained in a particular threat scenario that applies a given countermeasure	Useless for long-term investments; Time value of money is not considered

2.3 Countermeasure Selection Methodologies

Research in the selection of appropriate countermeasures to mitigate the impacts of attacks is still in progress. Bistarelli et al. [BFP07,BFPT08] have proposed qualitative methods (e.g. defense trees and conditional preference networks), while Duan et al. [DCH06], Cavusoglu et al. [CMR04], and Ferenc and Salim [FS09a], suggest quantitative methods (e.g. genetic algorithm , game theory) that

use cost sensitive metrics (explained in Section 2.2) to evaluate, rank and select countermeasures. This section details and classifies these methods into two approaches: qualitative and quantitative.

2.3.1 Qualitative Approaches

Qualitative approaches (e.g. defense trees and conditional preference networks [BFP07, BFPT08]) have been proposed to evaluate and select countermeasures based on expert knowledge, organization's objective, and other useful criteria. The selection process does not generally rely on cost sensitive metrics, but they can use numerical data to decide upon several candidates. In addition, the degree of automation is generally low, meaning that in most cases, these are stactic methods that requires the human intervention to select the countermeasure. The reminder of this section details some of these approaches.

2.3.1.1 Defense Trees and Conditional Preference Networks

Bistarelli et al. [BFP07, BFPT08] propose an approach that uses two qualitative instruments for the selection of defense strategies to protect an IT system from the risk of attacks. The first approach is the use of defense trees to model attack/defense scenarios, and the second approach is the use of Conditional Preference Networks (CP-nets) to model qualitative conditional preference over attacks and countermeasures.

Defense tress are an extension of attack trees that represent an attack against a system and the way it can be mitigated by a set of countermeasures. The main difference between attack and defense trees is that the former represents only the action that an attacker can perform, while the latter adds the set of countermeasures that can be introduced into the system to mitigate the possible damages produced by an attack action.

Conditional preferences networks (CP-net for short) are a graphical formalism that specifies and represents qualitative conditional preference relations. CP-nets capture preference statements that are able to express a conditional preference over some variables. The following definition is proposed [BFP07]:

> A CP-net is a directed graph N = (V,E), where V = $x_1, ..., x_n$ is a set of variables and E = $(x_i, x_j) : x_i, x_j \epsilon V$ is a set of edges between variables. The function $P_a(x)$ gives for each node x ϵ V , the node x' ϵ V s.t. (x', x) ϵ E. The conditional preference table of the CP-net describes a strict partial order $(D(x_i), \succ_i^u)$ where $D(x_i)$ is the domain of the variable x_i, and $\succ_i^u$ represents the conditional preference of the instantiations of variable x_i given an instantiation u of the variable $P_a(x_i)$.

The conditional preference table is specified by the system administrator based on expert knowledge and statistical information. As a result, it is possible to determine, in a qualitative manner, the attack strategies that an attacker may follow to damage a system, the different actions that compose each attack and the countermeasures that a system administrator can implement on the system.

2.3.1.2 Multi-objective Selection

The multi-objective countermeasure selection approach proposed by Neubauer et al. [NSW06, NP10] is an organizational process that provides a structured and repeatable methodology that includes the following steps:

- Evaluation criteria according to the organization's strategy
- Assessment of the existing IT security infrastructure
- Identification of Stakeholder preferences
- Determination of the solution space of all efficient countermeasures
- Selection of the individual best countermeasure

In addition, the approach takes into account interdependencies among security countermeasures and provides an environment for multiple users. A moderator is required to provide advance and professional support during the workshop and the interactive selection allows decision makers to playfully explore the alternative that matches their preferences. The workshop is divided into two sections that are performed in a full-day meeting. The first section consists on the assessment of the existing countermeasure portfolios and the subsequent generation of promising solutions. The second section pretends to iteratively reduce the number of portfolios until identifying the portfolio that best fits the stakeholder's objective.

The portfolio presents the different countermeasures or combinations of them according to multiple criteria (e.g. monetary value, accept cost, setup costs, setup time, etc.) As a result, the approach serves as a valuable tool to improve security awareness of top management as they may run through different scenarios and potential solutions which should decrease the probability to overlook relevant risks.

2.3.1.3 Threat Tree

Bedi et al. [BGS+11] proposes an approach that uses threat tree to select optimal countermeasures. Threat modeling involves understanding adversaries' goals in attacking a system based on system's assets of interest. The process consists of decomposing the application, identifying, ranking, and mitigating threats. In order to identify threats it is necessary to go through each of the security critical entities and creating threat hypotheses that violate confidentiality, integrity, or availability of the entity. Threat trees are then designed for each threat requiring mitigation to analyze the threat through attack paths. The root node of a tree is the threat, each leaf node is an attack to accomplish the threat and the path from leaf to root is the way an attacker achieves the threat, as depicted in Figure 2.3.

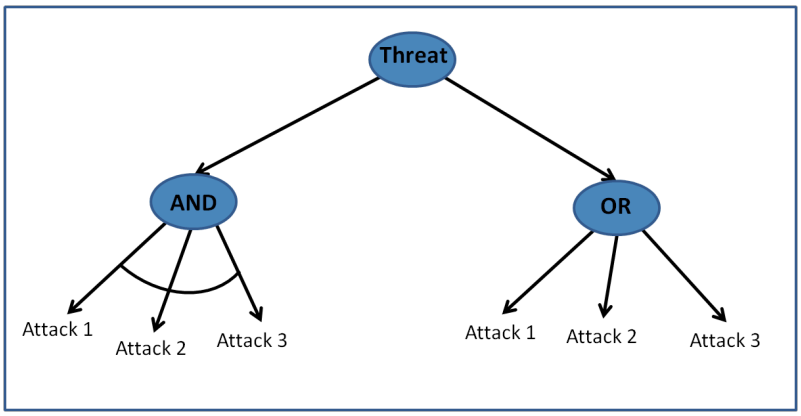

FIGURE 2.3 - Bedi et al. Threat Tree

In Figure 2.3, the AND refinement means that in order to occur the root threat, all the corresponding attack must occur, whereas, the OR refinement means that in order to occur the root

threat, at least one attack should occur.

The approach to select optimal countermeasures needs to be adopted at the design phase of software life-cycle. The threat that needs to be mitigated is at the root node. The attacks at leaf nodes can not be refined further as they are executed by the attacker to accomplish the threat. Countermeasures are therefore applied against the attacks at leaf nodes to prune the attack branches from the threat tree to avoid the threat at the root level.

The solution has been designed to generate a multi-threat attack graph by combining all the individual threats responsible for the security compromise of the system and removing duplicate nodes in multiple threat trees. This graph gives a unique set of attacks requiring mitigation as output. In addition, the solution prioritizes the identified attacks for mitigation on the basis of frequency and the expected damage the threats can generate to the system. Some of the attacks having threat_index less than the associated threshold value are ignored for mitigation, making this approach economical for software security. As a result, the mechanisms is proven to optimally save the system from being compromised.

2.3.2 Quantitative Approaches

Quantitative approach consists of those studies in which the data concerned is analysed in terms of numbers. Quantitative methods (e.g. genetic algorithm [DCH06], game theory [CMR04, FS09a]) generally use one or several cost sensitive metrics (explained in Section 2.2) to perform the evaluation and selection of countermeasures. They are typically dynamic, and the degree of automation is higher than the one presented in qualitative methods. The reminder of this section explains some of these approaches.

2.3.2.1 Game Theory

Cavusoglu et al. [CMR04] and Ferenc et al. [FS09a] have used Game Theory to evaluate and select countermeasures for a given attack. Game theory is used to analyze problems in which the payoffs to players depend on the interaction between players' strategies. The analogy in the IT security investment environment is that firms and hackers are players. The firm's payoff from security investment depends on the extent of hacking it is subjected to. The hacker's payoff from hacking depends on the likelihood of being caught, which, in turn, depends on the level of investment the firm makes in IT security. The first step in using game theory to analyze such strategic interactions among players is to develop a game tree that depicts the strategies of players.

The game starts by selecting the type of traffic to the system, which can be external (with probability ε) or internal (with probability $1-\varepsilon$). A given node represents external users that can be either authorized or unauthorized. Similarly, one node characterizes internal users that can be either honest or dishonest. A dishonest user can take two actions: hack or not to hack. If the hacker decides to hack, the game moves to the following node. The firm makes decisions about whether or not to monitor based on the state (signal or no-signal). The firm must make decisions without knowing exactly which node the game has reached. However, it can determine the probability of intrusion in the signal and no signal states using Bayes Rule as illustrated in Equation 2.7.

$$\begin{aligned} P(i,s) &= \frac{P(s,i)P(i)}{P(s,i)P(i)+P(s,\bar{i})P(\bar{i})} \\ P(i,\bar{s}) &= \frac{P(\bar{s},i)P(i)}{P(\bar{s},i)P(i)+P(\bar{s},\bar{i})P(\bar{i})} \end{aligned} \tag{2.7}$$

Where:

$P(i, s)$ Probability of intrusion given signal,

$P(i, \bar{s})$ Probability of intrusion given no-signal,

$P(i)$ Probability of intrusion

$P(s)$ Probability of signal

$P(s, i)$ Probability of signal given intrusion

$P(\bar{s}, i)$ Probability of no-signal given intrusion ,

$P(\bar{s}, \bar{i})$ Probability of no-signal given no-intrusion

As a result, the model can be used in a variety of ways, for instance, it can be used to select a specific security measure. Similarly, the model is used as a what-if analysis tool to explore different options and evaluate the effect of a given parameter in the countermeasure selection.

2.3.2.2 Genetic Algorithms

Genetic Algorithms have been shown to work in many large complex search problems with affordable space and time requirement [DCH06]. The idea of this approach is inspired on a biological metaphor that searching could be viewed as a competition among a population of evolving candidate problem solutions (represented by chromosomes). Therefore, through operations analogous to gene transfer in sexual reproduction, solutions from one population are taken and used to form a new population by means of operators such as crossover and mutation. A fitness function evaluates each solution to decide whether it is capable of contributing to the next generation. The underlying hope is new population is better than the old one, thus sufficient evolution would ultimately lead to an optimized solution.

The approach evaluates the performance of selected security measures through Equation 2.8

$$NPV = Cost^{dev} + (ALS - Cost^{op}) \times \sum_{i=1}^{n} \frac{1}{(1+r)^{i-1}} \tag{2.8}$$

Where:

NPV represents the Net Present Value,

$Cost^{dev}$ refers to development costs,

ALS is the Annual Loss Savings that results of deploying a set of countermeasures,

$Cost^{op}$ refers to operational costs,

n is the number of periods under consideration,

r represents the discount rate

The genetic algorithm approach allows to specify an upper limit on acceptable unmitigated ALE; and the countermeasure failure, to find a solution that provides enough protection even under failure conditions. As a result, the genetic algorithm is able to find the best countermeasure combination in all studied cases.

2.3.2.3 Decision Matrix

Whenever several solutions are presented to mitigate a given threat or attack, a decision must be taken to select the most convenient countermeasure, whether the organization prefers the solution with the lowest cost or the most effective one, it is not always easy to reach a consensus. Norman [Nor10] proposes a decision matrix to help security administrators in deciding upon the most appropriate countermeasure to implement. This tool was designed to prevent terrorism attacks and other physical attacks, but it can be extended to other domains such as the Information Technology. The decision matrix lays out the goals, risks, costs, and several other factors. In addition, countermeasures are scored based on their costs, their ability to achieve goals, and to mitigate threats, as depicted in Figure 2.4.

	Goals Achieved						Risk Mitigated							Accepted Risk					Estimated		
Countermeasure Methods	1	2	3	4	5	6	A	B	C	D	E	Score	Rank	A	B	C	D	E	Cost	Effectiveness	Convenience
Fence entire property	1			1	1		1	1	1	1		7	2					x	400,000	High	High
Fence parking lots and garage			1	1	1		1					4	3		x	x	x	x	100,000	Low	High
Use landscaping to create a barrier				1	1	1						3	4	x	x	x	x	x	400,000	Low	High
Use landscaping and fencing to enclose property and deny access	1	1	1	1	1	1	1	1	1	1		10	1					x	500,000	High	High

Goals Description

1. To deny access to unauthorized persons
2. To create a pleasant and visually pleasing environment
3. Cost effectiveness based on goals and threats mitigated or eliminated
4. Convenience for employees
5. Conformance to business culture
6. Conformance to aesthetic values

Risk Description

A. Harmless unexpected visitor
B. Unauthorized external visitor
C. Property criminal
D. Personal or sexual attack criminal
E. Workplace violence visitor

FIGURE 2.4 - Norman T. Decision Matrix

The matrix presented in Figure 2.4 shows the risks an organization is willing to accept if a given countermeasure is selected. The construction of this matrix starts by listing the goals of the countermeasures (e.g. access control, deterrence, detection, assessment, delay), numbered from 1 to N. This is followed by listing the possible risks for the countermeasure to mitigate (e.g. confidentiality, integrity, availability), lettered from A to X. Then, we list the countermeasure methods in rows as well as goals achieved, score, rank, accepted risks, estimated costs, effectiveness and convenience. As a result, security analysts are able to select, optimal countermeasures based on a multi-factor evaluation matrix.

It is important to note that we can add as many goals and risks as possible to evaluate the different countermeasures. Scores are based on highest number of goals achieved and risks mitigated or eliminated. The ranking column is bases on the highest score, and the columns for costs, effectiveness and convenience are estimations based on expert knowledge.

2.3.2.4 Conflicting Incentives Risk Analysis (CIRA)

The CIRA method models risk in terms of conflicting incentives between the risk owner and other stakeholders in regards to the execution of actions [RS12, RS13]. The method uses a variety of metrics to calculate the utility factor of the risk owner and the stakeholders (e.g. privacy, usability, compliance, availability, wealth, etc). The whole process takes about 21 hours (in average), and is performed as follows:

1. Identify the risk owner
2. Identify the risk owner's key utility factors

3. Give an intuition of the scope/system-identify the kind of strategies/operations that can potentially influence the above utility factors
4. Identify roles/functions that may have the opportunities and capabilities to perform these operations
5. Identified the named strategy owner(s) that can take on this role
6. Identify the utility factor of interest to the strategy owner(s)
7. Determine how the utility factors can be operationalized
8. Determine how the utility factors are weighted by each of the stakeholders
9. Determine how the various operations result in changes to the utility factors for each of the stakeholders
10. Estimate the utility of each stakeholder
11. Compute the incentives
12. Determine risk
13. Evaluate risk

As a result, the CIRA method is able to analyze risks in a non-trivial setting by helping analysts to get a better understanding of the risks. A graph that shows acceptable and unacceptable risks is drawn along with the channels considered as sensitive risk areas (Figure 2.5). Strategies are executing in decreasing order of utility as perceived by each of the strategy owners.

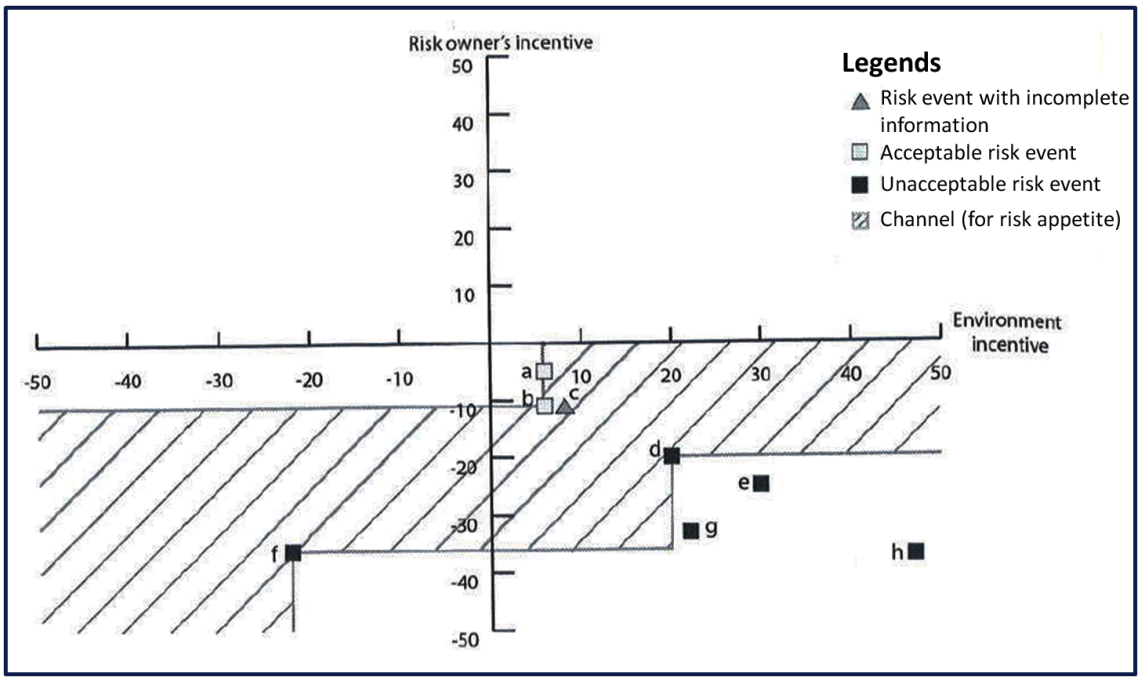

FIGURE 2.5 - Rajbhandari and Snekkenes. The Incentive Graph

In the example shown in Figure 2.5 risks 'a' and 'b' are acceptable, risk 'c' does not provide enough information to be assessed, and risks from 'd' to 'h' are unacceptable. The channel for risk appetite is therefore drawn on quadrants 3 and 4 of the coordinate system.

2.4 Conclusions

In this chapter we presented a state of the art in intrusion and threat response models and techniques, which focuses on three main aspects: (i) *a countermeasure taxonomy*, that considers the limitations of current response taxonomies to propose an approach based on four dimensions: the countermeasure strategy, the service property, the response time, and the countermeasure's impact; (ii) *cost sensitive models*, that separate financial from information security models, and classifies them in absolute or relative measurements; and (iii) *countermeasure selection methodologies*, that present the current research in the selection of appropriate countermeasures to mitigate the impacts of intrusions or attacks, and classifies them into two approaches: qualitative and quantitative.

Due to the multiple shortcomings, the existing methodologies to select appropriate countermeasures are still very limited. So far, they have been used to evaluate individual countermeasures, no attempt has been made to evaluate multiple countermeasures simultaneously. Therefore, we will first propose a cost sensitive model to evaluate individual and combined countermeasures and to select the candidate that provides the highest benefit to the organization. For this purpose, we will analyse several parameters related to the attack, the countermeasure, and the security infrastructure; and we will propose a complete methodology to estimate and evaluate them for individual attack scenarios. We will further provide an approach to analyse and select countermeasures in a scenario of multiple attacks, which takes into account the notion of the attack surface to study the volume covered by each individual attack and the selected countermeasure.

Part I

Countermeasure Selection for Individual Attack Scenarios

Chapter 3

Comprehensive Individual Countermeasure Selection

> Measurement is the first step that leads to control and eventually to improvement. If you can't measure something, you can't understand it. If you can't understand it, you can't control it. If you can't control it, you can't improve it.
>
> *H. James Harrington*
>
> -

Contents

Research in SIEM technologies has traditionally focused on providing a comprehensive interpretation of threats, in particular to evaluate their importance and prioritize responses accordingly. However, in many cases, threat responses still require humans to carry out the analysis and make a decision e.g., understanding the threats, defining the appropriate countermeasures and deploying them. This is a slow and costly process, requiring a high level of expertise, and remaining error-prone nonetheless. Thus, recent research in SIEM technologies has focused on the ability to automate the process of selecting and deploying countermeasures.

Debar et al. [DTCCB07], and Riveiro et al. [RGF+10] propose automatic response mechanisms, such as the adaptation of security policies, to overcome the limitations of static or manual response. Although these approaches improve the reaction process (making it faster and/or more efficient), they remain limited since these solutions do not analyse the impact of the selected countermeasures. Inappropriate selection of countermeasures result in disastrous consequences for the organization. An impact analysis of all the security candidates is therefore essential in the decision process to select appropriate countermeasures for a given attack.

In this chapter, we propose a novel and systematic process to select individual countermeasures from a pool of candidates, by ranking them based on a trade-off between their efficiency in stopping the attack observed by the SIEM, and their ability to preserve, at the same time, the best service to normal users.

In order to quantitatively analyse the impact of the attack, our model considers two aspects of security policies related to threat responses: firstly, the cumulative long term security policy changes due to previous attacks; and secondly, the fact that security policies may need to be automatically adapted to the current context.

Taking into account previous quantitative models [Jef04, Sch11, CM05, BSB07, SAS06, SBW07a, KLI08, KCBCD10, Khe10], we propose a model that not only evaluates the cost and benefit of a given security solution, but also considers the impact of the attack over a given system, and the organization infrastructure value. Our approach adjusts the metrics introduced by Kheir et al. [KCBCD10, Khe10] to select the countermeasure that provides the highest benefit to the organization.

An overview of the approach is provided in Section 3.1. Section 3.2 presents a sensitivity analysis performed to the RORI index; Section 3.3 describes the process for evaluating and selecting individual countermeasures; and Section 3.4 concludes by analysing the characteristics of the proposed model.

3.1 Overview of the Approach

The Return On Response Investment (RORI) was first introduced by Kheir et al. [KCBCD10, Khe10] as an extension of the Return On Security Investment (ROSI) Index. RORI identifies three cost dimensions for intrusion response i.e. the response collateral damages (CD), the response efficiency (RG = ICb − RC), and the response operational costs (OC). The RORI index formula is depicted in Equation 3.1.

$$RORI = \frac{[ICb - RC] - OC}{CD + OC} \tag{3.1}$$

Where:

- ICb is the expected intrusion impact in the absence of security measures. It measures the costs of the damages due to intrusions or attacks.
- RC is the combined impact for both intrusion and response. RC represents the sum of the expected intrusion impacts after a response is enacted, with the cost that is added by the selected response.
- OC is the operational cost that includes the response set-up and deployment costs such as manpower and over provisioning.
- CD is the response collateral damage which represents the cost that is added by the security measure.

3.1.1 Current RORI Limitations

The deployment of the Return On Response Investment (RORI) index into real world scenarios has presented the following shortcomings:

1. The absolute value of parameters such as ICb and RC is difficult to estimate, whereas a ratio of these parameters is easier to determine, which in turn reduces errors of magnitude.
2. The RORI index is not defined when no countermeasure is selected. Since the operational cost (OC) is associated to the security measure, the RORI index will lead to an indetermination when no solution is enacted (NOOP).
3. The RORI index is not normalized with the size and complexity of the infrastructure.

3.1.2 RORI Improvements

Equation 3.1 proposes to calculate the losses before and after countermeasures (i.e. ICb - RC), which requires to consider the severity and likelihood of a given security incident. However, while evaluating the different possible combinations of countermeasures (detailed in Chapter 4), we realized that it was neither easy nor practical to estimate the values of ICb and RC for a combined solution. That is when the Annual Loss Expectancy (ALE) and Risk Mitigation parameters come into play, making it possible to obtain the residual impact of the incident by multiplying the Annual Loss Expectancy before countermeasure (ALE) with the risk mitigation percentage (RM).

We propose an improvement of the RORI index by taking into account not only the countermeasure cost and its associated risk mitigation, but also the infrastructure value and the expected losses that may occur as a consequence of an intrusion or attack. The improved RORI index handles the choice of applying no countermeasure to compare with the results obtained by the implementation of security solutions (individuals and/or combined countermeasures), and provides a response that is relative to the size of the infrastructure. The improved Return on Response Investment (RORI) index is calculated according to Equation 3.2.

$$RORI = \frac{(ALE \times RM) - ARC}{ARC + AIV} \times 100 \tag{3.2}$$

Where:

- ALE is the Annual Loss Expectancy and refers to the impact cost obtained in the absence of security measures. ALE is expressed in currency per year (e.g., $/year) and will depend directly on the attack's severity and likelihood.

- RM refers to the Risk Mitigation level associated to a particular solution. RM takes values between zero and one hundred percent (i.e. $0\% \leq \text{RM} \leq 100\%$). In the absence of countermeasures, RM equals 0%.
- ARC is the Annual Response Cost that is incurred by implementing a new security action. ARC = OC+CD from Equation 3.1. ARC is always greater than or equal to zero ($ARC \geq 0$), and it is expressed in currency per year (e.g., \$/year).
- AIV is the Annual Infrastructure Value (e.g., Cost of equipment, Services for regular operations, etc.) that is expected from the system, regardless of the implemented countermeasures. ARC is greater than zero ($AIV > 0$), and it is expressed in currency per year (e.g., \$/year).

The improvements of the RORI index are summarized as follows:

- The $ICb - RC$ parameters are substituted by $ALE \times RM$ which can be used more easily to evaluate response goodness of single and combined solutions, while reducing error magnitude.
- The AIV parameter also provides a response relative to the size of the infrastructure. AIV is correlated to the Annual Loss Expectancy (ALE) of the system, and allows to compare the RORI results of different systems regardless of their size.
- The introduction of the Annual Infrastructure Value (AIV) parameter handles the case of evaluating NOOP, which results into a value of zero, meaning that no gain is expected if no solution is implemented.

3.2 Sensitivity Analysis

When analysing the investment in information security, we should not expect an increase in the profits, instead, we should expect a mitigation of the risk to which the organization is exposed. RORI is a relative index that indicates the percentage of benefit perceived if a given countermeasure is implemented. Since the index produces repeatable and consistent results, the model is used to compare security solutions based on relative values.

RORI ranges from $\frac{-ARC}{ARC+AIV}$ (in its lower bound) to $\frac{ALE}{AIV}$ (in its upper bound). A positive RORI means that we expect to diminish the risk up to a certain level and therefore it is beneficial to apply the security solution. For instance, a RORI of 50% means that we expect to mitigate half of the risk to which the organization is exposed. However, when evaluating the NOOP option (no countermeasure is evaluated to react against an attack), we should expect 0% of mitigation.

The worst scenario (the countermeasure cost is higher than the benefits it provides) will have $ALE \times RM << ARC$, therefore $RORI \rightarrow \frac{-ARC}{ARC+AIV}$. The best scenario (perfect mitigation) will have RM=1, ARC=0, therefore $RORI = \frac{ALE}{AIV}$. If the expected benefit is equal to the countermeasure cost, RORI will tend to zero. However, if the expected benefit is lower than the countermeasure cost, RORI will attain a negative value. Only in these cases, where the benefit is higher than the cost of implementing a security measure, RORI will attain a positive value.

Two analyses were performed in order to evaluate the influence of each variable on the selection of security solutions: A single-factor sensitivity analysis and a two-variable sensitivity analysis. The following sections describe both studies.

3.2.1 Single-Factor Sensitivity Analysis

We used the Receiver Operating Characteristic (ROC) curve [Fau06] to perform the single factor sensitivity analysis. In order to identify which variables have the highest impact on the selection of a security solution, we selected the account takeover attack from the Mobile Money Transfer

Service use case (explained in Chapter 5) and we varied one parameter from Equation 3.2, keeping all other variables at their base case values. Four experiments were performed:

Experiment 1: Variation of the AIV. Knowing that for an account takeover attack, ALE = 1200€, we evaluated all the proposed countermeasures, keeping their Annual response Cost (ARC) and Risk Mitigation (RM) values fixed, while changing the Annual Infrastructure Value (AIV) from 0 to 200.000€. Figure 3.1 depicts the results obtained for this evaluation.

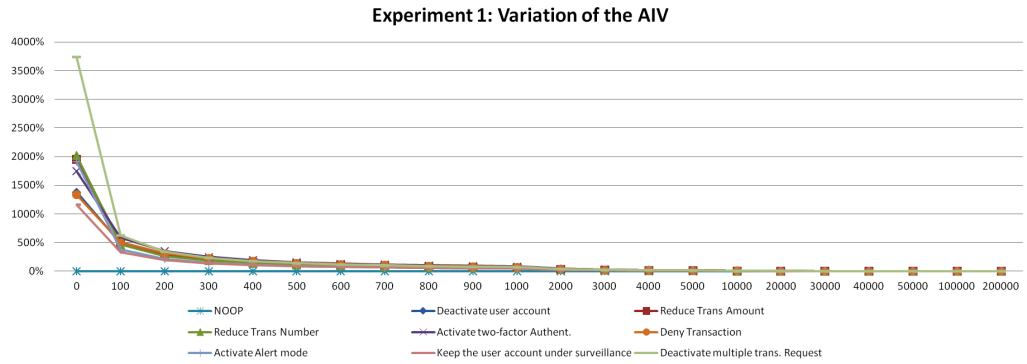

FIGURE 3.1 - RORI index as a function of the AIV

The results show that RORI improves as the Annual Infrastructure Value (AIV) decreases. As expected, AIV demonstrated to have a moderate influence on the variation of the RORI index. The higher the AIV, the lower the RORI index. In some cases, the analysis showed that an AIV inferior to 10% of the ALE may change the ranking of the evaluated countermeasure.

Experiment 2: Variation of the ALE. For the same scenario used in Experiment 1, we evaluated the influence of the ALE over the RORI index by changing ALE from 0 to 200000€, while keeping all other factors unchanged. Figure 3.2 depicts the results obtained for this evaluation.

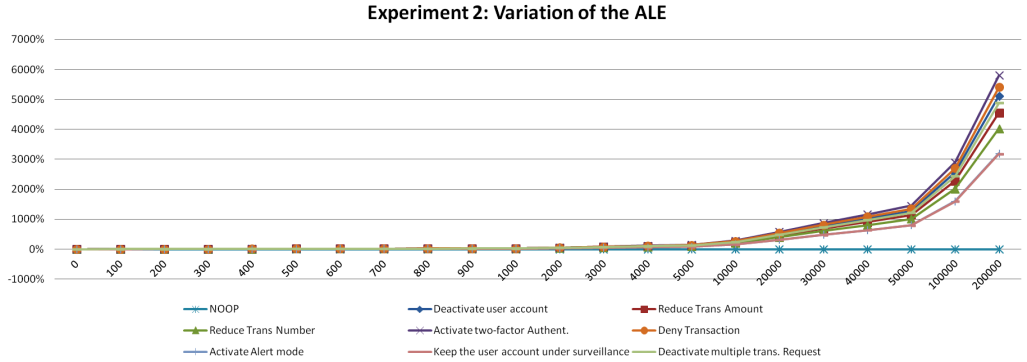

FIGURE 3.2 - RORI index as a function of the ALE

Results demonstrate that the RORI index improves as the ALE increases. This is due to the fact that in the RORI calculation, the ALE variable is multiplied by the Risk Mitigation metric

(RM), which increases the RORI results as the ALE increases. However, ALE demonstrated to have a low influence in the ranking and selection of the security solution. For all of the cases where ALE changed, none of the countermeasures changed their original position on the ranking.

Experiment 3: Variation of the RM. For the same scenario used in Experiments 1 and 2, we evaluated the influence of the Risk Mitigation (RM) factor over the RORI index by changing RM from 0 to 100%, while keeping all other factors unchanged. Figure 3.3 shows the results obtained for this evaluation.

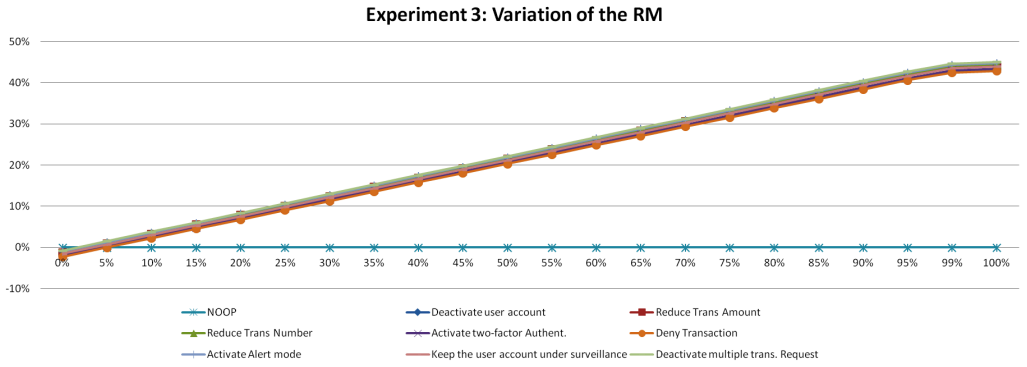

FIGURE 3.3 - RORI index as a function of the RM

The analysis in terms of the Risk Mitigation (RM) variable shows that RM has a linear influence on the selection of the security solution. For all studied attacks, the higher the RM value, the higher the RORI index. It is important to highlight that for a RM inferior to 5%, the RORI index decreases to take values below zero. Moreover, RM below 5% changes the ranking of the countermeasures, which in some cases places in the last position a countermeasure that was ranked first originally. As a result, the RORI index is very sensitive to the changes of the RM variable.

Experiment 4: Variation of the ARC. For the same scenario used in Experiments 1, 2, and 3, we evaluated the influence of the Annual Response Cost (ARC) factor over the RORI index by changing ARC from 0 to 20.000€, while keeping all other factors unchanged. Figure 3.4 shows the results obtained for this evaluation.

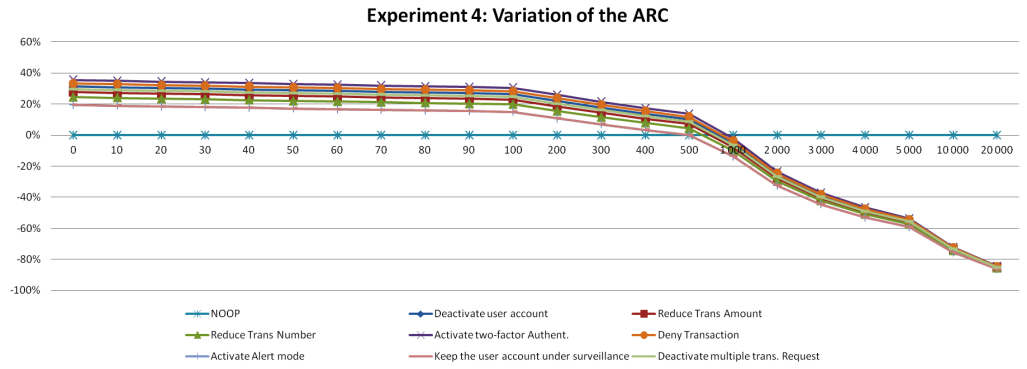

FIGURE 3.4 - RORI index as a function of the ARC

The results demonstrate that the Annual Response Cost (ARC) parameter has a very high influence on the selection of a security solution. The RORI index decreases as the ARC increases. In addition, for those cases where the ARC is too high compared to the benefit it provides, the RORI index decreases to take values below zero; making the solution financially less interesting to be implemented. In such cases, the NOOP alternative appears as a better option.

3.2.2 Two-Variable Sensitivity Analysis

In order to evaluate the effects on the RORI results, we conducted 6 experiments where two variables were changed while the others kept their base case values. Table 3.1 describes the characteristics of each experiment. A summary of the results obtained for each experiment is presented in Table 3.2.

TABLE 3.1 - Characteristics of the Two-Variable Sensitivity Analysis

N	Experiment	Characteristics	Objective
1	ALE vs AIV	For a given countermeasure, the values of RM and ARC are constant while ALE and AIV vary from 0 to 100.000€	Evaluate the influence of ALE and AIV in the selection of countermeasures
2	ALE vs RM	For a given countermeasure ARC remains constant and AIV=2600€while ALE varies from 0 to 100.000€and RM varies from 0 to 100%	Analyse the effects of ALE and RM over the RORI index
3	ALE vs ARC	For a given countermeasure RM remains constant and AIV=2600€while ALE and ARC vary from 0 to 100.000€	Evaluate the influence of ALE and ARC in the selection of countermeasures
4	RM vs ARC	A countermeasure is randomly selected, ALE=1200€, AIV=2600€, RM varies from 0 to 100% and ARC varies from 0 to 100.000€	Analyse the sensibility of the RORI index over a variation of the RM and ARC parameters
5	RM vs AIV	For a given countermeasure, ARC remains constant and ALE=1200€, RM varies from 0 to 100% and AIV varies from 0 to 100.000€	Evaluate the influence of RM and AIV in the selection of countermeasures
6	AIV vs ARC	For a given countermeasure, RM remains constant and ALE=1200€while AIV and ARC vary from 0 to 100.000€	Evaluate the impact of AIV and ARC over the RORI index

The main conclusions reached for this analysis are described as follows:

- If ARC is orders of magnitude below AIV ($ARC \ll AIV$), then the impact of ARC on the RORI is very weak. In this case, $ARC + AIV \cong AIV$, therefore $RORI \cong (\frac{ALE \times RM}{AIV})$. However, if ARC is orders of magnitude above AIV ($ARC \gg AIV$), then the impact of ARC on the RORI index is very strong. In this case, $ARC + AIV \cong ARC$, therefore $RORI \cong \frac{(ALE \times RM) - ARC}{ARC}$.
- If ALE is orders of magnitude below AIV ($ALE \ll AIV$), then ALE negatively impacts the RORI index, since $ALE \times RM \cong 0$, therefore $RORI \cong \frac{-ARC}{ARC+AIV}$. However, if ALE is orders of magnitude above AIV ($ALE \gg AIV$), then the RORI index is positively impacted. In this case, $AIV \cong 0$, therefore $RORI \cong \frac{(ALE \times RM) - ARC}{ARC}$.
- If ALE is orders of magnitude below ARC ($ALE \ll ARC$), then ALE negatively impacts the RORI index, since $ALE \cong 0$, therefore $RORI \cong \frac{-ARC}{ARC+AIV}$. However, if ALE is orders of magnitude above ARC ($ALE \gg ARC$), then the RORI index is positively impacted. In this case, $ARC \cong 0$, therefore $RORI \cong \frac{ALE \times RM}{AIV}$.

- If RM increases compared to the AIV, ALE and ARC values, the RORI index will depend on the magnitude of the ALE metric compared to ARC and AIV. In this case, $ALE \times RM \cong ALE$, therefore $RORI \cong \frac{ALE-ARC}{ARC+AIV}$, making the solution more attractive as the ALE increases.

TABLE 3.2 - Two-Variable Sensitivity Analysis Results

Variable Parameters	Results
ALE vs AIV	The impact of the ALE parameter over RORI increases as the AIV decreases. The higher the AIV, the less attractive the solution. As a result, a benefit ($ALE \times RM$) that is far greater than the infrastructure value (AIV) is always preferable.
ALE vs RM	The impact of ALE over RORI increases as the RM increases. The higher the ALE and RM values, the more attractive the solution. Thus, the ideal solution should provide the highest benefit to the system.
ALE vs ARC	The impact of ALE over RORI increases as the ARC decreases. The lower the annual response cost, the higher the RORI results. Consequently, a countermeasure that is far less expensive than the benefits it provides is preferable.
RM vs ARC	The impact of ARC over the RORI index decreases as the RM increases. A countermeasure that costs more than the benefits it provides should be discarded. Thus, the ideal solution should have the highest risk mitigation value and the lower response cost.
RM vs AIV	The impact of the AIV over RORI decreases as the RM increases. The higher the RM and the lower the AIV, the more attractive the solution. Consequently, the ideal solution should have the highest risk mitigation and the lowest infrastructure value.
AIV vs ARC	The impact of ARC over RORI increases according to its relative significance compared to the AIV parameter. As a result, an alternative that is far less expensive than the infrastructure value is preferable.

3.3 Countermeasure Selection Process

The process for selecting optimal countermeasures is performed in two steps: The RORI Calculation and The Countermeasure Evaluation. The following sections detail each step of the process.

3.3.1 RORI Calculation

The Return On Response Investment metric proposed in Section 3.1.2 is used as a quantitative approach to evaluate and rank a set of countermeasures, which allows to select the one that best mitigates the effects of a given attack. The input parameters for the RORI calculation are of two types: fixed parameters, which depend on the system (AIV), and the intrusion or attack (ALE); and variable parameters, which depend on the countermeasure (RM, ARC).

The calculation of the parameters presented in Equation 3.2 follows the approaches proposed by Kosutic [Kos11] and Lockstep Consulting [Con04] for the ROSI model, as well as the approach proposed by Kheir et al. [Khe10] for the RORI model. The rest of this section details each parameter.

3.3.1.1 Fixed Parameters:

3.3.1.1.1 Annual Loss Expectancy (ALE) ALE refers to the Impact Cost that is produced in the absence of countermeasures. ALE is expressed in currency per year (e.g. $/year) and includes loss of assets (La), Loss of data (Ld), Loss of reputation (Lr), Legal procedures (Lp), Loss of revenues from clients or customers (Lrc), other losses (Ol), Contracted insurance (Ci), and the annual rate of occurrence (ARO), as depicted in Equation 3.3.

$$ALE = (La + Ld + Lr + Lp + Lrc + Ol - Ci) \times ARO \tag{3.3}$$

Where:

- Loss of assets (La), expressed in currency (e.g. $), refers to the value of the physical assets that would be affected by an intrusion or attack (e.g. hardware, furniture, physical infrastructure, and other assets).
- Loss of data (Ld), expressed in currency (e.g. $), refers to the value of the non-physical assets that would suffer a damage or modification as a consequence of an intrusion or attack (software, databases, electronic documents, and other kind of information loss).
- Loss of reputation (Lr), expressed in currency (e.g. $), refers to the loss of image or credibility as a consequence of not providing the products or services at the expected levels.
- Legal procedures (Lp), expressed in currency (e.g. $), refers to the losses due to legal penalties that may arise as a consequence of not accomplishing the contracted obligations with clients.
- Loss of revenues from existing clients (Lrec), expressed in currency (e.g. $), are the losses occurred during the incident as a consequence of not providing the products or services to the clients at the expected levels.
- Loss of revenue from potential clients (Lrpc), expressed in currency (e.g. $), refers to the losses of not acquiring new clients as a consequence of the occurred incident.
- Other losses (Ol), expressed in currency (e.g. $), considers all other losses that may occur during the incident (e.g. recovering time, external services, etc.)
- Contracted insurance (Ci), expressed in currency (e.g. $), if the organization has an insurance that covers part of the losses, this value is subtracted from the total amount of losses.
- Annual Rate of Occurrence (ARO), expressed in times per year, refers to the estimated frequency of the attack.

3.3.1.1.2 Annual Infrastructure Value (AIV) AIV corresponds to the fixed costs that are expected on the system regardless of the implemented countermeasure. AIV is greater than zero ($AIV > 0$), and it is expressed in currency per year (e.g. $/year). AIV includes the following costs: Equipment Costs (Ec), Personnel costs (Pc), Service costs (Sc), Other costs (Oc), and Resell Value (Rv), as shown in Equation 3.4. This is a one time investment and remains constant on the evaluation of all the different countermeasures.

$$AIV = Ec + Pc + Sc + Oc - Rv \tag{3.4}$$

Where:

- Equipment cost (Ec), expressed in currency per year (e.g. $/year), refers to the annual cost of security equipments, products or materials (e.g. purchasing, renting, leasing, licensing, etc) required for the regular operations of the infrastructure.
- Personnel cost (Pc), expressed in currency per year (e.g. $/year), includes the cost of employees required for the regular operations of the infrastructure (e.g. salaries, bonuses, overtime payment, etc).

- Service cost (Sc), expressed in currency per year (e.g. $/year), includes the costs of regular services (e.g. electricity, infrastructure costs, contracted insurance, etc), related to the business unit for its regular operations.
- Other costs (Oc), expressed in currency per year, (e.g. $/year), refers to all other costs required for the regular operation of the system's infrastructure.
- Resell costs (Rv), expressed in currency per year (e.g. $/year), refers to the value of the security equipments after their usage.

3.3.1.2 Variable Parameters:

3.3.1.2.1 Risk Mitigation (RM) RM refers to the risk mitigation associated with a given countermeasure. RM is calculated as the countermeasure Surface Coverage (SC) times its Effectiveness Factor (EF), as depicted in Equation 3.5. RM takes values between zero and one hundred percent ($0\% \leq \text{RM} \leq 100\%$). In the absence of countermeasures, RM equals 0%. The calculation of a combined RM is detailed in Chapter 4.

$$RM = SC \times EF \tag{3.5}$$

Where:

- Surface Coverage (SC), is the percentage of the attack surface that is covered and controlled by a given countermeasure.
- Effectiveness Factor (EF), considers the percentage of reduction of the total incident cost that is given from the enforcement of a security measure.

3.3.1.2.2 Annual Response Cost (ARC) ARC refers to the costs associated to a given countermeasure. ARC is always greater than or equal to zero ($ARC \geq 0$), and it is expressed in currency per year (e.g., $/year). It includes direct costs: e.g., Cost of implementation (CoI), Cost of maintenance (CoM), Other direct costs (Odc); and indirect costs (e.g., consequences that may originate the adoption of a particular countermeasure to a legitimate user), as shown in Equation 3.6.

$$ARC = CoI + CoM + Odc + Ic \tag{3.6}$$

Where:

- Cost of implementation (Ci), expressed in currency per year (e.g. $/year), refers the cost of deployment, installation and/or implementation of security measures to mitigate a given attack.
- Cost of maintenance (Cm), expressed in currency per year (e.g. $/year), includes the cost of regular services (e.g. electricity, consulting, analysis, testing, etc.) that are needed for normal operations of the implemented countermeasures.
- Other direct costs (Odc), expressed in currency per year (e.g. $/year), refers to all other direct costs (e.g. suppliers and partners) that are needed to put in place countermeasures.
- Indirect costs (Ic), expressed in currency per year (e.g. $/year), include all other costs, such as consequences that may originate the adoption of a particular countermeasure to legitimate users.

ALE and AIV parameters are defined statically and depend on the system, as well as the intrusion or attack detected. RM and ARC parameters depend on the proposed security measures.

Their calculation requires the system to determine the countermeasure surface coverage, which includes the probability of the union of events.

3.3.2 Quantification of the RORI parameters

The quantification of the parameters composing the RORI model proposed in Equation 3.2 is a task that requires expert knowledge, statistical data, simulation and risk assessment tools. Our experience in quantifying impact losses, as well as countermeasure costs and benefits for different security systems demonstrate that within 3 to 4 hours of discussions with use case providers and simple simulation runs, we are able to estimate each parameter composing the RORI model. In addition, the quantification of the RORI parameters only requires accuracy in their relative values (not in their absolute values). If all the parameters are estimated by a standard methodology, the RORI evaluation should produce repeatable and consistent results. The remaining of this section proposes a simple and well structure methodology to help security analysts in the estimation of such parameters.

3.3.2.1 Quantification of the Annual Loss Expectancy (ALE)

For the estimation of the ALE, we adopted the approach proposed by Lockstep [Con04] to use the scale values of severity, which convert qualitative estimations into quantitative values of costs. For instance, a 'minor' loss of asset (La) represents a cost of $1,000; whereas a 'serious' loss of asset (La) represents a cost of $1,000,000. The estimation of all other losses (i.e. Ld, Lr, Lp, Lrec, Lrpc, Ol) follows the same approach.

The likelihood of an incident is transformed into a frequency value, which results into the Annual Rate of Occurrence (ARO) parameter. For instance, a 'low likelihood' means that the incident is likely to occur once every year, and represents therefore, a value of ARO = 1.0; whereas, a 'high likelihood' means that the incident is likely to occur once per month or less, and represents a value of ARO = 12.0.

Both parameters (i.e. severity and likelihood) are estimated using a survey and scoring system, which combine expert knowledge and statistical data to quantify risk exposure. In order to handle uncertainty, we use the Monte Carlo simulation approach as proposed by ROI and ROSI implementers [?, ?, Con04, SAS06]. The approach represents the solution of the problem as a parameter of a hypothetical population, and uses a random sequence of numbers to construct a sample of the population, from which statistical estimates of the parameter are obtained [Hal].

To run our simulation, we chose triangular distributions [EHP00] to evaluate the most likely values assigned to each level of security and likelihood, with minimum and maximum possible values of each level (Tables 3.3 and 3.4). This type of statistical computations can be easily achieved using basic statistical software or spreadsheet editors (e.g. Quadrant[1], XLSim[2]).

After 250 iterations, we are able to obtain a value of the losses and frequency that compose the ALE parameter, which represents the expected annual loss as a consequence of the realization of a given threat.

[1] Quadrant: The Quick and dirty risk analysis tool, available at: www.qdrnt.com/home.htm
[2] Monte Carlo simulation for excel featuring distribution strings, available at: http://xlsim.com/xlsim/index.html

TABLE 3.3 - Severity transformed to probabilistic Costs

Severity	Min Cost($)	Most Likely Cost ($)	Max Cost ($)
Insignificant	0	0	0
Minor	0	1,000	2,000
Significant	5,000	10,000	20,000
Damaging	50,000	100,000	200,000
Serious	500,000	1,000,000	2,000,000
Grave	5,000,000	10,000,000	20,000,000

TABLE 3.4 - Likelihood transformed to Probabilistic ARO

Likelihood	Min ARO	Most Likely ARO	Max ARO
Negligible	0.0	0.10	0.05
Very Low	0.1	0.8	0.5
Low	0.8	1.5	1.0
Medium	1.5	2.5	2.0
High	2.5	20.0	12.0
Very High	20.0	120.0	50.0
Extreme	120.0	1,000.0	500.0

3.3.2.2 Quantification of the Annual Infrastructure Value (AIV)

This parameter is calculated as the sum of the annual value of all the equipments - i.e. Policy Enforcement Points (PEP) - that are needed to be deployed in the preliminary phase of the system architecture in order to guarantee a desired level of security. The AIV includes the cost of purchasing, licensing, and/or leasing the security equipments in a given organization.

It is important, however, to answer the following questions while estimating the AIV parameter:

- What kind of PEPs and which quantity is required for the system security (e.g. Firewalls, IPS, IDS, SIEM, etc.)?
- What is the lifetime expectancy of the PEP?
- What is the annual cost of purchase, licensing or leasing of the PEP?
- How many employee-hours are required for the operation of the PEP?
- How many hours/year is the PEP expected to be active?
- Is there an insurance contracted for the PEP? If so, how much does it cost per year?
- How many times per year does the PEP need to be checked or maintained?
- Is there any other cost associated with the operation of the PEP in the security infrastructure?
- What is the resell value of the PEP?

If for instance, a PEP that is purchased at $10,000 and has an estimated lifetime of 5 years, will have an Equipment cost (Ec = 2,000$/year). The same PEP that requires 36 employee-hours a year to operate appropriately, and knowing that the cost of each employee-hour is equivalent to $10, will have a Personnel cost (Pc = 360$/year). Similarly, the same PEP that needs an insurance of $250 per year and which is active 24/7 consuming in average $250 per year on operational costs, will have a Service cost (Sc = 500$/year). In addition, the PEP needs to be maintained twice per year at a cost of $200, will have Other costs (Oc = 400$/year). And finally, the PEP resell value

is estimated to be (Rv = \$500/5 years = 100\$/year). As a result, the annual cost of the selected PEP is equivalent to 3,160\$/year. If the security infrastructure is composed of several PEPs, the same procedure is applied to estimate the cost of each PEP. The resulting Annual Infrastructure Value (AIV) represents the sum of all the PEP's costs.

3.3.2.3 Quantification of the Risk Mitigation (RM)

In order to calculate the risk mitigation of a given countermeasure we need to first measure the portion of the attack the countermeasure covers and then, the level of effectiveness of such solution in mitigating the attack.

The countermeasure surface coverage is a value inherent to the type of attack and the system that it affects, making it possible to obtain different surface coverage values for the same countermeasure applied in different attack scenarios. For instance, blocking a suspected user might cover 85% of the attack surface, while increasing the surveillance to punctually block operations covers only 60% of the attack surface. To obtain these figures, it is necessary to perform a risk assessment following any of the international standards (e.g. NIST [oST70], ISO [Org08]), or any of the risk management methodologies (e.g. MEHARI [Clu10], EBIOS [ANS10], CRAMM [Ent05]), which allows the identification of assets, threats, vulnerabilities, likelihood, controls, and consequences.

Organizations such as Microsoft [Mic12] have released tools to assist security administrators in the identification and analysis of the attack surface. The percentage of assets and threats that are controlled by a given countermeasure represents the surface coverage.

In order to quantify the Effectiveness Factor (EF), we follow the Norman's methodology [Nor10], which defines the risk of an attack (R) as the product of its vulnerability (V), likelihood or probability (P), and severity or consequence (C), (i.e., $R = V \times P \times C$). A risk can be mitigated by decreasing the vulnerability, probability, and/or consequence. Each factor is estimated in a range from 1 to 10, where 1 represents the least likely value, and 10 represents the most likely value for a given parameter. Therefore, the Effectiveness Factor (EF) is calculated as the risk reduction percentage that results from the application of a given countermeasure. For instance, if the risk of a given attack before countermeasure is $R_1 = 10 \times 7 \times 7 = 490$ and the resulting risk after the application of a particular countermeasure is $R_2 = 7 \times 6 \times 6 = 252$, then $EF = 100 - (\frac{R_2 \times 100}{R_1}) = 51.43\%$

3.3.2.4 Quantification of the Annual Response Cost (ARC)

Starting from the point that the cost of a countermeasure is not just its purchase price, we identify direct and indirect costs to be considered while estimating the Annual Response Cost (ARC) parameter. Based on educated expert knowledge and statistical data it is possible to determine each element composing the ARC.

In contrast to the AIV parameter, the ARC is a variable cost associated with the implementation of a given countermeasure. For instance, let us suppose that the user authentication information of a Web service is stored in a database. Whenever users want to access the system, they need to provide their corresponding login and password. However, for suspected users, the organization wants to implement a countermeasure that asks for a two-factor authentication (e.g. a challenge question, a security pin, etc.). The implementation of this countermeasure requires the organization to expend additional employee-hours which in turn represents a given cost. This latter is defined as the cost of implementation (Ci).

In addition, the countermeasure is going to be active only for suspected users for a given period of time, which means that the system will turn the countermeasure from 'on' to 'off' according to the security tests and analysis performed. These tests and analysis represent the cost of maintenance

(Cm) to the organization.

The activation/deactivation of a given countermeasure engenders other direct and indirect costs. For instance, requesting an additional authentication method to legitimate users may cause these users to unsubscribe from the service and search for another one. This collateral damage represents an indirect cost (Ic) to the organization. Collateral damages can be quantified as the variation between the current and the projected productivity that an organization experiences due to a side effect of a given solution [SAS06].

3.3.3 Countermeasure Evaluation

We initialize the process of countermeasure evaluation with a default RORI value equal to zero (RM and ARC parameters equal 0, since no countermeasure is implemented). The countermeasure evaluation process starts by selecting the first countermeasure on the list and calculating its RORI index (Step 1). The obtained RORI is compared with the one by default (Step 2), as depicted in Figure 3.5.

If the resulting RORI is different to the one by default (Step 3), the system checks if the current RORI is greater to the default value, in such a case, the countermeasure becomes the selected one, and it overrides the default RORI value (Step 3a). However, if the resulting RORI value is lower than the one by default, the system checks for another countermeasure to evaluate and the default RORI remains unchanged (Step 3b).

If the resulting RORI is equal to the default value (Step 4), the system checks for the annual response cost (ARC) and selects the one with the lowest cost value (Step 4a) since it is always preferred to implement a security solution that costs the least and provides the highest benefit. In case the ARC of the evaluated solution is higher than the ARC of the default countermeasure, the system checks for another countermeasure to evaluate and the default RORI remains unchanged (Step 4b). It may happen that, when comparing the costs of two countermeasures, they are exactly the same. In such a case, the system keeps the current RORI value as the default one and checks for another solution to evaluate.

The process is repeated to evaluate the second countermeasure on the list, then the third, and so on, until no countermeasure is left (Step 5a). The system selects the last countermeasure taken as default, since it is the one that provides the highest RORI index (Step 5b).

3.3.4 Remaining Limitations of the RORI-based Countermeasure Selection

The evaluation and selection of countermeasures depends on the appropriate estimation of the infrastructure value, attack impact and the definition of the security policies needed to mitigate the attack. Such definition should include the costs and benefits associated to a particular security policy in a given attack scenario.

The main limitation of the RORI-based model is the accuracy in the estimation of the different parameters that compose the formula. Estimating the Annual Loss Expectancy of an attack to occur on a given system and the Risk Mitigation level of a particular countermeasure is difficult and requires a considerable effort. An objective estimation of these two factors is rather infeasible, since it requires predictions of an event that has not yet occurred. However, a qualitative estimation of RM and ALE remains possible, making the RORI evaluation an interesting tool to help security analysts in the decision making process.

In addition, the RORI model presented in this chapter does not consider interdependence among

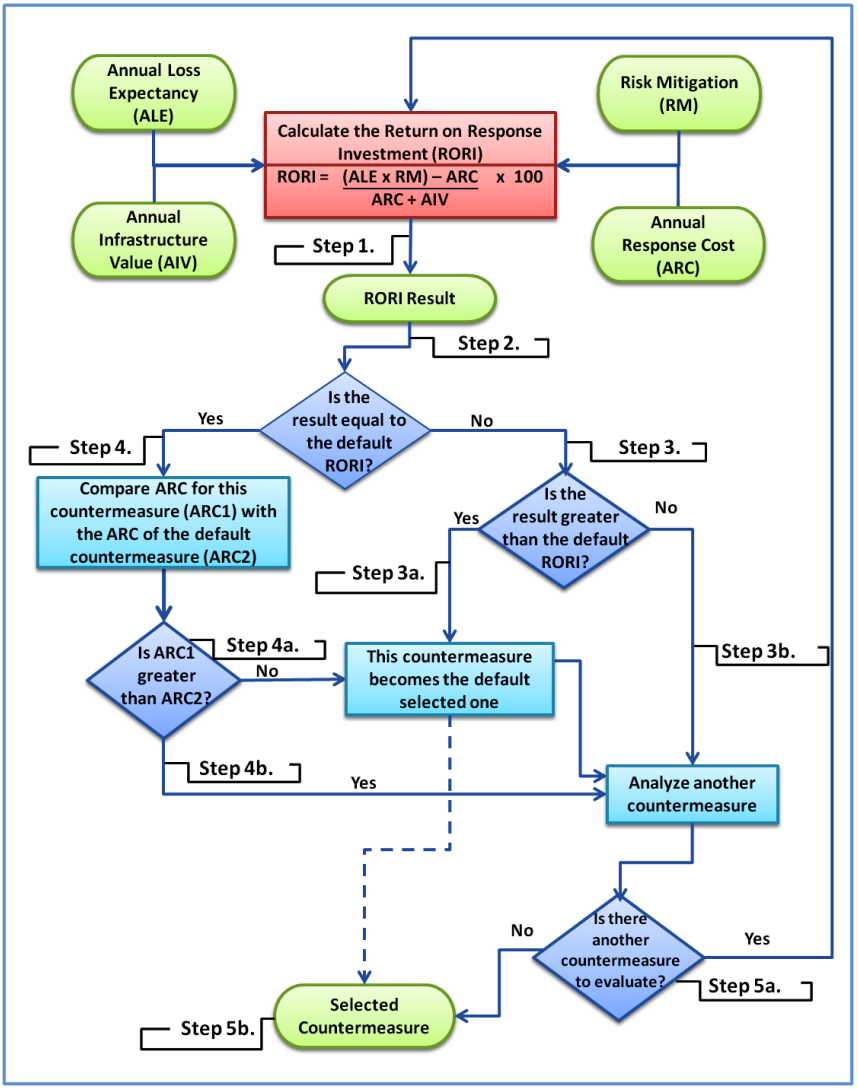

FIGURE 3.5 - Countermeasure Evaluation Flowchart

countermeasures (i.e., how the application of a countermeasure affects the effectiveness of others), nor it discusses the restrictions and/or conflicts that may originate with the implementation of the selected countermeasure (e.g., partially or totally restrictive countermeasures).

Finally, the model limits the action to only one countermeasure over a given attack, and does not discuss neither the effects that one or multiple countermeasures may have on several risks, nor the effects of applying multiple countermeasures at a time.

3.4 Conclusions

In this chapter we introduced a quantitative approach to select optimal security countermeasures based on the Return On Response Investment (RORI) index, making it possible to evaluate response collateral damages and response effects on intrusions.

Our solution is split into two steps: the calculation of the Return on Response Investment (RORI) index, which evaluates the expected losses that result for a particular attack versus the benefits that can be obtained if a countermeasure is implemented; and the process of selection and ranking of individual countermeasures. Within the process, the countermeasure with the highest RORI index is selected as the one that provides the highest benefit to the organization. The RORI index takes into account not only the cost and the risk mitigation value associated to a particular solution, but also the losses and operational cost of the infrastructure.

As a result, we are able to evaluate, rank and select the countermeasure that provides the highest benefit to the organization (the highest RORI index). The next chapter addresses some limitations from Section 3.3.4 by studying the effect of combining two or more security solutions for a single attack scenario, and proposes a methodology to evaluate and select optimal group of countermeasures.

Chapter 4

Combined Countermeasure Selection

So their combinations with themselves and with each other give rise to endless complexities, which anyone who is to give a likely account of reality must survey.

PLATO, The Timaeus

-

Contents

Countermeasures have been defined in Chapter 1 as security actions required to oppose an attack by eliminating or preventing it, by minimizing the harm it can cause, or by discovering and reporting it so that corrective actions can be taken [Kis11]. A combined countermeasure results from the implementation of two or more security measures applied simultaneously to mitigate a given attack. A combined countermeasure is therefore analysed as a single security measure with a

combined cost and a combined effectiveness. Determining the combined cost and effectiveness for multiple countermeasures requires a considerable effort in the estimation of the surface coverage for each security candidate and their overlapped area.

Most solutions suggest the deployment of multiple countermeasures as a single treatment to mitigate the effects of modern attacks [NLSO11, HDGKJ06, HTRM10, DCH06]. However, the methodology to evaluate and select combined countermeasures is either hardly explained or very complicated to implement.

This chapter therefore proposes a simple and well-structured model to select the optimal combination of countermeasures based on the Return on Response Investment (RORI) index. This latter compares the expected impact of the attacks when no action is enacted (NOOP), against the expected impact after applying countermeasures. Our solution is built upon the process defined in Chapter 3, and performs a systematic analysis of all possible combinations of countermeasures to select the one that returns the highest RORI value.

The rest of the chapter is structured as follows. Section 4.1 briefly presents the limitations of current approaches regarding combined countermeasures. Section 4.2 introduces the notion of probability, and the different combination approaches to be used in the proposed model. Section 4.3 presents the combinatorial axioms, and the methodology to calculate the surface coverage of multiple countermeasures. The process for selecting optimal combination of countermeasures is explained in Section 4.4. Finally, Section 4.5 concludes regarding the combination approach.

4.1 Limitations of Current Solutions

None of the existing cost-sensitive models has been designed to evaluate the implementation of combined countermeasures against a given attack. Although the literature in the domain suggests that the implementation of multiple countermeasures provides more interesting results in the mitigation of attacks, most metrics only consider the evaluation of a single countermeasure for a single attack scenario.

For instance, the RORI index proposed by Kheir et al. [KCBCD10] presents limitations in the estimation of some parameters of the formula. Estimating the impact after countermeasure is a difficult task that may result in inaccurate values and requires in most of the cases expert knowledge and/or statistical data. In Chapter 3 we propose an improvement of the model by substituting the aforementioned parameter for the risk mitigation (RM) value associated to each security solution. This improves accuracy on the RORI calculation and provides a model that does not need to be fed by historical or statistical data and that can be used to evaluate single and combined countermeasures.

Nakatsu et al. [NLSO11] and Harwood et al. [HTRM10] already propose an approach to combine multiple countermeasures, but it is neither practical nor easy to perform due to the complexity in calculating the different parameters. In addition, the selection of countermeasures is performed manually. In comparison, our approach is relatively simple and does not require the intervention of a security analyst to evaluate and rank all possible security measures.

4.2 Combinatorial Parameters

Implementing multiple countermeasures simultaneously requires the calculation of their union and intersection area. Since the surface coverage of multiple countermeasures generally overlap, the use of several parameters (i.e. probability of events, combinatorial models and approaches), is

important to estimate the exact mitigation value of a combined solution. This section details each parameter and provides the means to calculate this mitigation value.

4.2.1 Event Probability

The probability of an event is a numerical value (between 0 and 1) that describes the likelihood or relative frequency of the event to occur [GS97, RH98] . In the occurrence of multiple events, two cases are distinguished: mutually exclusive and non-mutually exclusive events.

4.2.1.1 Mutually Exclusive Events

For mutually exclusive events (no outcomes in common), the probability that at least one of the events occurs is equal to the sum of their individual probabilities [Olo05, Wan04], as shown in Equation 4.1

$$\bigcup_{i=1}^{n} P(E_i) = \sum_{i=1}^{n} P(E_i) \tag{4.1}$$

In a more explicit way, we have in Equation 4.2 the probability of the union of “n" mutually exclusive events.

$$P(E_1 \cup E_2 \cup ... \cup E_n) = P(E_1) + P(E_2) + ... + P(E_n) \tag{4.2}$$

However, the probability that all the mutually exclusive events occur at the same time is equal to zero, since the the occurrence of any one of them automatically implies the non-occurrence of the remaining n-1 events. Equation 4.3 shows this relation.

$$P(E_1 \cap E_2 \cap ... \cap E_n) = 0 \tag{4.3}$$

4.2.1.2 Non-Mutually Exclusive Events

For non-mutually exclusive events, the probability that at least one event occurs is expressed as the inequality depicted in Equation 4.4.

$$\bigcup_{i=1}^{n} P(E_i) \leq \sum_{i=1}^{n} P(E_i) \tag{4.4}$$

In a more explicit way, we have in Equation 4.5 the probability of the union of “n" non-mutually exclusive events.

$$\begin{aligned} P(E_1 \cup E_2 \cup ... \cup E_n) \quad &\leq P(E_1) + P(E_2) + ... + P(E_n) - \textstyle\sum_{1 \leq i < j \leq n} P(E_i \cap E_j) + \\ &\textstyle\sum_{1 \leq i < j < k \leq n} P(E_i \cap E_j \cap E_k) + ... + (-1)^{n+1} \prod_{i=1}^{n} P(E_i) \end{aligned} \tag{4.5}$$

However, the probability that all non-mutually exclusive events occur at the same time is equal to the product of their individual probabilities [Olo05, Wan04], as shown in Equation 4.6

$$\bigcap_{i=1}^{n} P(E_i) = \prod_{i=1}^{n} P(E_i) \tag{4.6}$$

In a more explicit way, we have in Equation 4.7 the probability of the intersection of “n" mutually exclusive events.

$$P(E_1 \cap E_2 \cap ... \cap E_n) = P(E_1) \times P(E_2) \times ... \times P(E_n) \tag{4.7}$$

4.2.2 Combination Approaches

To determine the maximal number of combined solutions that can result from a set of individual countermeasures, we define a set of elements where the order is not important and repetitions are not allowed. This section describes the two approaches (e.g., restrictive and non-restrictive), that are considered in the combination of security countermeasures based on the aforementioned set of elements.

4.2.2.1 Non-restrictive Combination Approach

Non-restrictive countermeasures are those that can be perfectly combined (e.g., no restrictions exist in their combination). The maximum number of combined countermeasures is calculated as the sum of the nth row (counting from zero) of the binomial coefficients [Gri85, Ros94], as expressed in Equation 4.8, where ’n’ is the total number of elements to be combined and ’k’ is the set of combined elements (from zero to ’n’).

$$\sum_{0 \leq k \leq n} \binom{n}{k} = 2^n - 1 \tag{4.8}$$

For instance, for a group of 4 countermeasures, the maximum number of combined solutions is $\binom{4}{k} = 2^4 - 1 = 15$, as shown in Table 4.1.

4.2.2.2 Restrictive Combination Approach

Three cases may appear when one or more countermeasures are restrictive and cannot be combined with other countermeasures.

4.2.2.2.1 Mutually Exclusive Countermeasures In probability, two events are mutually exclusive if they cannot occur simultaneously (i.e., they have no outcomes in common) [RH98, Olo05]. Considering the case of having four countermeasures (C1, C2, C3, C4), and knowing that C1 and C3 are mutually exclusive (they cannot be combined), the total number of possible combinations is given in Equation 4.9 [And, ML].

$$\sum_{0 \leq k \leq n} \binom{n}{k} = \left[(k+1)(2^{n-k})\right] - 1 \qquad for \;\; k \geq 2 \tag{4.9}$$

Where ’n’ is the total number of elements to be combined and ’k’ the number of mutually exclusive countermeasures. In this case, we will have n=4 and k=2, therefore $\binom{4}{2} = \left[(2+1)(2^{4-2})\right] - 1 =$ 11 possible combinations, as shown in Table 4.1.

4.2.2.2.2 Partially Restrictive Countermeasures One countermeasure can be implemented along with some other countermeasures but not with all of them. For instance, let us assume that from the set of 4 countermeasures (C1, C2, C3, C4), C1 can only be combined with C4 (the combinations of C1 with C2 and C1 with C3 creates a conflict on the system). The total number of possible combinations is given in Equation 4.10 [FS09b].

$$\sum_{0 \leq k \leq n} \binom{n}{k} = \left[(2^{k-1}+1)(2^{n-k})\right] - 1 \qquad for \;\; k \geq 2 \tag{4.10}$$

Where 'n' is the total number of elements to be combined and 'k' the number of partially restrictive countermeasures. In this case, we will have n=4 and k=3, therefore $\binom{4}{3} = \left[(2^{3-1}+1)(2^{4-3})\right] - 1 = 9$ possible combinations can be generated, as shown in Table 4.1.

4.2.2.2.3 Totally Restrictive Countermeasures From the group of selected countermeasures, one or more of them cannot be combined with any other countermeasure. In order to know the exact number of combinations, we apply Equation 4.11 [Ros94, War07].

$$\sum_{0 \leq k \leq n} \binom{n}{k} = 2^{n-k} + (k-1) \qquad for \;\; k \geq 1 \tag{4.11}$$

Where 'n' is the number of elements to be combined and 'k' the number of totally restrictive countermeasures. For instance, having 4 countermeasures (C1, C2, C3, C4) and knowing that C1 is totally restrictive, we will have $\binom{4}{1} = 2^{4-1} + (1-1) = 8$ possible combinations as shown in Table 4.1.

TABLE 4.1 - Combinations generated from a group of 4 countermeasures

Type	**Non-Restrictive**	**Mutually Exclusive**	**Partially Restrictive**	**Totally Restrictive**
Single	C1, C2, C3, C4	C1, C2, C3, C4	C1, C2, C3, C4	C1, C2, C3, C4
Double	C1+C2, C1+C3, C1+C4, C2+C3, C2+C4, C3+C4,	C1+C2, C1+C4, C2+C3, C2+C4, C3+C4,	C1+C4, C2+C3, C2+C4, C3+C4,	C2+C3, C2+C4, C3+C4,
Triple	C1+C2+C3, C1+C2+C4, C1+C3+C4, C2+C3+C4,	C1+C2+C4, C2+C3+C4	C2+C3+C4	C2+C3+C4
Quadruple	C1+C2+C3+C4			
Total	**15**	**11**	**9**	**8**

4.3 Proposed Methodology

When two or more countermeasures are implemented to mitigate a given attack, their combination affects the estimation of the Risk Mitigation (RM) and the Annual Response Cost (ARC). These two parameters are required to calculate the improved RORI index proposed in Section 3.1.2. In addition, RM requires the estimation of the countermeasure surface coverage. This section

proposes two combinatorial axioms (one for the calculation of the combined RM, and another for the calculation of the combined ARC) and a methodology to estimate the surface coverage for joint and disjoint countermeasures, as an approach to evaluate multiple countermeasures.

4.3.1 Combinatorial Axioms

The following axioms have been defined in order to calculate the cost and risk mitigation level that results from the combination of two or more countermeasures:

Axiom 1 The cost of a combined countermeasure is equal to the sum of all individual countermeasure's cost, e.g., $ARC(C_1 \cup C_2) = ARC(C_1) + ARC(C_2)$, where ARC is the countermeasure cost and C_1, C_2 are the individual countermeasures. Equation 4.12 shows a more general case.

$$ARC(C_1 \cup ... \cup C_n) = ARC(C_1) + ... + ARC(C_n) \tag{4.12}$$

This axiom is used for both, joint and disjoint countermeasures, since the enforcement of multiple countermeasures generally implies higher costs of implementation, consulting, maintaining, etc, compared to the costs incurred out of the enforcement of a single solution. In addition, with a pessimistic approach, we are able to estimate the return on response investment for a combined solution in the worst of the cases, meaning that if the solution is effective for a higher cost, it will still be effective if the cost is lesser than the one estimated.

Axiom 2 The Risk Mitigation (RM) for a combined solution is calculated as the probability of their combined surface coverage $SC(C_1 \cup C_2 \cup ... \cup C_n)$ times the effectiveness factor (EF). For two mutually exclusive countermeasures (C_1, C_2), the risk mitigation is calculated as $RM(C_1 \cup C_2) = SC(C_1) \times EF(C_1) + SC(C_2) \times EF(C_2)$. However, for two non-mutually exclusive countermeasures, the risk mitigation can be calculated in two ways:

1. As the sum of their individual surface coverage times the Effectiveness Factor minus the surface coverage of their intersection times the minimum effectiveness factor, e.g., $RM(C_1 \cup C_2) = SC(C_1) \times EF(C_1) + SC(C_2) \times EF(C_2) - SC(C_1 \cap C_2) \times min(EF(C_1), EF(C_2))$. Equation 4.13 shows a more general case.

$$\begin{aligned} RM(C_1 \cup ... \cup C_n) &= SC(C_1) \times EF(C_1) + ... + SC(C_n) \times EF(C_n) \\ &- \textstyle\sum_{1 \le i < j \le n} (SC(C_i \cap C_j) \times min(EF(C_i), EF(C_j))) \\ &+ \textstyle\sum_{1 \le i < j < k \le n} (SC(C_i \cap C_j \cap C_k) \times min(EF(C_i), EF(C_j), EF(C_k))) \\ &+ ... + (-1)^{n+1} \textstyle\prod_{i=1}^{n} SC(C_i) \times min(EF(C_1), ..., EF(c_n)) \end{aligned} \tag{4.13}$$

2. As the difference between the two partially covered countermeasure surfaces times their effectiveness factor plus the surface coverage of their intersection times the maximum effectiveness factor, e.g., $RM(C1 \cup C2) = [SC(C_1) - (SC(C_1) \times SC(C_2))] \times EF(C_1) + [SC(C_2) - (SC(C_1) \times SC(C_2))] \times EF(C_2) + SC(C_1 \cap C_2) \times max(EF(C_1), EF(C_2))$.

4.3.2 Countermeasure Surface Coverage

In Chapter 6 we will introduce the attack volume to determine the space of the infrastructure that is exposed to multiple attacks, as well as the portion of such infrastructure that is covered by each

attack and countermeasure. In this section, we study a single attack scenario, therefore we propose an approximation of the countermeasure coverage based on the attack surface notion.

In information security technology, the concept of surface coverage is related to the attack surface, defined by Manadhata [MW10] as the subset of the system's resources that a malicious entity may use to send/receive data into/from the system to attack the system. Thus, the more exposed the system's surface, the more attack opportunities and hence, the more likely the system will be the target of an attack [HW07].

Intuitively, a countermeasure surface coverage represents the level of action that a security solution may have on a system's attack surface. In other words, the surface coverage is the percentage of the attack surface that is covered and controlled by a given countermeasure. The surface coverage is required to calculate the risk mitigation level (RM) for a single and combined countermeasure.

The union of two or more surfaces is an index that ranges from the maximum surface coverage of the group of countermeasures (e.g., max $SC(C_1, ..., C_n)$) in its lower bound, to the sum of the individual surfaces (e.g., $\sum SC(C_1), ..., SC(C_n)$) in its upper bound. The intersection of two or more surfaces is an index that ranges from zero in its lower bound, to the minimum surface coverage of the group of countermeasures in its upper bound (e.g., min $SC(C_1, ..., C_n)$). Two cases can be distinguished in the calculation of a surface coverage for a combined countermeasure (e.g., joint and disjoint surfaces). Figure 4.1 depicts these cases.

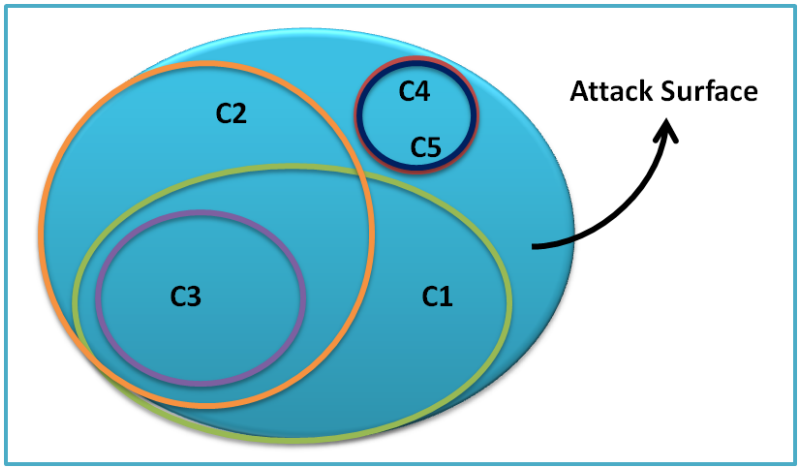

FIGURE 4.1 - Countermeasure Surface Coverage

- **Disjoint Surfaces:** The surface of one countermeasure is disjoint from the surface coverage of another countermeasure if they have no elements in common. Therefore, having two disjoint countermeasures (C_1, C_4), the surface coverage of the union is calculated as $SC(C_1 \cup C_4) = SC(C_1) + SC(C_4)$, and the surface coverage of the intersection is equal to zero.

- **Joint Surfaces:** The surface of one countermeasure is partially or totally covered by another countermeasure. For partially covered countermeasures (e.g., C_1, C_2), the union of the combined solution is calculated as the sum of the individual surfaces minus their intersection (Equation 4.14), and the surface coverage of the intersection is calculated as the average of the lower and upper bounds (Equation 4.15).

$$SC(C_1 \cup C_2) = SC(C_1) + SC(C_2) - SC(C_1 \cap C_2) \tag{4.14}$$

$$SC(C_1 \cap C_2) = \frac{SC(C_1 \cap C_2)_{LOW} + SC(C_1 \cap C_2)_{UP}}{2} \tag{4.15}$$

It is important to highlight that in the practice, an accurate value of the intersection of two or more countermeasures is difficult to estimate. This is a time consuming process that requires the technical definition of each countermeasure and a great expertise in the domain. However, we propose an approximation of this value that is computed as the average between the minimum (LOW) and the maximum (UP) possible values that may result from the intersection of multiple countermeasures.
The lower bound of the intersection between two countermeasures $SC(C_1 \cap C_2)_{LOW}$ is equal to zero if the sum of their surfaces is lesser or equal to one; and it gets the value of $(SC(C_1) + SC(C_2) - 1)$ otherwise.
The upper bound of the intersection between two countermeasures is equal to the total coverage of the smallest surface.
For a general case, the union and intersection of combined solutions is calculated using Equation 4.16, Where $x = SC(C_1) + ... + SC(C_n) - (n-1)$, and 'n' is the maximal number of countermeasures to combine.

$$\begin{aligned}
SC(C_1 \cup ... \cup C_n) = & \quad \textstyle\sum_{i=1}^{n} SC(C_i) - \sum_{1 \le i < j \le n} SC(C_i \cap C_j) + \\
& \textstyle\sum_{1 \le i < j < k \le n} SC(C_i \cap C_j \cap C_k) + ... + (-1)^{n+1} \prod_{i=1}^{n} SC(C_i) \\
SC(C_1 \cap ... \cap C_n) \cong & \quad \tfrac{SC(C_1 \cap ... \cap C_n)_{LOW} + SC(C_1 \cap ... \cap C_n)_{UP}}{2} \\
SC(C_1 \cap C_n)_{LOW} = & \quad \begin{cases} 0 & if\ SC(C_1) + ... + SC(C_n) \le n-1 \\ x & if\ SC(C_1) + ... + SC(C_n) > n-1 \end{cases} \\
SC(C_1 \cap ... \cap C_n)_{UP} = & \quad min\ \{SC(C_1), ..., SC(C_n)\}
\end{aligned} \tag{4.16}$$

For totally covered surfaces (e.g., C_1, C_3), the union of the combined surface is calculated as $SC(C_1 \cup C_3) = SC(C_1)$, and the surface coverage of the intersection is calculated as $SC(C_1 \cap C_3) = SC(C_3)$. A variant of this case is produced when both countermeasures have the same surface values, in which the surface coverage of the union and the intersection are calculated as $SC(C_4 \cup C_5) = SC(C_4 \cap C_5) = SC(C_4) = SC(C_5)$.

4.4 Countermeasure Selection Process

The process for selecting optimal countermeasures is performed in two steps: Individual Countermeasure Evaluation, which determines the parameters associated to the intrusion or attack (e.g., ALE and AIV) and evaluates the RORI index for all individual solutions (using equation 3.2); and Combined Countermeasure Evaluation, which determines the parameters associated to the combined solutions (e.g., ARC and RM) and evaluates the RORI index for all combined solutions.

4.4.1 Requirements

A combined countermeasure results from the simultaneous implementation of two or more countermeasures to mitigate a given attack. A combined solution is therefore analysed as a single countermeasure with a combined cost and a combined effectiveness. The combination of security solutions is allowed if the following requirements are met:

1. The countermeasures are not mutually exclusive,

2. The total surface coverage of one countermeasure is totally covered by another countermeasure with an equal surface coverage.

4.4.2 Process Description

The process for the selection and ranking of combined security solutions is depicted in Figure 4.2.

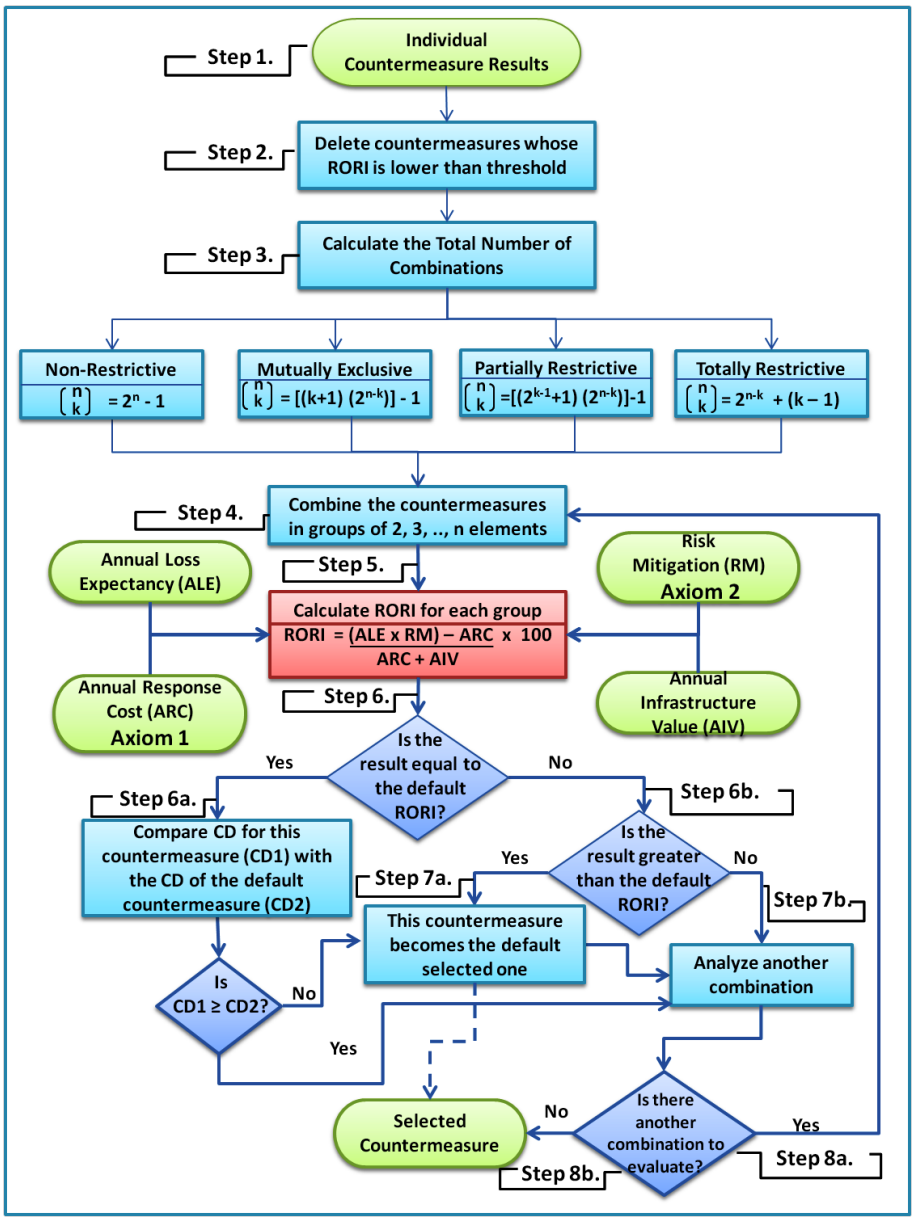

FIGURE 4.2 - Countermeasure Combination Flowchart

The process takes as input the results obtained from the evaluation of individual countermeasures (step 1), as described in section 3.1.2. The system then eliminates those countermeasures for

which the RORI index is below the average or below a predefined threshold (step 2). This action helps the system optimize the evaluation process.

Once we have the list of combinable countermeasures (step 3), we calculate the total number of possible combinations from the list of candidate countermeasures. Then, it is possible to generate groups of 2, 3, ..., n countermeasures, where “n" is the total number of elements to be combined (step 4).

The RORI metric is then calculated for each combination (step 5), taking into account that for a combined solution, the cost is calculated as the sum of all the individual countermeasure costs (Axiom 1) and the risk mitigation of a combined solution is calculated as the probability of the union of events (Axiom 2). The Annual Infrastructure Value and the Annual Loss Expectancy remains unchangeable for all combined solutions.

In order to evaluate all the combined countermeasures, it is necessary the use of axioms 1 and 2 described in section 4.3.1. In step 6, the obtained RORI value is compared with the value by default (the highest RORI value obtained in the Single Countermeasure Evaluation).

If the resulting RORI is equal to the default value, the system checks for the countermeasure cost (ARC) and selects the one with the lowest value (step 6a). Otherwise, the system compares if the current RORI is greater to the default value (step 6b), in such a case, the countermeasure becomes the selected one, and it overrides the default value (step 7a). However, if the resulting RORI is lower than the default value (step 7b), the system checks for another combination to evaluate (step 8a).

When no other combination is left to be evaluated (step 8b), the system selects the default countermeasure as the best solution, since it is the one that provides the highest RORI index.

4.5 Conclusions

In this chapter we introduced an approach to select optimal security countermeasures based on the Return On Response Investment (RORI) index, making it possible to evaluate combined countermeasures to mitigate the effects of intrusions or attacks.

Our solution is divided into two phases: The Evaluation of Individual Countermeasures, which determines the parameters associated to the intrusion or attack (e.g., ALE and AIV) and evaluates the RORI index for all individual solutions; and The Evaluation of Combined Countermeasures, which determines the parameters associated to the combined solutions (e.g., ARC and RM) and evaluates the RORI index for all possible combinations of security measures. As a result, the system is able to rank and select the individual or combined countermeasure that provides the highest cost-effectiveness ratio, thus the highest benefit to the organization. Even if the countermeasures overlap, our solution provides an approximation of the area covered by all countermeasures.

In order to show the applicability of our model, the next chapter provides a deployment of the RORI index and the operations required to evaluate and select optimal countermeasures over two real case scenarios. The first one, a Mobile Money Transfer Service is provided by a large telecommunication company based in France, and the second one, a Critical Infrastructure Control Process (Dam) is provided by a medium-sized telecommunication company based in Italy.

Chapter 5

Application of the Countermeasure Selection Model in Single Attack Scenarios

Knowledge is of no value unless you put it into practice.

Anton Pavlovich Chekhov

-

Contents

The evaluation of multiple countermeasures in complex systems generally requires the implementation of a prototype to simulate the system and deploy the model over a real case study. This chapter addresses the deployment in two use cases provided by the MASSIF Consortium [Con11] to show the applicability of the Return on Response Investment (RORI) model in real world scenarios. In addition, we provide examples on how to evaluate the RORI parameters in order to select optimal countermeasures. Section 5.1 introduces a Mobile Money Transfer Service (MMTS) case study, and discusses an account takeover attack scenario, as well as the evaluation and selection of individual and combined countermeasures. Section 5.2 describes a Critical Infrastructure Process Control case study, and discusses a Control Station Hacking Attack scenario, as well as the evaluation and selection of individual and combined countermeasures. More information regarding the two use cases can be found in [Con11c]. Finally, Section 5.3 concludes

with a discussion regarding the deployment of the use case scenarios.

5.1 Case Study: Mobile Money Transfer Service (MMTS)

The Mobile Money Transfer Service (MMTS) is a system where virtual money, called mMoney (for mobile money), is used to carry out financial operations such as purchasing goods, cashing salaries, paying bills, taking loans, paying taxes or receiving social benefits. Each actor of the system (e.g., customer, retailer, merchant, bank, etc.) has an mWallet where the virtual money is stored. mWallets are not stored locally inside the mobile phone but in a platform maintained by a central authority.

The MMTS responds to the need of improving the security infrastructure of a service that uses a mobile telephone to perform various types of money transfers and transactions. This use case discusses an account takeover attack and evaluates all possible countermeasures for its mitigation. The data related to this use case, as well as all numeric values used to estimate costs and benefits, have been provided and validated by France Telecom - Orange Group[1].

5.1.1 Regular Operation

For this service to work, the operator is associated with a bank which issues mMoney. The operator and its associated bank have to report specific data and activities to the central bank and they must ascertain that the amount of mMoney circulating in the system remains constant. The number of transactions carried out in a day is roughly estimated to 200,000 transactions per day. This type of service provides a new payment system which could be misused for money laundering and terrorist funding. Figure 5.1 describes the work-flow of this scenario.

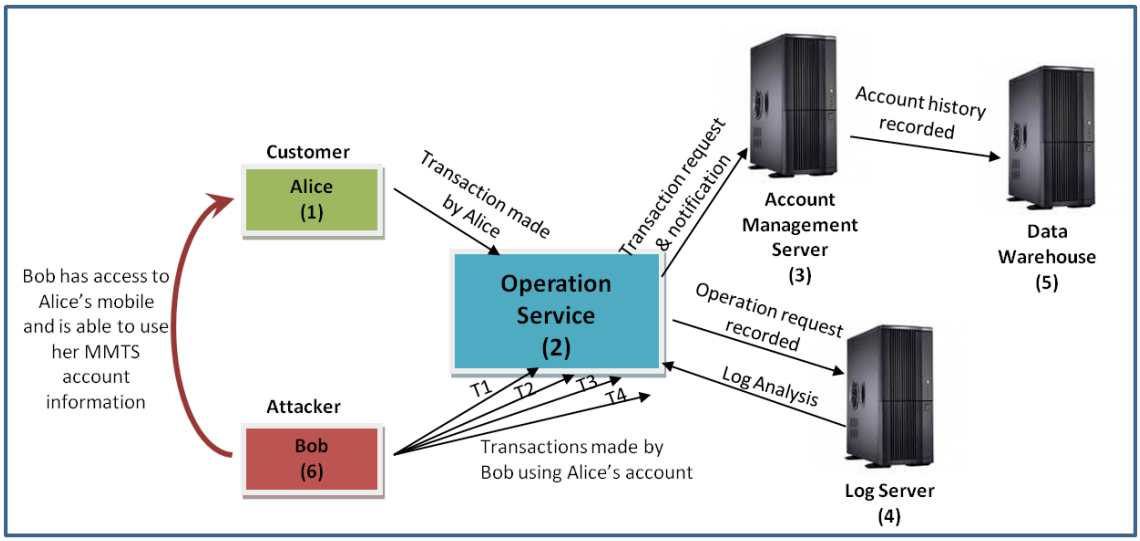

FIGURE 5.1 - Mobile Money Transfer Service Work-flow

In Figure 5.1, Alice (1) is a customer of the Mobile Money Transfer System (MMTS), who generally uses the system two or three times per month in order to pay some bills (e.g., electricity, telephone). The operation service (2) receives all the transactions requested by the MMTS

[1] http://www.orange.com/en/group

customers. The service is composed by the account management server, the log server, and the data warehouse. The account management server (3) keeps all the information regarding the user's behaviour. The log server (4) registers data that are relevant to analyse abnormal activities such as: failed authentications, requests for PIN modification, transaction requests, etc. The data warehouse (5) contains historical data about accounts, which is useful to analyse customer's behaviour and detect frauds.

The system is able to detect abnormal behavior by analysing the logs and matching monitored operations with logic rules for irregular behaviour [GMHD11, GMHD12, Con11b]. If the monitored operation does not match the rules, it is allowed to be executed; on the contrary, the system classifies it as abnormal. A complete example of the inference rules used for the MMTS use case scenario is given in Appendix B.

5.1.2 Attack Scenario: Account Takeover

An account takeover is a password-based attack that exploits vulnerabilities on the user's side (e.g., social engineering, key-loggers, etc.) and steals the mobile user account to perform transactions in favour of the attacker. An increment in the number of transactions performed in a period of time, or a raise in the amount of money being transferred for a particular user, as compared to the normal user behaviour is interpreted as an account takeover attack (an attacker performs some transactions on the MMTS platform using the credentials of a legitimate user).

In the example, Bob (6), the attacker, after a couple of attempts to get authenticated using Alice's credentials, gains access to Alice's MMTS account. He then performs transactions, such as purchasing items and transferring money to a bank account under his control. As a result, the system (2) detects the anomalous behaviour from the logs: For the past two hours, the user Alice, who has always had a regular behaviour (no more than 3 accesses to the system per month), has already used the MMTS several times within a day to carry out several transactions.

According to France Telecom - Orange Group, an account takeover attack has an estimated "Minor" severity level[2] (equivalent to 100 €) and a "High" likelihood[1] (once per month, equivalent to 12). The Annual Loss Expectancy (ALE) for this attack is expected to be 1200 €, and the Annual Infrastructure Value (AIV) is calculated as the value of all the Policy Enforcement Points (PEP) that are needed to be deployed in the preliminary phase of the system architecture. Table 5.1 lists the PEP for this scenario and summarizes information regarding their type, costs and mitigated threats [3]. The AIV corresponds to the annual cost of purchasing, licensing, implementing and/or maintaining a security equipment in the MMTS Infrastructure.

From the list of equipments proposed in Table 5.1, we select 1 Network Intrusion Detection System (Snort), 1 Network Monitoring (FreeNATS), 1 Firewall (Comodo), 1 Intrusion Prevention System (Cisco SA 500 Series), and a Strong Authentication Method (Software Token) as the security solutions to be deployed in a regular system architecture for a Mobile Money Transfer Service. A combination of all of them provide a wider and more complete coverage of the different threats to which the system is exposed. The AIV is therefore estimated as 2600€ (the cost of all the selected solutions).

[2] The estimated values for the severity and likelihood of an attack may vary from one country to another and depend greatly on the standard of living of each country.

[3] More information about these threats can be found in Appendix B

TABLE 5.1 - Security Equipments for a Mobile Money Transfer Service

	PEP	Type	AIV	Threats that mitigate							
				T1	T2	T3	T4	T5	T6	T7	T8
E1	Intrust	HIDS	800€	✓		✓					✓
E2	Tripwire	HIDS	250€	✓		✓					✓
E3	Verisys	HIDS	400€	✓		✓					✓
E4	Snort	NIDS	400€	✓	✓	✓	✓	✓	✓	✓	✓
E5	NetCrunch	Net. Monitoring	1500€	✓	✓	✓	✓	✓	✓	✓	✓
E6	FreeNATS	Net. Monitoring	500€	✓	✓	✓	✓	✓	✓	✓	✓
E7	Comodo	Firewall	300€	✓	✓	✓	✓		✓		✓
E8	Endian	Firewall	150€	✓	✓	✓	✓		✓		✓
E9	Cisco	IPS	1000€	✓	✓	✓	✓	✓	✓	✓	✓
E10	Kaspersky	Antivirus	300€						✓		✓
E11	OS update	OS Hardening	500€				✓		✓		✓
E12	Soft. Token	Auth. Method	400€	✓	✓	✓	✓	✓	✓	✓	

T1 Trafficking Collection T2 Hiding User Identity T3 Scams
T4 Account Takeover T5 Employee Complicity T6 Denial of Service
T7 Money Creation/Destruction T8 Other threats (e.g. malware, virus)

5.1.3 Individual Countermeasure Evaluation for the Account Takeover Attack

In order to react to an intrusion attempt it is necessary to change/update access control policies or implement new mitigation strategies. Following the methodology proposed in Chapters 3 and 4, we evaluated the following 9 countermeasures to mitigate an Account Takeover Attack in the MMTS platform. All countermeasures and the associated data have been validated by France Telecom - Orange Group

Description of Candidates:

C1.1 NOOP: This solution considers to accept the risk and does not require any modifications. The cost and risk mitigation level are equal to zero.

C1.2 Deny Transaction: This alternative allows the user to authenticate but he/she is not able to perform any kind of transaction.

C1.3 Deactivate User Account: A temporarily deactivation of the user account (e.g., for a period of 24, 48 or 72 hours) will prevent attackers from succeed.

C1.4 Reduce Transaction Amount: This candidate limits the use of the suspected user account to perform transactions for a maximum amount of money (e.g., up to 30€, 50€, 100€).

C1.5 Reduce Number of Transactions: This alternative limits the user to perform a controlled number of transactions per day (e.g., 2, 3, or 5 transactions per day), meaning that for this specific account, MMTS users can only perform transactions that cannot exceed a predefined threshold.

C1.6 Active Alert Mode: This countermeasure fires an alert indicating that the defined user account is suspected to be under attack.

C1.7 Keep the Account under Surveillance: This candidate will take the user account into quarantine in order to closely evaluate its behaviour and punctually block operations.

C1.8 Activate Two-factor Authentication: This alternative requests an additional authentication (e.g., pass phrase, challenge response, PIN), in order to authenticate the user and authorize him/her to perform the required transaction.

C1.9 Deactivate Multiple Transaction Requests: This security measure limits the user to emit only one transaction at a time.

Table 5.2 shows the different countermeasures that are proposed to react to an account takeover attack and details information regarding the Policy Enforcement Points (PEP), the Annual Response Cost (ARC), the countermeasure Surface Coverage (SC), the Effectiveness Factor (EF), the Risk Mitigation value (RM), the RORI index, and the restrictions (partial or total restrictions to be combined with other solutions) associated to each countermeasure. These figures have been provided by France Telecom - Orange Group.

TABLE 5.2 - Individual Countermeasure Evaluation (Account Takeover Attack)

Counter Measure	**PEP**[1]	**ARC**[2]	**SC**[3]	**EF**[4]	**RM**[5]	**RORI**[6]	**Restriction**
C1.1	-	0€	0.00	0.00	0.00	0.00%	Totally Rest.
C1.2	E7	60€	0.85	0.85	0.72	30.34%	Totally Rest.
C1.3	E9	55€	0.85	0.80	0.68	28.66%	Totally Rest.
C1.4	E4	35€	0.70	0.75	0.53	25.77%	C1.1,C1.2,C1.3
C1.5	E4	30€	0.70	0.85	0.60	22.81%	C1.1,C1.2,C1.3
C1.6	E4	25€	0.60	0.70	0.42	18.25%	C1.1,C1.2,C1.3
C1.7	E6	40€	0.60	0.70	0.42	17.58%	C1.1,C1.2,C1.3
C1.8	**E12**	**50€**	**0.85**	**0.90**	**0.77**	**32.75%**	**C1.1,C1.2,C1.3**
C1.9	E9	35€	0.80	0.80	0.64	27.82%	C1.1,C1.2,C1.3

[1]Policy Enforcement Point [2]Annual Response Cost [3]Countermeasure Surface Coverage
[4]Effectiveness Factor [5]Risk Mitigation Level [6]Return On Response Investment

From the list of proposed countermeasures, the first alternative (C1.1) is to accept the risk by executing NOOP. This action does not require any modifications and therefore the risk remains the same. C1.1 is totally restrictive and its associated cost is 0, since no countermeasure is implemented. As a result, the RORI index for C1.1 is 0.00%.

The second and third options are totally restrictive and suggest to avoid the attack either by denying the transaction (C1.2), or by deactivating temporarily the user account (C1.3). As a result, the attack is greatly reduced ($70 - 75\%$). The RORI index for C1.2 and C1.3 is 30.34% and 28.66% respectively. Since the RORI index of C1.2 is greater than the one by default, C1.2 becomes the default countermeasure.

Alternatives four and five propose to reduce the transaction amount (C1.4) or to reduce the number of transactions per day (C1.5), as part of a strategy to prevent attackers from stealing large amount of money from their victims without deactivating the user account. As a result, the attack is mitigated $53 - 60\%$. The RORI index for C1.4 and C1.5 is estimated to be 25.77% and 22.81% respectively. The default countermeasure remains unchanged since the RORI index for C1.4 and C1.5 is lower than the one by default.

Countermeasures six and seven recommend to activate the alert mode (C1.6),or to keep the user account under surveillance (C1.7) e.g., for a period of 24, 48 or 72 hours. Both countermeasures

have a risk mitigation value of 42% and a RORI index of 18.25% and 17.58% respectively. The default countermeasure remains unchanged since the RORI results for these two countermeasures are lower than the one by default.

The two remaining countermeasures (C1.8, C1.9) recommend to activate additional authentication methods (e.g., two-factor authentication request) and to deactivate multiple transaction requests. By implementing these solutions, the risk is expected to be mitigated $64 - 77\%$. As a result, the RORI index for C1.8 and C1.9 is 32.75% and 28.55% respectively.

The highest RORI index from the evaluation of the different countermeasures to react against an account takeover attack correspond to 32.75%, the risk is expected to be mitigated 77% with an annual response cost of 50€. As a result, countermeasure eight (C1.8), which recommends to activate additional authentication methods (e.g., two-factor authentication request), becomes the selected single countermeasure for an account takeover attack in the MMTS System.

5.1.4 Combined Countermeasure Evaluation for the Account Takeover Attack

Since the average RORI for the previous list of countermeasures is 22.66%, we select C1.4, C1.5, C1.8 and C1.9 as the candidates to combine (their RORI index is greater or equal to the average and they are not totally restrictive). The maximum number of possible combinations is $\binom{4}{2} = 2^4 - 1 = 15$.

In order to perform the combined countermeasure evaluation, we discard totally restrictive countermeasures (solutions that can not be combined because they create conflicts on the system). These countermeasures present the flag "all" in the restriction column from Table 5.2. We consider, however, mutually exclusive and partially restrictive countermeasures while calculating the maximum number of combinations. Table 5.3 summarizes these results.

TABLE 5.3 - Combined Countermeasure Evaluation (Account Takeover Attack)

N	Candidates	ARC[1]	SC[2]	EF[3]	RM[4]	RORI
1	C1.4	35€	0.70	0.75	0.53	25.77%
2	C1.5	30€	0.70	0.85	0.60	22.81%
3	C1.8	50€	0.85	0.90	0.77	32.75%
4	C1.9	35€	0.80	0.80	0.64	27.82%
5	C1.4+C1.5	65€	0.55	0.75	0.71	29.42%
6	**C1.4+C1.8**	**85€**	**0.63**	**0.85**	**0.83**	**33.87%**
7	C1.4+C1.9	70€	0.60	0.80	0.76	31.31%
8	C1.5+C1.8	80€	0.63	0.75	0.82	33.79%
9	C1.5+C1.9	65€	0.60	0.75	0.72	29.76%
10	C1.8+C1.9	85€	0.73	0.80	0.83	33.71%
11	C1.4+C1.5+C1.8	115€	0.48	0.75	0.83	32.39%
12	C1.4+C1.5+C1.9	100€	0.45	0.75	0.76	29.85%
13	C1.4+C1.8+C1.9	120€	0.53	0.80	0.83	32.15%
14	C1.5+C1.8+C1.9	115€	0.53	0.75	0.83	32.23%
15	C1.4+C1.5+C1.8+C1.9	150€	0.38	0.75	0.83	30.71%

[1] Combined Annual Response Cost
[2] Combined Countermeasure Surface Coverage
[3] Combined Effectiveness Factor
[4] Combined Risk Mitigation Level

From Table 5.3, ARC and RM are calculated using Axiom 1 and Axiom 2 respectively, as

explained in Section 4.3.1. SC is calculated using Equation 4.15, EF follows Equation 4.13 that suggest to select the minimum Effectiveness Factor from the group of combined countermeasures. The RORI index is calculated using Equation 3.2, as proposed in Section 3.1.2.

After comparing the RORI index on all the different alternatives, we determined that the best solution reduces 83% of the risk, and the RORI index is expected to be 33,87%. As a result, alternative 2 (a combination of C1.4 and C1.8), which proposes to reduce the transaction amount and to activate double authentication process becomes the selected combined countermeasure for an account takeover attack in the Mobile Money Transfer Scenario.

5.2 Case Study: Critical Infrastructure Process Control

This section describes a use case provided by Epsilon Italy SRL[4], a telecommunication enterprise based in Naples (Italy), operating in the field of Information and Communication Technology (ICT) services. The case study responds to the needs of improving the security of a system whose mission is to control a critical infrastructure, specifically a Dam. The following subsections describe the case study and detail a scenario of attack, as well as the general operations required to rank, select and deploy optimal countermeasures.

5.2.1 General Description

A dam is a barrier that impounds water or underground streams. Dams are generally used for water supply, hydroelectric power generation, irrigation, and water activities [Con11]. There are many different parameters to be monitored to assess the safety of the dam and foresee possible failures or anomalies. The reference system architecture involves SCADA[5] components. Three main groups of components are identified in the system: control devices (i.e. Control Station and Visualization Station), input and output (I/O) devices (i.e. sensors and actuators components), and a SCADA gateway (as shown in Figure 5.2).

As depicted in Figure 5.2, the *supervisor* gathers and delivers data from and to a Control Station and a Visualization Station. The *Control Station* allows on-line and real time data analysis as well as data and event storing. The *Visualization Station* presents historical data stored in a database through a web server interface. *Sensors and actuators* are responsible for retrieving measurements related to specific physics phenomena. The *SCADA gateway* is responsible for evaluating, processing, storing retrieved measurements and elaborating proper commands for the actual system.

Dam operations must be supported by continuous monitoring of the key parameters. The adoption of *Wireless Sensor Networks* (WSN) provides means to monitor these parameters in a relatively inexpensive way. A WSN is composed of several wireless enabled sensors that can be deployed across the dam to monitor environmental, geophysical, and geotechnical parameters. The sensors are able to self-organize, creating a wireless mesh network that allows measured data to reach a data collection device.

In regular operations, when the water level of the dam exceeds an alerting threshold (max, min or overflow thresholds), the monitoring station generates an alert, notifying the control station about the corresponding alerting level and the water level growth rate. The control station triggers the discharging operation specific for the alerting threshold until the monitoring station notifies that the water level has reached a regular level.

[4] www.epsilonline.com/index.php

[5] A SCADA system is an industrial control system targeted to monitor and control infrastructure, industrial, and facility based processes.

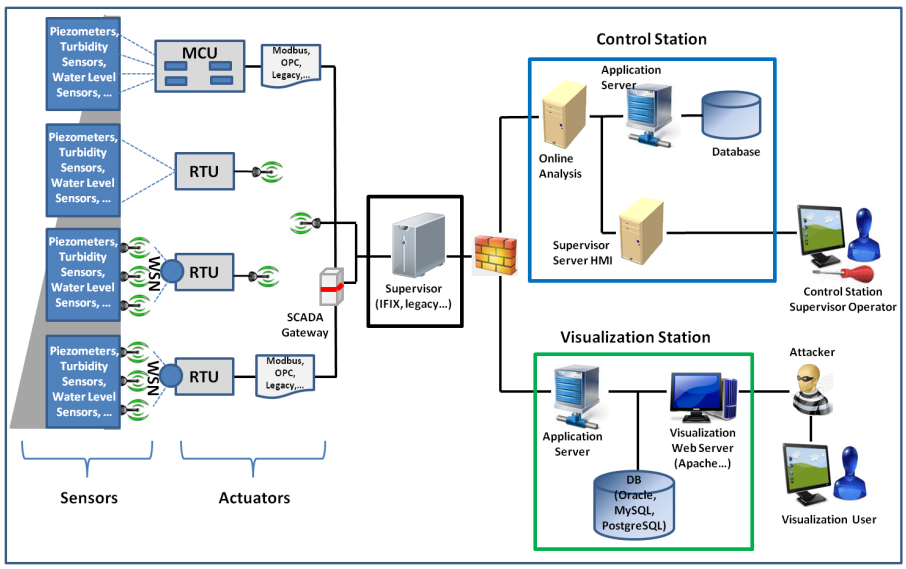

FIGURE 5.2 - Critical Infrastructure Process Control (dam) Scenario Components

5.2.2 Control Station Hacking Attack

A machine connected to the visualization station succeeds in controlling remotely a machine in the control station (through password theft, bug exploit, or other techniques). Consequently, the malicious user modifies sensor settings (e.g., policies and alerting thresholds) and sends commands to the actuators, asking them to maintain the monitoring values under the new thresholds.

The hypothesis is that the attacker has access to the remote machine with a stolen password. He does not install any malicious software, but he knows packet format, and generates well-formed packets. The visualization station is located in a DMZ branch of the LAN, while the control station is in a protected branch of the LAN. The traffic between the two subnets passes through a firewall. A constraint of the system is that the commands to drive the actuators can be made only from the control station.

According to Epsilon Italy SRL, a control station hacking attack has an estimated “Serious" severity (equivalent to 1,000,000€) and an estimated “Negligible" likelihood (equivalent to 0.05 times per year), since the control station is supposed to be highly protected. The ALE for a control station hacking attack is expected to be 50,000€/year and the annual infrastructure value (AIV) is calculated as the sum of all the security equipments' costs that need to be deployed in the preliminary phase of the system architecture. Table 5.4 proposes a list of security equipments for this scenario and summarizes the information regarding their AIV and the threats they mitigate[6].

Taking into account that a combination of security solutions provide a wider and more complete coverage of the different threats to which the system is exposed, we selected Sensor Tamper Resistance, IDS/IPS, Firewall, Access Control Mechanisms, System Behaviour Monitoring and Communication Protocols (e.g., DNP3, ICCP) as the security solutions to deploy for a Critical Infrastructure Process Control . The Annual Infrastructure Value is therefore estimated at 8.700€ (the cost of all the selected solutions).

[6]All threats are defined in the Glossary of Terms of this dissertation. The complete case study can be found in MASSIF Deliverable 2.1.1 [Con11]

TABLE 5.4 - Security Equipments for a Critical Infrastructure Process Control

Policy Enforcement Point (PEP)	**AIV**	**Mitigated Threats**							
	(€)	T1	T2	T3	T4	T5	T6	T7	T8
E1 Stronger Cryptography	2000					✓	✓	✓	✓
E2 Sensor Tamper Resist.	1000€	✓	✓			✓	✓	✓	
E3 Back-up Power Supply	1500					✓			
E4 IDS-IPS	2500	✓	✓	✓	✓	✓	✓	✓	✓
E5 Firewall	2000	✓	✓	✓	✓	✓	✓	✓	✓
E6 Anti-virus/malware	500	✓	✓						✓
E7 Access Control	1500	✓	✓	✓	✓	✓	✓	✓	✓
E8 System Monitoring	1200	✓	✓	✓	✓	✓	✓	✓	✓
E9 Communication Protoc.	500	✓	✓		✓	✓		✓	

T1 Water Level Sensor Compromise T2 Tilmeter Compromise T3 Administration Password Theft
T4 Hazardous Water Release T5 Hydroelectric Power Plant Anti-Islanding Hacking
T6 Visualization Station Misuse T7 Control Station Hacking T8 Infectious threats

5.2.3 Individual Countermeasure Evaluation (Control Station Hacking Attack)

In partnership with Epsilon Italy SRL, we defined the following 7 countermeasures to mitigate a control station hacking attack. All the data related to the cost and benefits of countermeasures have been validated by Epsilon Italy SRL [Con11c].

Description of Candidates:

C2.1 NOOP: This candidate considers to accept the risk by executing no operation. This action does not require any modifications, therefore the cost and risk mitigation are equal to zero.

C2.2 Privilege Separation: Enforce separation of privileges is useful by preventing users to perform actions they are not allowed.

C2.3 Active Alert Mode: This alternative proposes to fire an alert indicating that the control station is suspected to be under attack.

C2.4 Disable Remote Connections to the Control Station: Allow only local connections to the control station to authorized users (Switch from "remote" to "not-remote").

C2.5 Enable Multiple Monitoring Indication: This countermeasure activates two or more monitoring systems to verify the water level indication obtained by the sensors.

C2.6 Restart Sensor Settings: It erases the current sensor values and request for new thresholds.

C2.7 Activate Back-up Sensors: Switch "off" current sensors and switch "on" back-up sensors.

Table 5.5 summarizes the parameters for each individual countermeasure. The first column shows the different alternatives proposed. The second column shows the Policy Enforcement Point associated to each countermeasure. The third column details the estimated Annual Response Cost

(ARC). The fourth and fifth columns show the Surface Coverage (SC) and Effectiveness Factor (EF) associated to each solution.

Each countermeasure has an associated SC and EF factors that indicate the level at which the countermeasure is expected to mitigate the effects of a given attack. For instance, enforcing separation of privilege (C2.2) covers around 60% of the Control Station Hacking Attack with an effectiveness of 80%. The application of this countermeasure results in a Risk Mitigation (RM) of 48%. RM is shown in column sixth and represents the multiplication of SC and EF metrics. RORI results are displayed in column seventh and finally, information regarding the restrictions to combine the countermeasures is provided in column eighth.

TABLE 5.5 - Individual Countermeasure Evaluation (Control Station Hacking Attack)

Counter Measure	**PEP**[1]	**ARC**[2]	**SC**[3]	**EF**[4]	**RM**[5]	**RORI**[6]	**Restriction**
C2.1	-	0€	0.00	0.00	0.00	0%	all
C2.2	E7	200€	0.60	0.80	0.48	267%	C1
C2.3	E4	300€	0.45	0.60	0.27	147%	C1
C2.4	E5	500€	0.85	0.70	0.60	318%	C1
C2.5	E8	700€	0.75	0.85	0.64	332%	C1
C2.6	E9	200€	0.55	0.70	0.39	214%	C1
C2.7	**E4**	**400€**	**0.70**	**0.90**	**0.63**	**342%**	**C1**

[1]Policy Enforcement Point [2]Annual Response Cost [3]Countermeasure Surface Coverage
[4]Effectiveness Factor [5]Risk Mitigation Level [6]Return On Response Investment

From the set of proposed countermeasures, the highest RORI value corresponds to C2.7 (Activate Back-up Sensors), with a cost of 400€, a risk mitigation of 63%, and a benefit of 342%. This candidate solution becomes the selected single countermeasure for a Control Station Hacking Attack.

5.2.4 Combined Countermeasure Evaluation (Control Station Hacking Attack)

From the results obtained out of the individual countermeasure evaluation we know that all countermeasures can be combined except for C2.1 (NOOP), since this is a totally restrictive solution. Taking into account that the average RORI for the group of countermeasures is 231%, we select C2.2, C2.4, C2.5 and C2.7 as the candidates for combination (their RORI index is greater than the average).

In order to evaluate all possible combinations for the selected countermeasures, we apply Equation 4.8 for non-restrictive countermeasures, with n=4. The maximum number of possible combinations is $\binom{4}{k} = 2^4 - 1 = 15$. We consider only those countermeasures that do not present the flag "all" in the restriction column from Table 5.5, since they do not originate conflicts on the system while implementing them simultaneously. Table 5.6 summarizes the results obtained on the evaluation of all combined countermeasures for a Control Station Hacking Attack.

The first two columns from Table 5.6 show the possible combinations from the list of suitable candidates. The third column shows the annual response cost (ARC) that results from the combined solution. ARC is determined according to axiom 1 (section 4.3.1).

The fourth and fifth columns show respectively the Surface Coverage (SC) and Effectiveness Factor (EF) associated to each combination. SC is calculated using Equation 4.15, EF follows

TABLE 5.6 - Combined Countermeasure Evaluation for a Control Station Hacking Attack

N	Combination	ARC[1]	SC[2]	EF[3]	RM[4]	RORI
1	C2.2	200€	0.60	0.80	0.48	267%
2	C2.4	500€	0.85	0.70	0.60	318%
3	C2.5	700€	0.75	0.85	0.64	332%
4	C2.7	400€	0.70	0.90	0.63	342%
5	C2.2+C2.4	700€	0.53	0.70	0.71	369%
6	C2.2+C2.5	900€	0.48	0.80	0.74	375%
7	C2.2+C2.7	600€	0.45	0.80	0.75	397%
8	C2.4+C2.5	1200€	0.68	0.70	0.76	372%
9	**C2.4+C2.7**	**900€**	**0.63**	**0.70**	**0.79**	**401%**
10	C2.5+C2.7	1100€	0.58	0.85	0.78	386%
11	C2.2+C2.4+C2.5	1400€	0.40	0.70	0.77	369%
12	C2.2+C2.4+C2.7	1100€	0.38	0.70	0.80	398%
13	C2.2+C2.5+C2.7	1300€	0.33	0.70	0.75	360%
14	C2.4+C2.5+C2.7	1600€	0.50	0.70	0.81	379%
15	C2.2+C2.4+C2.5+C2.7	1800€	0.25	0.70	0.78	355%

[1]Combined Annual Response Cost [2]Combined Countermeasure Surface Coverage
[3]Combined Effectiveness Factor [4]Combined Risk Mitigation Level

Equation 4.13 that suggest to select the minimum Effectiveness Factor from the group of combined countermeasures. The Risk Mitigation level (RM) is shown in column sixth and results after the application of a combined solution. RM is calculated according to axiom 2 (section 4.3.1). RORI results are displayed in column seventh. RORI is calculated as proposed in equation 3.2

After comparing the RORI index on all the different alternatives, we determined that the optimal solution to secure the control and visualization station network costs 900€, reduces the risk in 79%, and results in a RORI index of 401%. Alternative 5 (a combination of C2.4 and C2.7), which proposes to disable remote connections and activate back-up sensors, becomes therefore the selected combined countermeasure for a Control Station Hacking Attack in the scenario of critical infrastructure process control.

5.3 Discussion

The deployment of our proposed cost sensitive model over a Mobile Money Transfer Service and a Critical Infrastructure Process Control shows the applicability of the RORI model and demonstrates that in most of the cases, combined countermeasures provide a higher RORI index and thus a higher benefit to the organization.

It is important to mention that the evaluation of countermeasures is performed in real case scenarios where it is possible to select single or combined countermeasures as a reaction strategy. Furthermore, performing an automatic evaluation of countermeasures is a key element to react rapidly and appropriately to a given attack, especially for critical infrastructure processes (e.g. the Dam scenario presented in the second part of this chapter).

In addition, the combination of two or more security policies covers a larger surface of the risk, which results into a higher risk mitigation level. However, it is worth noting that a higher risk mitigation value does not always mean a higher RORI index, nor the least expensive solution is

always the most interesting one. Several parameters are taken into account in order to select the solution that provides the highest cost-effectiveness ratio to the system.

In Part II, we will propose a methodology to evaluate and select countermeasures in a scenario of multiple attacks. For this purpose, the current approximation of the countermeasures overlap becomes insufficient. To solve this issue, we will study the total volume exposed to a given attack, and the portion covered by each individual attack in the system, as well as their countermeasure coverage.

Part II

Countermeasure Selection for Multiple Attack Scenarios

Chapter 6

Attack Volume: Formalization and Modeling

We find no sense in talking about something unless we specify how we measure it; a definition by the method of measuring a quantity is the one sure way of avoiding talking nonsense...

Relativity and Common Sense
Sir Hermann Bondi - 1964

Contents

In the first part of this dissertation, we proposed to use an estimation of the attack surface to compute the surface coverage of one or multiple countermeasures. This approach allowed us to obtain an approximation to the value of the Risk Mitigation (RM) parameter, making it possible to evaluate the Return On Response Investment (RORI) index in a scenario of individual attacks. However for multiple attack scenarios, this approach is not accurate enough to determine the equipment(s), subject(s) and/or action(s) that are included in the security incidents.

We, therefore, propose in this chapter, an approach that represents the volume of the system, attacks, and countermeasures based on a three-dimensional coordinate system (i.e. user, channel, and resource). We do not use approximations as we have already proposed in Chapter 4; instead, we replace the estimation of the surface coverage (SC) by a qualitative value built on the system specifications and infrastructure. As a result, the union and intersection of the different volumes (i.e. system, attacks, and countermeasures) is calculated using geometrical operations.

The coordinates of each element are derived from a URI. The URI is decomposed into three dimensions along the following axes:

1. user - privilege - role
2. channel - protocol - parameter
3. resource - path - machine

We represent each element according to the 3 aforementioned axes. These axes are complementary to each other and relate to the OrBAC model, where the user account is modeled as a subject (with a given role in the organization), that is assigned certain privileges; the channel is modeled as an action (with the required parameters) that carry on protocols to identify resources on the system ; and the resource is modeled as an object (machine) with a path in the URI.

Each URI is eventually represented as a parallelepiped in this space. We will define all the users, channels and resources available in these 3 axes. We will then model these URI elements into the corresponding axis in the coordinate system. We need, therefore, a bijection between the URIs and the coordinate system in order to make the appropriate representations. The coordinates (i.e. account, channel, resource) define a parallelepiped with a corresponding volume in our system.

The CARVER methodology, discussed in Section 6.2.3, is used to give an appropriate weight to each element composing the axes in our coordinate system. Weights are associated to the criticality of a given component in our system, making it possible to connect the volumes with the risks. For instance, a regular user might be assigned one unit, whereas an administrator might be assigned 4 units (big volumes are equivalent to high risk, whereas small volumes are equivalent to low risk). Once we determine the union and intersection of the attack volumes, we will be able to compare their associated risks.

Two concepts are considered while comparing two or more risk volumes: Residual risk [1], which results when the risk volume is higher than the countermeasure volume; and Collateral damage [2], which results when the countermeasure volume is higher than the risk volume.

The rest of the chapter is structured as follows: Section 6.1 introduces the state of the art of attack surface. It also introduces the uniform resource identifier and the OrBAC model. Section 6.2 introduces the coordinate system and its attack dimensional vectors. Section 6.3 presents the contribution of axes in the volume calculation. Section 6.4 details the calculation of the system, attack, and countermeasure volume. Finally, a discussion is presented in Section 6.5.

6.1 State of the Art

This section introduces three concepts: firstly, the attack surface model, which presents a methodology to measure and compare different attacks in a given system. Secondly, the Uniform Resource Identifier (URI), which identifies the main components used in accessing a computing resource. Lastly, the Organization-Based Access Control (OrBAC) model, which formalises the process to define security policies in a given organization.

6.1.1 Attack Surface

This subsection presents the related work in the attack surface measurements, and highlights their benefits and constraints.

6.1.1.1 Operating Systems Attack Surface

According to Howard et al. [How04, HW07], the attack surface of an application is the union of code, interfaces, services, protocols, and practices available to users. Their definition of the attack surface focuses on the accessibility to unauthorised users. Intuitively, the more exposed the system's surface, the more likely the system is to be attacked. Thus, mechanisms such as improving access controls, having stronger encapsulation processes, improving authentication of users, etc., help in increasing the system's security while reducing its attack surface.

Three dimensions are considered to determine the attack surface of an operating system (e.g. Linux, Windows):

1. Target and enablers: An attack target is a process or resource on a system that is subject to be attacked, and that plays a critical role in the adversary's achieving his/her goal. The term enabler identifies any accessed process or data resource used as part of the means of the attack.
2. Channels and protocols: Channels are the means by which an attacker gains access to the target on the system, whereas protocols determine the rules of interactions among the parties communicating in a channel.
3. Access rights: They refer to the privileges or rights (e.g., read, write, execute) that are associated to each process or data resource of a state machine.

As a result, the more targets, the more channels and the more generous access rights, the larger the attack surface. However, this method presents the following shortcomings:

[1] The risk that remains after the controls are taken into account.
[2] The damage to things that are incidental to the intended target.

- The approach does not provide a systematic method to identify or assign weights to the attack vectors. The measurement method requires therefore, a security expert to assign weights. Non-experts can not use the method easily.
- The attack vectors have been identified based on the history of attacks on each operating system (i.e. Windows, Linux). However, since the process is performed manually, it is not possible to determine if all attack vectors have been identified.
- The approach focuses on measuring the attack surfaces of operating systems and cannot be generalized to other software systems such as web servers, IMAP servers, and software applications.

6.1.1.2 Software Systems Attack Surface

Manadhata et al. [MWFM06, MKW08, Man08, MW10], define a system's attack surface as the subset of resources used to attack a system. Not all resources are part of the attack surface, nor all of them equally contribute to the attack surface measurement. An attacker may use the system's entry and exit points, channels, and untrusted data items to attack the target. The set of entry and exit points are methods (e.g., get, read, write, print,...) that receive/send data items directly from/to the system's environment. Channels (e.g., TCP, SSL, Unix socket,...) are used to connect to the system and invoke a method. Data items (e.g., files, cookies, database records, registry entries,...) are used by the system'users to send/receive data indirectly into/from the system.

The proposed approach measures the attack surface of a software system (e.g., IMAP server, FTP daemons, Operating Systems) based on the analysis of its source code, through three dimensions: methods, channels, and data. The system's attack surface measurement is quantified as the triple shown in Equation 6.1.

$$AS = \left\langle \sum_{m \epsilon M^{E_s}} der_m(m), \sum_{c \epsilon C^{E_s}} der_c(c), \sum_{d \epsilon I^{E_s}} der_d(d) \right\rangle \tag{6.1}$$

Where:

M^{E_s} is the system's set of entry and exit points;

C^{E_s} is the system's set of channels;

I^{E_s} is the system's set of untrusted data items;

$der_m(m)$ is the damage potential and effort ratio for the methods;

$der_c(c)$ is the damage potential and effort ratio for the channels;

$der_d(d)$ is the damage potential and effort ratio for the data items.

Hence, the smaller the attack surface, the more secure the system. A small attack surface mitigates the risk by making the exploitation harder and by lowering the exploitation's damage. The method to calculate a system's attack surface is summarized as follows:

1. Given a system $S1$, and its environment E, identify a set of entry and exit points (M^E_{S1}), a set of channels (C^E_{S1}), and a set of untrusted data items (I^E_{S1}) of A.
2. Estimate the damage potential-effort ratio, $der_m(m)$, of each method m $\in M^E_{S1}$; the damage potential-effort ratio $der_c(c)$, of each channel c $\in C^E_{S1}$; and the damage potential-effort ratio

$der_d(d)$, of each data item d $\in I^E_{S1}$.

3. Calculate the attack surface using Equation 6.1.

4. The attack surface of a system $S1$ $(M^E_{S1}, C^E_{S1}, I^E_{S1})$ is larger than the attack surface of a system $S2$ $(M^E_{S2}, C^E_{S2}, I^E_{S2})$ if either:

   i $M^E_{S1} \supset M^E_{S2} \wedge C^E_{S1} \supseteq C^E_{S2} \wedge I^E_{S1} \supseteq I^E_{S2}$, or

   ii $M^E_{S1} \supseteq M^E_{S2} \wedge C^E_{S1} \supset C^E_{S2} \wedge I^E_{S1} \supseteq I^E_{S2}$, or

   iii $M^E_{S1} \supseteq M^E_{S2} \wedge C^E_{S1} \supseteq C^E_{S2} \wedge I^E_{S1} \supset I^E_{S2}$

A resource's contribution to the attack surface of a system depends on the potential damage of resources (i.e., the level of harm the attacker causes to the system by using the resource and the effort the attacker expends to obtain the privileges in order to use the resources in an attack). Thus, the higher the potential damage or the lower the effort, the higher the resource's contribution to the attack surface. Similar to the cost-benefit ratio, the resource's contribution is quantified as a damage-effort ratio, where the damage is the benefit of the attacker in using the system's resource and the effort is the cost to the attacker in using the resource.

The measurement of a system's attack surface over three dimensions allows system administrators to choose a dimension appropriate to their need. However, the method presents several shortcomings:

- The approach is only used to evaluate the attack surface of a code source, in the absence of source code, the proposed methodology is useless.
- The damage potential estimation includes only technical impact (e.g., privilege elevation) and not monetary impact (e.g., monetary loss).
- The attack surface model only compares the level of attackability between two similar systems. No attempt has been made to compare the attack surface of different system environments.
- The method does not make assumptions about the capabilities of attackers or resources in estimating the damage potential-effort ratio. Instead, it proposes to assign numeric values to each attribute based on the expert knowledge on the system and its environment.
- The methodology does not allow the security administrator to evaluate multiple attacks occurring simultaneously in a given system.

6.1.1.3 Other Attack Surface Approaches

Petajasoja et al. [PKTT11] propose an approach to analyse a system's attack surface using the Common Vulnerability Scoring System (CVSS). As a result, it is possible to identify most critical interfaces and help in prioritizing the test effort. However, this approach limits the attack surface to known vulnerabilities, it is not meant to be used as a reaction strategy and only compares relative security of similar infrastructures.

Microsoft has recently realised an attack surface analyser tool [Fis12] that identifies changes made to an operating system attack surface by the installation of new software. However the tool can be used only for Windows operating systems and is useless to measure a network attack surface.

Taking into account the aforementioned limitations, we propose a geometric approach to model the information derived from a URI into a coordinate system following the OrBAC formalism. Each axis in our coordinate system is assigned a weighting factor based on the CARVER methodology. The remaining of this section details the CARVER approach, the URI, and the OrBAC model.

6.1.2 CARVER Methodology

Norman [Nor10] and the Federation of American Scientists [oAS91] propose a methodology to measure the priority of each element in a given system, based on the following factors:

- **Criticality (C)** measures the impact that an asset has on carrying out the organization's mission. A target is said to be critical when its destruction or damage has a significant impact on production or service. Criticality depends on several factors such as: time (e.g., the speed at which the impact of a target affects operations), quality (e.g., the level of damage caused to output, production or service), surrogate (e.g., effect in the output, production or service), relativity (e.g., number of targets, position, relative value).
- **Accessibility (A)** refers to the ability and means to communicate or interact with a system, use system resources to handle information, gain knowledge of the information the system contains, or control system components and functions [Kis11].
- **Recuperability (R)** measures the time that a target needs to replace, repair, or bypass destruction or damage. Recuperability varies with the available sources and type of targeted components.
- **Vulnerability (V)** is a weakness in an information system, system security procedures, internal controls, or implementation that can be exploited or triggered by a threat source [Kis11]. A target is vulnerable if the operational element has the means and expertise to successfully attack the target.
- **Effect (E)** measures all significant impact (whether desired or not), at the target and beyond, that may result once the selected target is attacked.
- **Recognizability (R)** is the degree to which a target can be recognized by an operational element. Factors such as the size and complexity of the target, the existence of distinctive signatures, the presence of masking or camouflage influence the level of recognizability of a given target.

The methodology assigns numerical values on a scale of 1 to 10 to each considered factor and places them in a decision matrix. The sum of the values indicate the severity of a given dimension. This methodology is used in Section 6.2.3 to assign a weight to each axis, and each element of the axis composing the coordinate system.

6.1.3 Uniform Resource Identifier (URI)

Today, the universal manner in which a computing resource is accessed is by using a Uniform Resource Identifier (URI). A URI is a compact sequence of characters that identifies an abstract or physical resource [JW04, BLFM05]. URIs constitute an agreement about how the Internet community allocates names and associates them with the resources they identify. URIs are characterized as follows:

- **Uniform:** by allowing different types of resource identifiers to be used in the same context, as well as uniform semantic interpretation of common syntactic conventions across different types of resource identifiers.
- **Resource:** by allowing abstract concepts (e.g., operators and operands of a mathematical equation, types of relationships, or numeric values such as zero, one, infinity); as well as

concrete concepts (e.g., electronic documents, images, services) to be accessible.

- **Identifier:** by giving a unique identification (e.g., name, address, context) to distinguish a resource among others

Considering that the information provided in a URI can be used to identify an attack, it is therefore an essential element in the attack and risk evaluation. The generic URI syntax consists of a hierarchical sequence of components referred to as scheme, authority, path, query, and fragment [BLFM05], as shown in Listing 6.1. Appendix C provides more details on each URI component.

Listing 6.1 - URI Generic Syntax

```
e.g., foo://exemple.com:8042/over/there?name=ferret#nose
   [scheme][authority]        [path]      [query]       [fragment]
```

6.1.4 OrBAC Model for Countermeasures

The Organization Based Access Control (OrBAC) is a model designed to specify security policies at the organizational level. In our approach, the OrBAC model defines the countermeasures to be implemented. OrBAC uses abstract entities (Role, Activity, View) instead of concrete entities (subject, action, object) to reason on the roles that subjects, actions and objects play in an organization. These entities are defined by Abou et al., [KBB+03] and Miege [Mie05], as shown in Table 6.1.

Table 6.1 - OrBAC Entities

Concrete	Abstract
Subjects: model either active entities (e.g. user Alice, and Bob) or organizations	**Roles:** refer to groups of subjects with the same permissions (e.g. administrators, internal users, and external users)
Actions: model mainly computer actions (e.g. access, read, and write)	**Activities:** regroup actions that share the same principles (e.g. consulting, transferring, and paying bills)
Objects: model inactive entities (e.g. logs, and account information.)	**Views:** regroup objects that possess the same properties (e.g. Log server, Account Management Server)

The main objective of this model is to allow organizations to define their own security policies. OrBAC links abstract entities (e.g. roles, activities, views) and concrete entities (e.g. subjects, actions and objects) throughout relationships (e.g. empower, consider, use). In this manner, for a given organization, subject 's' is empowered in the role 'r'; action 'α' is considered in activity 'a'; and object 'o' is used in view 'v'.

OrBAC policies can be seen as a two-level security policy: on the one hand, abstract authorizations (e.g. Permission, Prohibition, Obligation) are granted to organizational entities (e.g. Organizations, Roles, Activities, Views, Contexts), and on the other hand, concrete authorizations (e.g. Is_permitted, Is_ prohibited, Is_Obliged) are granted to concrete entities (e.g. Subjects,

Actions, Objects). In addition, as described by Miege [Mie05], and Cuppens et al. [CCBM04] it is possible to consider hierarchies in organizations, roles, activities and views, which allows the inheritance of Permissions, Prohibitions and Obligations associated with these hierarchies. These policies are defined as predicates in the OrBAC model.

Equation 6.2 gives an example of an abstract security policy, and Equation 6.3 presents an example of a concrete security policy expressed in the OrBAC model.

$$Permission(org, r, a, v, c) \tag{6.2}$$

Meaning that organization 'org' grants role 'r' the positive authorization to perform activity 'a' on view 'v' in context 'c'.

$$Is\_permitted(s, \alpha, o) \tag{6.3}$$

Meaning that subject 's' is permitted to perform a concrete action 'α' on object 'o'.

Miege [Mie05], and Cuppens et al. [CCB07] propose the definition of dynamic policies through OrBAC contexts, which specifies conditions that must be satisfied in order to activate a security policy. Each context is seen as a ternary relation between subjects, actions, and objects defined within an organization. This derives the hold predicate shown in Equation 6.4.

$$Hold(org, s, \alpha, o, c) \tag{6.4}$$

Meaning that in an organization org, subject 's' performs an action 'α' over an object 'o' within a particular context 'c'. Different contexts have been defined as: temporal (related to the time at which a subject requests to access the system), spatial (related to the subject location), user-declared (which depends on the subject purpose), prerequisite (which depends on the characteristics that join a subject, an action and an object) and provisional (related to previous actions performed by the subject on the system) to model security policies. In our approach, contexts are used to activate countermeasures.

6.2 Coordinate System

Similar to the Cartesian Coordinate System, we propose a coordinate system composed of three dimensions with the following characteristics:

- The system is composed of three orthogonal axes, any two of them being perpendicular,
- There exists a single unit of length for all three axes,
- There exists a single orientation for each axis.

The three axes are bounded by the size of the system. The volume encompassed by the three axes represents the risk at which the system is exposed, and corresponds to the maximum attack volume. Inside this volume, we find sub-volumes that correspond to the attacks and/or countermeasures applied on the system.

It is important to mention that we chose to model the coordinate system using 3 axes due to the fact that we obtain from a URI the 3 main components of an information system (i.e. subject, object, and action). However, the number of axes in our coordinate system may change. We have a flexible system that is modeled in three dimensions (i.e. user account, resource, and channel), similar to the OrBAC model, but it can be adapted for two, four or more dimensions.

The remaining of this section presents the volume definition, as well as the dimensions composing our coordinate system, and the assigned weight to each dimension.

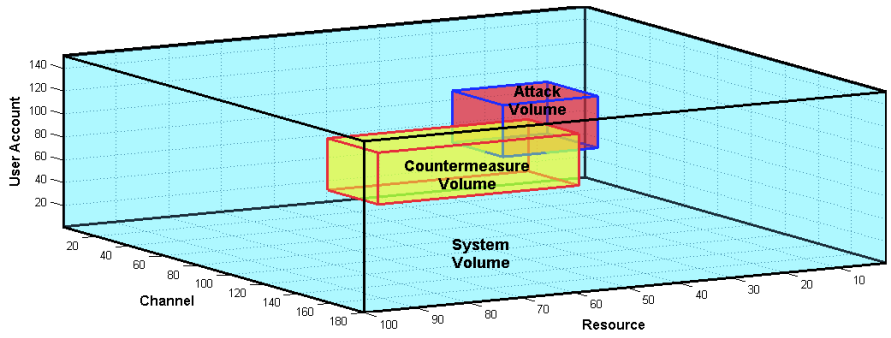

FIGURE 6.1 - Volume Graphical Representation

6.2.1 Volume Definition

A volume is the quantity of three-dimensional space enclosed by some closed boundary [Har10]. We study 3 types of volumes: the system volume (the maximal space susceptible to be attacked), the attack volume (part of the system volume that is compromised), and the countermeasure volume (part of the system volume that is protected by a given countermeasure). For each element we define the system dimensions (discussed in Section 6.2.2), and we assign a weighting factor depending on the contribution of each axis to the calculation of the volume. This weighting factor corresponds to the criticality of each element represented on the axis, as well as, the relative criticality of an axis to another. Figure 6.1 depicts the graphical representation of each type of volume.

6.2.1.1 System Volume (SV):

It represents the maximal space a given system (e.g. S1) is exposed to attackers. This volume includes tangible assets (e.g., PCs, mobile phones, network components, etc.), as well as intangible assets (e.g., confidential information, business reputation, etc) that are vulnerable to known and unknown threats. Each of these assets are represented in the system volume as user accounts, channels, and/or resources.

6.2.1.2 Attack Volume (AV):

Within the complete system volume exposed to attackers (including all possible vulnerable resources of the given system), we concentrate on a given attack to identify the portion of the volume being targeted based on the vulnerabilities it can exploit. These vulnerabilities are related to all the dimensions that comprise the system volume (i.e. user accounts, channels, and resources).

Vulnerabilities are generally mistakes made by programmers when writing software (e.g. typos, math errors, incomplete logic or incorrect use of functions or commands). These vulnerabilities can be used directly by hackers to access protected data, which in turns allows them to modify information, use the system to their own advantage, shut-down services, or publicly access systems without the organization's knowledge [Mar01].

6.2.1.3 Countermeasure Volume (CV):

The countermeasure volume represents the level of action that a security solution has on a given system. In other words, the countermeasure volume is the percentage of the system volume that is covered and controlled by a given countermeasure. An attack is covered by a countermeasure if their volumes overlap. The countermeasure can exceed the attack volume and cover part of the system that is not covered by the attack. Two cases are distinguished:

- Total Coverage, where all exploitable vulnerabilities associated to a given attack A_1 are controlled by a given countermeasure (e.g. C_1), in this case we have a perfect mitigation (100% of the attack volume coverage);
- Partial Coverage, where only a portion of the exploitable vulnerabilities associated to a given attack A_1 is controlled by a given countermeasure (e.g. C_1).

6.2.2 System Dimensions

In analogy with the OrBAC model, we identified 3 main dimensions that contribute directly to the execution of a given attack: User account (subject), Resource (object), and Channel (the way to execute actions, e.g. connect, read, write, etc). This latter is represented as the transitions between subjects and objects. For instance, in order to access a web-server (object) of a given organization, a user (subject) connects to the system by providing his/her login and password (action).

From the URI perspective, we consider the user information as the User account dimension, the host and port information as the Channel dimension, and the path, query and fragment elements as the Resource dimension. It is important to note that the URI scheme is not considered in the definition of the attack dimensions, since it does not provide valuable information to identify an attack. The remaining of this section details each dimension.

6.2.2.1 User Account:

A user account is a unique identifier for a user in a given system that allows him/her to connect and interact with the system's environment. A user account is generally defined by a user name or login name associated with a password. A user account is associated to a given status in the system, from which his/her privileges and rights are derived (i.e. super administrator, system administrator, standard user, guest, internal user, or nobody).

- Super Administrator: Also called "root user", is the account with the highest level of access within the system (i.e. permission to view and modify all fields on the system). There is only 1 super administrator in the system.
- System Administrator: It is created by the root user and it is able to view and modify most fields of the system. There are as many systems administrators as required.
- Standard User: It is a limited user account that is granted the right to use most software and system settings that do not affect other users. There are as many standard users as required.
- Guest: This is a user account with a temporary access to the system. Guest accounts have the same access as standard users but it is further restricted by not being able to install software, hardware or change settings. There are as many guests as required.
- Internal User: It is used for a person who has a member status and uses this account to access and interact with the system. An internal user has restricted rights (e.g. it cannot access the administrator interface, but it can perform simple edit operations on the interface). There are as many local users as required.

- Nobody: It is a user account that owns no files, is in no privileged groups, and has no abilities except those which every other user has.

Table 6.2 presents the different categories of user accounts according to their associated rights and privileges.

TABLE 6.2 - User Account Categories

User Account	Rights and Privileges			
	Read	**Write**	**Modify**	**Admin Access**
Super Admin	all	all	all	yes
System Admin	all	most	most	yes
Standard User	all	some	some	no
Guest	all	few	few	no
Internal User	all	few	few	no
Nobody	all	none	none	no

6.2.2.2 Channel:

In order to have access to a particular resource, a user must use a given channel. This section considers the IP address and the port number to represent channels in TCP/IP connections. However, each organization must define the way its users connect to the system and have access to the organization's resources.

6.2.2.2.1 IP Address: The Internet Protocol (IP) address is a unique numerical label assigned to each device on a network (e.g. PC, printer). The IP address offers two main functions: (i) Identification of the host or network interface, and (ii) Location addressing [oSC80]. There are two versions of the IP addresses: IPv4, which defines an IP address as a 32-bit unit, limiting the address space to 4,294,976,296 (2^{32}) possible unique addresses [Tou13]; and IPv6, which defines an IP address as a 128-bit unit, limiting the space to a maximum of 3.403×10^{38} (2^{128}) unique addresses [DH95].

IP address can be either public, private, or reserved for special purposes [CVBH13]. Public IP addresses are those used when communicating with or connecting to the Internet. These addresses are designated by the Internet Assigned Numbers Authority (IANA) [CET+11] for use in web servers, e-mail servers, firewalls and other devices that are directly connected to the Internet. Private IP addresses are those assigned to a device on a private TCP/IP Local Area Network (LAN) that is accessible only within the LAN. For a resource inside the LAN to be accessible over the Internet, a device within the LAN must be connected to the Internet with a public IP address, and the network must be appropriately configured. Special purpose IP address are those reserved for particular purposes e.g. tests, loopback, broadcast, or just reserved for further allocation.

Examples of public, private and reserved IP address are depicted in Table 6.3.

6.2.2.2.2 Port Number: A port is an application-specific or process-specific software construct, serving as a communication end-point in a computer's host operating system.

The purpose of a port is to identify different applications or processes running on a single computer and thereby enable them to share a single physical connection to a packet switch network like the Internet. Port numbers are divided into 3 ranges: The well-known ports, the registered ports and the dynamic ports [CET+11]. Well-known ports are those from 0 through 1023 (e.g. 20: File Transfer Protocol, 25: Simple Name System Protocol, 80: HyperText Transfer Protocol, 443:

TABLE 6.3 - IP address Categories

IP Address Types	Characteristics	Examples
Public	any address or number assigned to a device accessible over the Internet	1.0.0.0/8, 9.0.0.0/8, 129.0.0.0/8
Private	any address or number assigned to a device accessible only within a LAN	10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16
Reserved/ Special purpose	any address or number reserved for a specific purpose e.g. loop-back, broadcast, etc	127.0.0.0/8, 255.255.255.255/32, 240.0.0.0/4

TABLE 6.4 - Port Number Categories

Class	Characteristics	Port Numbers
1	Well-known and widely used	20, 21, 22, 23, 25, 53, 80, 110, 119, 143, 161, 443
2	Well-known and not widely used	From 0 to 1023 except ports from Class 1
3	Registered official ports that are used by multiple applications	1109, 1200, 1337, 1521, 1550, 1761, 2049, 2082, 2083, 2086, 2105, 2210, 2211, 2212, 2399, 2809, 4662, 5000, 5001, 5150, 5228, 5281, 6005, 6100, 6112, 6888, 6969, 7787, 7788, 7937, 8000, 8008, 8080, 8880, 8887, 8888, 9001, 9080, 9800, 9898, 15000, 20000, 26000
4	Registered official ports and not widely used	From 1024 to 49151 except ports from Class 3
5	Private ports	From 49152 to 65535

HTTP Secure). Registered ports are those from 1024 through 49151, and dynamic or private ports are those from 49152 through 65535. Table 6.4 presents the different categories of port numbers according to their use by applications [TKL+13].

6.2.2.3 Resource:

A resource is either a physical component (e.g. host, server, printer) or a logical component (e.g. files, records, database) of limited availability within a computer system. We identify 3 elements from the URI generic syntax (i.e. path, query, and fragment) that can be exploited to have access to a given resource. It is important to recall that the path section of a URI contains data organized in hierarchical form, that along with the non-hierarchical query component, serves to identify a resource. In addition, the query and fragment sections of a given URI allow indirect identification of a secondary resource [BLFM05].

We defined 2 levels of privileges (i.e. root, user), and 7 level of transitions (i.e. read, write, execute, and their combinations), and we assigned numerical values to each privilege and transitions based on their characteristics. Table 6.5 summarizes these values.

TABLE 6.5 - Privilege and Access Right Values for Resources

Privilege	Characteristic	Value	Transition	Characteristic	Value
Kernel	It grants complete access to all files and commands, including the system's kernel.	3	R	Read file names without additional information (e.g. content, size, type)	9
User	It allows users to run a limited number of applications in user mode.	1	W	Modify a file or entries in a directory (e.g. create, delete, rename)	6
			X	Execute files or scripts, access file contents and meta-info.	6
			R-W	Read and write	4
			R-X	Read and execute	4
			W-X	Write and execute	4
			R-W-X	Read, write and execute	3

6.2.3 Unit Volume Construction

As previously stated, a bijection between the URIs and the coordinate system is required in order to make the appropriate transformations. A bijection is a function giving an exact pairing of the elements of two sets [Ger07]. To have an exact pairing between X and Y (where Y needs to be different from X), four conditions must hold:

1. each element of X must be paired with at least one element of Y,
2. no element of X may be paired with more than one element of Y,
3. each element of Y must be paired with at least one element of X,
4. no element of Y may be paired with more than one element of X.

In formal mathematical terms, the function f: $X \rightarrow Y$ is bijective iff for all $y \in Y$ there is a unique $x \in X$ such that $f(x)=y$.

Since the three axes in our coordinate system are independent, the bijection from a URI to the coordinate system implies three bijections (one for each axis). It is important to note that a physical object can be accessed in many different ways in our system, and the bijective function must take into account all these ways. For instance, let us suppose that in a given system we have a DMZ and a private network with a firewall in between. Depending upon the URI used to access a machine in the system, the IP address will change. If the machine is seen behind the firewall, we will see a private IP address, if the same machine is seen in front of the firewall, we will see a public IP address, even though it is the same object. We will have, therefore, a bijection in the objects but not in their IP addresses. In such a case, we can decide that the two machines correspond to the same object with the same risk, or we can define the two machines as two separate objects, each one with a different risk level.

In those cases where the information of one axis is missing, we propose the following approaches:

a.- The optimistic approach, which suggests working in a 2-dimensional system by eliminating the axis that does not provide any information. In this case we will work with surfaces instead of volumes but we will keep the same geometry.

b.- The pessimistic approach, which suggest using the whole axis, making the assumption that the value of the axis is the same for all the possible cases.

Each axis contributes differently in the volume calculation. This contribution represents the criticality of a given element in the execution of an attack. Following the CARVER methodology (introduced in Section 6.1.2), we assign a weighting factor to each dimension (inter-dimension normalization), as well as to each category within the dimension (intra-dimension normalization). The remaining of the section details this methodology.

6.2.3.1 Inter-dimension Normalization:

Each attack dimension represents a portion of the total volume. Table 6.6 presents the CARVER estimation of each dimension and its corresponding weighting factor.

TABLE 6.6 - Attack Dimensions Weight

Attack Dimension	C	A	R	V	E	R	Total	%	Weight Factor
User Account	8	7	9	7	8	7	46	40%	2
Channel	5	6	5	6	5	4	31	28%	1
Resource	7	6	6	5	7	5	36	32%	1.5

6.2.3.2 Intra-dimension Normalization:

Each category within the axis contributes differently to the volume calculation. The weighting factor corresponds to the number of units that represent an instance in a given category. The remaining of this section details the weighting factor assigned to each category of the attack dimension.

6.2.3.2.1 User Account: We have previously defined six categories of user accounts (i.e. super administrator, system administrator, standard user, guest, internal user, and nobody). Each user account category has an associated weighting factor that corresponds to the CARVER analysis. Table 6.7 summarizes this information.

In an educational institution, for instance, the president, vice-president, and dean are assigned 'super administrator' accounts, which represents an access to 75-100% of the system's resources, and a weighting factor equal to 4. Department directors, coordinators and administrative staff are assigned 'system administrator' accounts, which represents an access to 50-75% of the system's resources, and a weighting factor of 3. Professors, student assistants, PhD students, and contracted staff are assigned 'standard user' accounts, which represents an access to 25-50% of the system's resources, and a weighing factor of 2. Invited professors are assigned 'guest' accounts; and registered students are assigned 'internal user' accounts, both accounts represent an access to 0-25% of the system's resources, and a weighting factor of 1. All external users are assigned by default the

TABLE 6.7 - User Account Weight

User Account	Resource Access	C	A	R	V	E	R	Total	%	Weight Factor
Super Admin	100%	10	7	8	10	9	5	49	28	4
System Admin	75%	8	7	7	9	8	5	44	25	3
Standard User	50%	5	6	6	7	7	4	35	20	2
Guest	25%	3	3	3	5	4	3	21	12	1
Internal User	25%	3	3	3	5	4	3	21	12	1
Nobody	0%	1	1	1	1	1	1	6	3	0

'nobody' user account, since they have access to none of the system's resources. The weighting factor to the nobody account is 0.

Figure 6.2 shows the graphical representation of the weighting values for the User Account dimension.

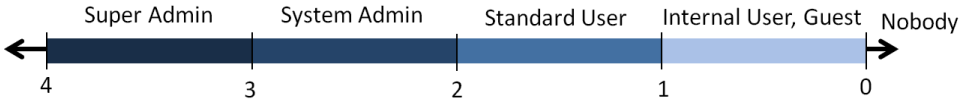

FIGURE 6.2 - User Account Weight Scale

6.2.3.2.2 Channels: We have previously defined the most commonly used channels in TCP/IP connections (i.e. IP address and port numbers). This section presents the normalization of these channels according to their contribution in the volume measurement.

IP Address: We assigned a weighting factor that ranges from 0 to 3, to each category of IP address. Table 6.8 summarizes this information.

TABLE 6.8 - IP Address Weight

IP address	C	A	R	V	E	R	Total	%	Weight Factor
Public	10	9	7	9	8	7	50	60	3
Private	5	1	5	3	5	3	22	27	1
Reserved/ Special purpose	3	3	3	1	3	1	11	13	0

Figure 6.3 shows the graphical representation of the weighting values for the IP Address element.

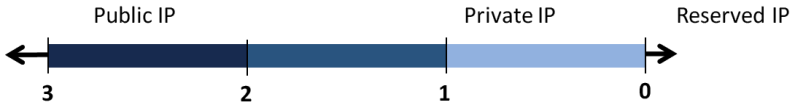

FIGURE 6.3 - IP Address Weight Scale

As depicted in Figure 6.3, public IP addresses are more likely to be used in the execution of an attack, representing a weighting factor of 3 units. Private IP addresses represent a weighting factor of 1 unit, and reserved/special purpose IP address are not very likely to be used in the execution of an attack, representing a weighting factor of 0 units.

Port Number: We assigned a weighting factor that ranges from 0 to 2,5 to each of the 5 previously defined categories of port numbers. Table 6.9 summarizes this information.

TABLE 6.9 - Port Number Weight

Port Number	C	A	R	V	E	R	Total	%	Weight Factor
Class 1	10	9	8	10	7	8	52	31	3
Class 2	8	7	8	5	5	8	41	25	2
Class 3	7	8	5	9	5	7	41	25	2
Class 4	3	2	3	4	3	5	20	12	1
Class 5	2	1	3	3	1	2	12	7	0

Figure 6.4 shows the graphical representation of the weighting values for the port number element.

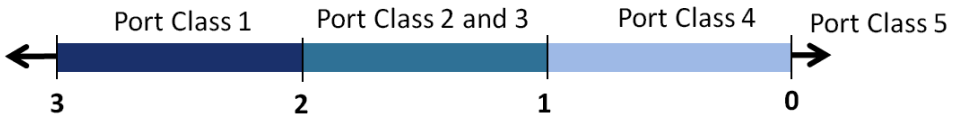

FIGURE 6.4 - Port Number Weight Scale

As depicted in Figure 6.4, Class 1 ports are more likely to be used in the execution of an attack, representing a weighting factor of 3 units. Class 2 and 3 ports are less likely to be used in the execution of an attack, representing a weighting factor of 2 units. Class 4 ports represent a weighting factor of 1 unit, and Class 5 Ports are not very likely to be used in the execution of an attack, representing a weighting factor of 0 units.

The resulting IP-Port couple is then represented as the sequence of affected IP address, followed by the active port numbers in ascending order (e.g. IP_1, IP_2,..., IP_n, $Port_1$, $Port_2$,..., $Port_n$).

6.2.3.2.3 Resource: Resources are used to attack a given system only if the attacker has the appropriate access rights and privileges. However, in order to acquire the required permissions, attackers must spend some effort. We assigned a weight to each resource based on the effort to obtain the access rights and privileges associated to a given resource.

The index that results from the division between the Privilege (PR) value and the Transition (TR) value (i.e. $\frac{PR}{TR}$) represents the level of access assigned to a given resource on the system. Table 6.10 summarizes this information and shows the weighting factor assigned to each resource dimension type.

Figure 6.5 shows the graphical representation of the weighting values for the resource dimension.

As shown in Figure 6.5, a compromised resource with a kernel privilege and Read-Write-Execute transition is assigned a weight of 5 units; a kernel privilege and Read-Write, Read-Execute, or

TABLE 6.10 - Resource Weight

Resource Type	Access	C	A	R	V	E	R	Total	%	Weight Factor
Kernel & R-W-X	100	10	10	9	9	9	8	55	23	5
Kernel & W-X/R-X/R-W	75	8	9	9	9	7	8	50	21	4
Kernel & W/X	50	6	7	7	8	7	5	40	17	3
Kernel & R / User & R-W-X	33	5	5	6	7	6	5	34	14	2
User & W-X/R-X/R-W	25	5	5	6	5	4	5	30	13	2
User & W/X	17	3	3	5	3	2	3	19	8	1
User & R	11	1	2	2	1	1	3	10	4	0

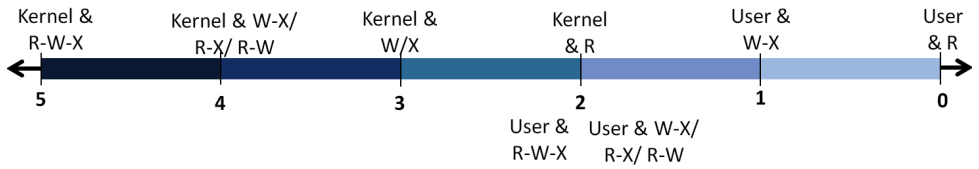

FIGURE 6.5 - Resource Weight Scale

Write-Execute transition is assigned a weight of 4 units; a kernel privilege with Write or Execute transition is assigned a weight of 3 units; and a kernel privilege with a Read-only transition is assigned a weight of 2 units. Similarly, a compromised resource with a user privilege and Read-Write-Execute transition is assigned a weight of 2 units; a user privilege with Read-Write, Read-Execute, or Write-Execute transition is assigned a weight of 2 units; a user privilege with Write or Execute transition is assigned a weight of 1 unit; and a user privilege with a Read-only transition is assigned a weight of 0 units.

6.3 Axis Contribution in the Volume Calculation

The contribution of a given axis (Co_{Ax}) in the volume calculation is determined as the union of all its elements 'E' that belongs to a given class 'c' from the system 'S' times its corresponding weighting factor 'WF', as proposed in Definition 1:

Definition 1: Axis contribution in the volume calculation
$Co_{Ax}(S) = \sum_{i=0}^{n} Count(E \in c(S)) \times WF_c$

6.3.1 User Account:

The contribution of the user account dimension (Co_{Acc}) in the calculation of the volume of system S_1 is determined as the sum of all active user accounts times their corresponding weighting factor, as shown in Definition 1.1:

Definition 1.1: User Account Contribution in the Volume Calculation
$Co_{Acc}(S_1) = (Count(E \in admin*(S_1)) \times WF(admin*)) + (Count(E \in admin(S_1)) \times WF(admin)) + (Count(E \in std\_user(S_1)) \times WF(std\_user)) + (Count(E \in int\_user(S_1)) \times WF(int\_user)) + (Count(E \in guest\ (S_1)) \times WF(guest))$

Given a system S_1, all of the system's resources 'r' are modified by the super administrator 'admin*', most of 'r' are modified by the system administrator 'admin', some of 'r' are modified by the standard user 'std_user', few of 'r' are modified by internal users 'int_user' and/or guest users 'guest', and none of 'r' is modified by external users 'nobody'; thus a compromised 'admin*' represents an access to 100% of 'r', a compromised 'admin' represents an access to 75% of 'r', a compromised 'std_user' represents an access to 50% of 'r', a compromised 'int_user' or 'guest' represents an access to 25% of 'r',and a compromised 'nobody' represents an access to 0% of 'r'.

6.3.2 Channel:

The contribution of the channel dimension ($Co_{Ip-Port}$) in the calculation of the volume of system S_1 is determined as the sum of the contributions of the IP address and the Port number, as shown in Definition 1.2:

Definition 1.2: Channel contribution in the volume calculation
$Co_{Ip-Port}(S_1) = Co_{Ip}(S_1) + Co_{Port}(S_1)$

The remaining of this section defines the calculation of each channel element.

6.3.2.1 IP Address:

The contribution of the IP address (Co_{Ip}) in the calculation of the volume of system S_1 is determined as the sum of all IP addresses composing system S_1 times their corresponding weighting factor, as shown in Definition 1.2.1:

Definition 1.2.1: IP contribution in the volume calculation
$Co_{Ip}(S_1) = Count(E \in Public\_IP(S_1)) \times WF(Public\_IP) + Count(E \in Private\_IP\ (S_1)) \times WF(Private\_IP)$

6.3.2.2 Port Number:

The contribution of the Port number (Co_{Port}) in the calculation of the volume of system S_1 is determined as the sum of all open ports composing system S_1 times their corresponding weighting factor, as shown in Definition 1.2.2:

Definition 1.2.2: Port contribution in the volume calculation
$Co_{Port}(S_1) = (Count(E \in class1(S_1)) \times WF(class1)) + (Count(E \in class2(S_1)) \times WF(class2)) + (Count(E \in class3(S_1)) \times WF(class3)) + (Count(E \in class4(S_1)) \times WF(class4))$

Given a system S_1, well known and widely used ports are represented by 'Class 1', well-known but not widely used ports are represented by 'Class 2', registered official ports that are used by multiple applications are represented by 'Class 3', registered official ports and not widely used are represented by 'Class 4', and private ports are represented by 'Class 5'; thus a compromised 'Class 1' port results into an access of 100% of resources, a compromised 'Class 2' port results into an access of 75% of resources, a compromised 'Class 3' port results into an access of 50% of resources, a compromised 'Class 4' port results into an access of 25% of resources, and a compromised 'Class 5' port results into an access of 0% of 'r'.

6.3.3 Resource:

The contribution of the resource dimension (Co_{Res}) in the calculation of the volume of system S_1 represents the sum of all system's resources times its corresponding weighting factor, as shown in Definition 1.3.

Definition 1.3: Resource contribution in the volume calculation
$Co_{Res}(S_1) = \sum_{i=0}^{n} Count(E \in Res_i(S_1)) \times WF(i)$

It is important to note from Definition 1.3, that the contribution of the resource dimension in the measurement of the surface volume requires the sum of all similar resources (i.e. resources with equal privileges and transitions) in order to assign a corresponding weighting factor. This latter is given in Table 6.10.

6.4 Volume Calculation

The projection of the three axis in our coordinate system generates a parallelepiped in three dimensions. The volume of a parallelepiped is the product of the area of its base 'A' and its height 'h'. The base is any of the six faces of the geometric figure, whereas the height is the perpendicular distance between the base and the opposite face. In Figure 6.6, for instance, the base of the parallelepiped is formed by the Channel (Ip-Port) and the Resource (Res) dimensions; and the height is formed by the user account (Acc) dimension.

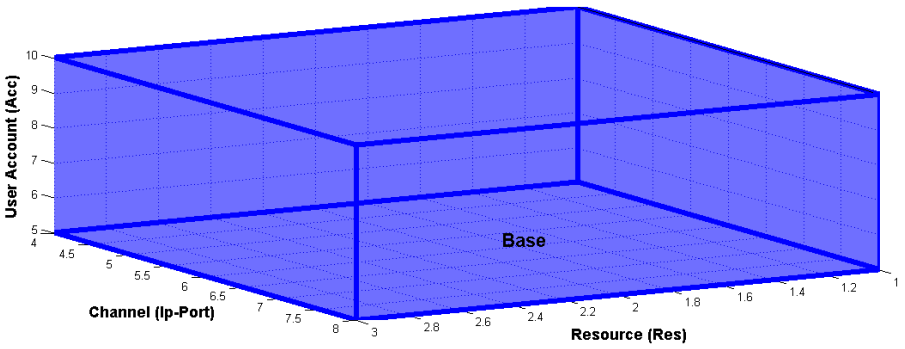

FIGURE 6.6 - Volume Representation

An alternative method to calculate the volume of a parallelepiped is performed by the scalar product of its vectors. For instance, having vectors a=(a_1, a_2, a_3), b=(b_1, b_2, b_3), and c=(c_1, c_2, c_3) to represent three edges that meet at one vertex, the volume then equals the absolute value of the scalar triple product (i.e. V=|a × (b × c)| = |b × (c × a)| = |c × (a × b)|.

The remaining of this section details the calculation of the different volumes defined in Section 6.2.1.

6.4.1 System Volume (SV) Calculation:

Consider a system S_1, which is a vector composed of three elements: a set of user accounts (Acc), a set of IP address and open ports (Ip-Port), and the system's resource (Res). The volume of system S_1 is represented by the vector SV(S_1)= ($Co_{Acc}(S_1)$, $Co_{Ip-Port}(S_1)$, $Co_{Res}(S_1)$). The system volume is calculated as the product of the axis contribution and its corresponding weighting factor, as shown in Definition 2.

Definition 2: System volume calculation
$SV(S_1) = Co_{Acc}(S_1) \times$ *WF(Acc)* $\times Co_{Ip-Port}(S_1) \times$ *WF(Ip-Port)* $\times Co_{Res}(S_1) \times$ *WF(Res)*

6.4.2 Attack Volume (AV) Calculation:

Consider A_1 as a given attack, $SV_{Acc}(A_1)$ as the A_1'user account-based volume, $SV_{Ip-Port}(A_1)$ as the A_1'Channel-based volume, and $SV_{Res}(A_1)$ as the A_1'resource-based volume. The volume of attack A_1 is represented by the vector: $AV(A_1) = (Co_{Acc}(A_1), Co_{Ip-Port}(A_1), Co_{Res}(A_1))$. The attack volume is calculated as the product of the axis contribution and its corresponding weighting factor, as shown in Definition 3.

Definition 3: Attack volume calculation
$AV(A_1) = Co_{Acc}(A_1) \times$ *WF(Acc)* $\times Co_{Ip-Port}(A_1) \times$ *WF(Ip-Port)* $\times Co_{Res}(A_1) \times$ *WF(Res)*

The coverage (Cov) of a given attack A_1 in the system S_1 is computed as the ratio between the attack volume (AV(A_1)) and the the system volume (SV(S_1)), as shown in Definition 3.1:

Definition 3.1: System-based attack coverage
$Cov(A_1/S_1) = \frac{AV(A_1 \cap S_1)}{SV(S_1)} \times 100$

6.4.3 Countermeasure Volume (CV) Calculation:

Consider a given countermeasure C_1, a set of user accounts as the attack vector 'Acc', a set of IP address and ports as the attack vector 'Ip-Port', and the system's resource as the attack vector 'Res'. The volume of countermeasure C_1 is represented by the vector: $CV(C_1) = (Co_{Acc}(C_1), Co_{Ip-Port}(C_1), Co_{Res}(C_1))$. The countermeasure volume is calculated as the product of the axis contribution and its corresponding weighting factor, as shown in Definition 3.

Definition 4: Countermeasure volume calculation
$CV(C_1) = Co_{Acc}(C_1) \times$ *WF(Acc)* $\times\ Co_{Ip-Port}(C_1) \times$ *WF(Ip-Port)* $\times\ Co_{Res}(C_1) \times$ *WF(Res)*

The coverage (Cov) of a given countermeasure (C_1) respect to a given attack (e.g. A_1) is calculated as the ratio between the countermeasure volume overlapping with the attack volume ($\mathrm{CV}(C_1 \cap A_1)$) and the attack volume ($\mathrm{AV}(A_1)$), as shown in Definition 4.1:

Definition 4.1: Attack-based countermeasure coverage
$Cov(C_1/A_1) = \frac{CV(C_1 \cap A_1)}{AV(A_1)} \times 100$

From Definition 4.1, the higher the ratio, the greater the mitigation level.

6.5 Conclusion

In this chapter we introduced the attack volume as an improvement of the attack surface model proposed by Howard et al. [How04, HW07] and Manadhata et al. [MWFM06, MKW08, Man08, MW10]. Based on the several limitations derived by the implementation of the attack surface model, we proposed an approach to model the information retrieved by a URI into a three-dimensional coordinate system (i.e. user, channel, and resource). All axes of the coordinate system are complementary, and relate to the OrBAC model, where the user account is modeled as a subject (role), with certain privileges; the channel is modeled as an action (with the required parameters) that carries on protocols to identify resources on the system; and the resource is modeled as an object (machine) with a path in the URI.

We propose a weighting factor, based on the CARVER methodology, to assign a weight on each axis according to six criteria (i.e. criticality, accessibility, recuperability, vulnerability, effect and recognizability). As a result, we are able to establish a direct connection between volumes and risks (e.g. big volume equals high risk). The figure that results from the projection of the 3 axis represents a parallelepiped, whose volume is then compared with other volumes on the system. Countermeasures are selected based on their coverage with respect to the attack volume.

In order to measure the volume of multiple attacks, a geometric approach is proposed in the next chapter, where the union and intersection of multiple volumes are calculated.

Chapter 7

Attack Volume: Geometric Approach

> Progress lies not in enhancing what is,
> but in advancing towards what will be.
>
> Khalil Gibran -

Contents

Innovation in Information Technology has brought numerous advancements but also some

consequences. Cyber attacks have evolved along with technology, reaching a state of high efficiency and performance. Distributed Denial of Service (DDoS), and Botnets, as well as Low and Slow Attacks are examples of this evolution. Attackers are becoming stronger and harder to detect, making the mitigation process a big challenge for security analysts. Part of the complexity is due to the fact that attacks are often divided into multiple stages, possibly distributed across multiple sources.

Most of the approaches to mitigate current attacks consider one attack at a time [Tho07, KCBCD10] by evaluating each step of the attack and proposing security solutions to either stop it or decrease its severity. However, very few effort is dedicated to study and analyze the effects of multiple attacks over a given target. Current research focuses on approaches to detect such attacks and demonstrate their robustness and the difficulty to mitigate them [AEG+10, FGV11, VF10, Fuj10]. Most of these works propose approaches to detect multiple attacks but few of them propose a methodology to react to these attacks.

This chapter therefore proposes an approach to react against multiple attacks based on the notion of the attack volume (detailed in Chapter 6). The rest of the chapter is structured as follows: Section 7.1 briefly introduces the state of the art in multiple attack scenarios, Section 7.2 describes the process to calculate the union and intersection of attack volumes. Section 7.3 discusses the countermeasure volume for multiple attacks. Section 7.4 deploys a case study to evaluate appropriate countermeasures in a scenario of multiple attacks. Finally, a discussion about multiple attack scenarios is given in Section 7.5.

7.1 State of the Art

7.1.1 Multiple Attacks

A computer attack refers to any kind of malicious activity that attempts to collect, disrupt, deny or destroy information system resources and/or the information itself [Kis11]. Multiple attacks can be executed over a single or multiple targets.

Most of the attacks on the cyber space consist of opportunistic attacks rather than those targeted to a specific entity. A single target attack originates when the attacker specifically concentrates his/her efforts in one person or a given organization. This kind of attacks are highly more effective and dangerous than others since the actions performed by the malicious entities are tailored, making it more difficult to detect and stop [Sof09].

A multiple target attack occurs when a malicious entity attacks various parties (regardless of who the victims are), by using one or various generic ways to attack such parties, hoping that some of them will be vulnerable to the attack. An attacker generally finds more interesting to deploy a multiple target attack than a single targeted one, since the former may provide a higher probability of success (with possibly less effort) in obtaining sensitive and valuable information than a single target attack, which requires to focus on a particular subject that might be better protected against such attacks.

7.1.2 Limitation of Current Solutions

Research in the detection and mitigation of multiple attacks is an open issue. Vetillard et al. [VF10] present a work that combines logical (software) and physical (hardware) attacks to build a successful attack path on a platform or application. Authors have shown and implemented an example of a combined attack that works on a basic implementation of Java Card. The combined

physical and logical attacks allow a wide range of different attacks, which can be implemented on platforms that were considered sufficiently safe.

Amiel et al. [AV07] and Clavier et al. [CFGR10] propose a new class of attack called Passive and Active Combined Attack (PACA) that is effective on defeating symmetric cryptography and a supposedly state of the art secure AES implementation. Results demonstrate that naively adding countermeasures together is not sufficient to mitigate a combined attack and that implementing these protections must be done carefully.

Similarly, authors in [DCMS06, MKBS07] explore Coordinated and Distributed Multiple Attacks (CDMA), a more damaging and highly synchronized Distributed Denial of Service Attack (DDoS). Authors determine that CDMAs target multiple vulnerabilities using diverse selection of protocols as well as varying the attacks over time. The resulting attack is more robust and difficult to mitigate as multiple attack vectors allow for a large target area on the victim which potentially gives the attack a greater probability of success.

Liu et al. [LSY11] propose a defense scheme for multiple-target attacks. The solution sets up heterogeneous thresholds for detecting suspicious items and identifies target items based on correlation analysis among suspicious items. As a result, the proposed scheme achieves interesting results in the detection of malicious users and has less impact on normal items that are not under attack.

To address the aforementioned limitations, we propose a geometric approach to calculate the union and intersection of multiple attacks, using the notion of attack volume from the previous chapter. Next section details the complete approach.

7.2 Attack Volume Union and Intersection

The calculation of the Attack Volume (AV) for multiple attacks requires the identification of their union and intersection. The union of two or more attack volumes is bounded and ranges from the maximum volume of the group of attacks in its lower bound, to the sum of the individual volumes in its upper bound. The intersection of two or more attack volumes ranges from zero in its lower bound, to the minimum volume of the group of attacks in its upper bound, as shown in the Definition 5.

Definition 5: Union and Intersection for multiple attacks
$AV(A_1 \cup ... \cup A_n) \in [max(AV(A_1), ..., AV(A_n)) - \sum AV(A_1), ..., AV(A_n)]$
$AV(A_1 \cap ... \cap A_n) \in [0 - min(AV(A_1), ..., AV(A_n))]$

Two cases can be distinguished in the calculation of the surface union and intersection: joint and disjoint attack surfaces.

7.2.1 Disjoint Attack Volumes

The volume of one attack is disjoint from another attack volume if they have no elements in common. Therefore, having n number of disjoint attacks $(A_1, ..., A_n)$, the surface volume of their union and intersection is calculated using Equations 7.1 and 7.2 respectively.

$$AV(A_1 \cup ... \cup A_n) = \sum_{i=1}^{n} AV(A_i) \tag{7.1}$$

$$AV(A_1 \cap ... \cap A_n) = 0 \tag{7.2}$$

From the previous equations, we derive the following definition:

Given two attacks (A_1, A_2), Attacks A_1 and A_2 are disjoint if their combined volume has no element in common, therefore, the attack volume of the union is calculated as the sum of their individual surfaces, and the attack volume of the intersection is equal to 0, as shown in Definition 5.1:

Definition 5.1: Union and Intersection for disjoint attacks
iff $A_1 \cap A_2 = \varnothing$ *then,* $AV(A_1 \cup A_2) = AV(A_1) + AV(A_2)$ *and* $AV(A_1 \cap A_2) = 0$

7.2.2 Joint Attack Volumes

The Volume of one attack is partially or totally covered by another attack if they share some or all of their elements. For n number of partially covered attacks (e.g., $A_1, ..., A_n$), the union is calculated as the sum of the individual attack volumes minus their intersections (Equation 7.3), and the intersection volume is calculated as the sum of the individual attack volumes minus their union (Equation 7.4).

$$\begin{aligned} AV(A_1 \cup ... \cup A_n) &= \textstyle\sum_{i=1}^{n} AV(A_i) - \sum_{i,j=1}^{n} \binom{n}{2} AV(A_i \cap A_j) \\ &\quad + \textstyle\sum_{i,j,k=1}^{n} \binom{n}{3} AV(A_i \cap A_j \cap A_k) + ... + \\ &\quad (-1)^{n+1} AV(A_i \cap ... \cap A_n) \end{aligned} \tag{7.3}$$

$$\begin{aligned} AV(A_1 \cap ... \cap A_n) &= \textstyle\sum_{i=1}^{n} AV(A_i) - \sum_{i,j=1}^{n} \binom{n}{2} AV(A_i \cup A_j) \\ &\quad + \textstyle\sum_{i,j,k=1}^{n} \binom{n}{3} AV(A_i \cup A_j \cup A_k) + ... + \\ &\quad (-1)^{n+1} AV(A_i \cup ... \cup A_n) \end{aligned} \tag{7.4}$$

From the previous equations, we derive the following definitions:

Given two attacks (A_1, A_2), Attacks A_1 and A_2 are joint if their combined volume has at least one element in common, therefore, the attack volume of the union is calculated as the sum of their individual volumes minus their intersection, and the attack volume of the intersection is calculated as the sum of their individual volumes minus their union, as shown in Definition 5.2.

Definition 5.2: Union and Intersection for joint attacks
iff $A_1 \cap A_2 \neq \varnothing$ *then,* $AV(A_1 \cup A_2) = AV(A_1) + AV(A_2) - AV(A_1 \cap A_2)$ *or*
$AV(A_1 \cap A_2) = AV(A_1) + AV(A_2) - AV(A_1 \cup A_2)$

Given two attacks (A_1, A_2), Attack A_1 is a subset of Attack A_2 if the volume of A_1 is a subset of the volume of A_2 ($\mathrm{AV}(A_1) \subseteq \mathrm{AV}(A_2)$), therefore, the attack volume of the union is equal to the attack volume of the bigger attack, and the attack volume of the intersection is equal to the attack volume of the smaller attack, as shown in Definition 5.3.

Definition 5.3: Union and Intersection for totally covered attacks
iff $A_1 \subseteq A_2$ *then,* $AV(A_1 \cup A_2) = AV(A_2)$ *and* $AV(A_1 \cap A_2) = AV(A_1)$

Given two attacks (A_1, A_2), Attacks A_1 and A_2 have the same volume if Attack A_1 is a subset of Attack A_2 and Attack A_2 is a subset of Attack A_1, therefore, the attack volumes of the union and the intersection are the same as their individual attack volume (Definition 5.4.)

Definition 5.4: Union and Intersection for equal volume attacks
iff $A_1 \subseteq A_2 \wedge A_2 \subseteq A_1$ *then,* $AV(A_1 \cup A_2) = AV(A_1 \cap A_2) = AV(A_1) = AV(A_2)$

Given two attack volumes as introduced in Definition 3 from Chapter 6 (e.g., $AV(A_1) = \langle Co_{Acc}(A_1), Co_{Ip-Port}(A_1), Co_{Res}(A_1)\rangle$; $AV(A_2) = \langle Co_{Acc}(A_2), Co_{Ip-Port}(A_2), Co_{Res}(A_2)\rangle$, the attack volume intersection is calculated as the sum of all elements 'E' that are included in both set of volumes times their corresponding weighting factor, as shown in Definition 5.5.

Definition 5.5: Attack volume intersection
$AV(A_1 \cap A_2) = Co_{Acc}(A_1 \cap A_2) \times$ *2* $\times Co_{Ip-Port}(A_1 \cap A_2) \times$ *1* $\times Co_{Res}(A_1 \cap A_2) \times 1,5.$

Where:

$Co_{Acc}(A_1 \cap A_2) = \sum_{i=0}^{n}(E_i \times WF(E_i) \mid E_i \in A_1 \wedge E_i \in A_2).$
$Co_{Ip-Port}(A_1 \cap A_2) = \sum_{i=0}^{n}(E_i \times WF(E_i) \mid E_i \in A_1 \wedge E_i \in A_2).$
$Co_{Res}(A_1 \cap A_2) = \sum_{i=0}^{n}(E_i \times WF(E_i) \mid E_i \in A_1 \wedge E_i \in A_2).$

From the previous equations, 'E' represents the elements associated to each attack dimension (i.e. IP address, channel, and resource) that are compromised during the execution of a given attack, and '$\mathrm{WF}(E_i)$' corresponds to the weighting factor of the element E_i as proposed in Section 6.2.3.1.

7.2.3 Dimension-based Attack Volume Calculation

The calculation of the attack volume union and intersection based on a given dimension derives the following definition:

Definition 6: Dimension-based union and intersection for multiple attacks
$Co_{Vec}(A_1 \cup A_2) = Co_{Vec}(A_1) + Co_{Vec}(A_2) - Co_{Vec}(A_1 \cap A_2)$
$Co_{Vec}(A_1 \cap A_2) = \sum_{E \in Vec_1 \cap Vec_2} WF(E).$

Given two attacks (A_1, A_2), a set of elements Vec_1=$\{E_1, E_2, ...E_n\}$ that are targeted by A_1 in this dimension, and a set of elements Vec_2=$\{E_a, E_b, ...E_x\}$ that are targeted by A_2, the contribution of the union to the volume is calculated as the sum of each individual volumes minus their intersection. The intersection of both attacks is calculated as the sum of the elements that belong to both dimensions (Vec_1, Vec_2) times their corresponding weighting factor.

The remaining of this section details the methodology to calculate the attack volume union and intersection for each attack dimension.

7.2.3.1 User Account:

Given two attacks (A_1, A_2), a set of user accounts UA_1=$\{Acc_1, Acc_2, ...Acc_n\}$ that are targeted by A_1, and a set of user accounts UA_2=$\{Acc_a, Acc_b, ...Acc_x\}$ that are targeted by A_2, the volume intersection of both attacks based on the user account dimension is calculated as the elements that belong to both set of user accounts (UA_1, UA_2) times their corresponding weighting factor, as shown in Definition 6.1.

Definition 6.1: User Account-based volume intersection
$Co_{Acc}(A_1 \cap A_2) = Count(admin* \in UA_1 \wedge UA_2) \times WF(admin*) + Count(admin \in UA_1 \wedge UA_2) \times WF(admin) + Count(std\_user \in UA_1 \wedge UA_2) \times WF(std\_user) + Count(int\_user \in UA_1 \wedge UA_2) \times WF(int\_user) + Count(guess \in UA_1 \wedge UA_2) \times WF(guess)$.

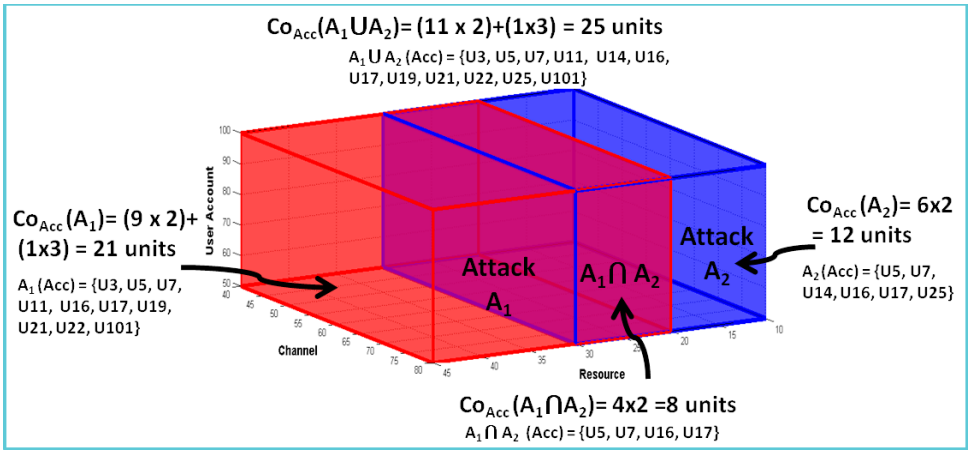

FIGURE 7.1 - User Account-based volume union and intersection

In Figure 7.1, Attack A_1 affects 9 standard users (i.e. U3, U5, U7, U11, U16, U17, U19, U21, U22) and 1 system admin account (i.e. U101), which represents a contribution on the attack volume of C_{Acc} (A_1) = 21 units (using the weighting factor from Table 6.7). Attack A_2 affects 6 standard user accounts (i.e. U5, U7, U14, U16, U17, U25), which represents a contribution on the attack volume of Co_{Acc} (A_2) = 12 units. The union of both attacks covers 11 standard user accounts and 1 system admin; the account volume contribution based on the union of A_1 and A_2 is therefore calculated as $Co_{Acc}(A_1 \cup A_2) = (11 \times 2) + (1 \times 3) = 25$ units. The intersection of attacks A_1 and A_2 covers 4 standard user accounts, the account volume contribution based on the intersection of both attacks is therefore calculated as $Co_{Acc}(A_1 \cap A_2) = (4 \times 2) = 8$ units.

7.2.3.2 Channel:

As presented in Definition 1.2, the contribution of the channel dimension ($Co_{Ip-Port}$) in the volume calculation is determined as the sum of the contributions of the IP address and the Port number. The remaining of this section defines the calculation of each channel element.

7.2.3.2.1 IP Address: Given two attacks (A_1, A_2), a set of IP address I_1=$\{IP_1, IP_2, ..., IP_n\}$ that are targeted by A_1, and a set of IP address I_2=$\{IP_a, IP_b, ...IP_x\}$ that are targeted by A_2, the volume intersection of both attacks based on the IP address is calculated as the elements that belong to both set of IP addresses (I_1, I_2) times their corresponding weighting factor. The contribution (Co) of the IP address is shown in Definition 6.2.

Definition 6.2: IP address-based volume intersection
$Co_{Ip}(A_1 \cap A_2) = Count(Public\_IP \in I_1 \wedge I_2) \times WF(Public\_IP) + Count(Private\_IP \in I_1 \wedge I_2) \times WF(Private\_IP)$.

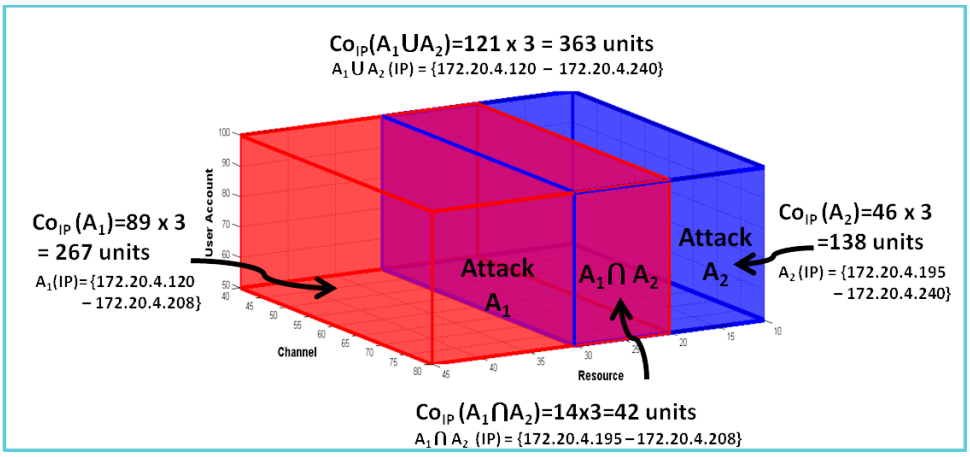

FIGURE 7.2 - IP address-based volume union and intersection

Figure 7.2 shows that having an attack 'A_1' that is produced in a network composed of the IP address set [172.20.4.120 - 172.20.4.208], and an attack 'A_2' that is realised in a network composed of the IP address set [172.20.4.195 - 172.20.4.240], we determine the IP address volume contribution based on A_1 as Co_{IP} (A_1) = 267 units, and the IP address volume contribution based on A_2 as $C_{IP}(A_2)$ = 138 units (using the weighting factor from Table 6.8). The intersection between attacks A_1 and A_2 results into the IP range [172.20.4.195 - 172.20.4.208], with an IP-based attack volume contribution of $Co_{IP}(A_1 \cap A_2)$ = 42 units; and the union of attacks A_1 and A_2 results into the IP range [172.20.4.120 - 172.20.4.240], with an IP-based attack volume contribution of $Co_{IP}(A_1 \cup A_2)$ = 363 units.

7.2.3.2.2 Port Number: Given two attacks (A_1, A_2), a set of port numbers P_1=$\{Port_1$, $Port_2$, ...$Port_n\}$ that are targeted by A_1, and a set of IP address P_2=$\{Port_a$, $Port_b$, ...$Port_x\}$ that are targeted by A_2, the surface intersection of both attacks based on the port number is calculated as the elements that belong to both set of port numbers (P_1, P_2) times their corresponding

weighting factor, as shown in Definition 6.3.

Definition 6.3: Port-based volume intersection
$Co_{Port}(A_1 \cap A_2) = Count(class1 \in P_1 \wedge P_2) \times WF(class1) + Count(class2 \in P_1 \wedge P_2) \times WF(class2) + Count(class3 \in P_1 \wedge P_2) \times WF(class3) + Count(class4 \in P_1 \wedge P_2) \times WF(class4)$.

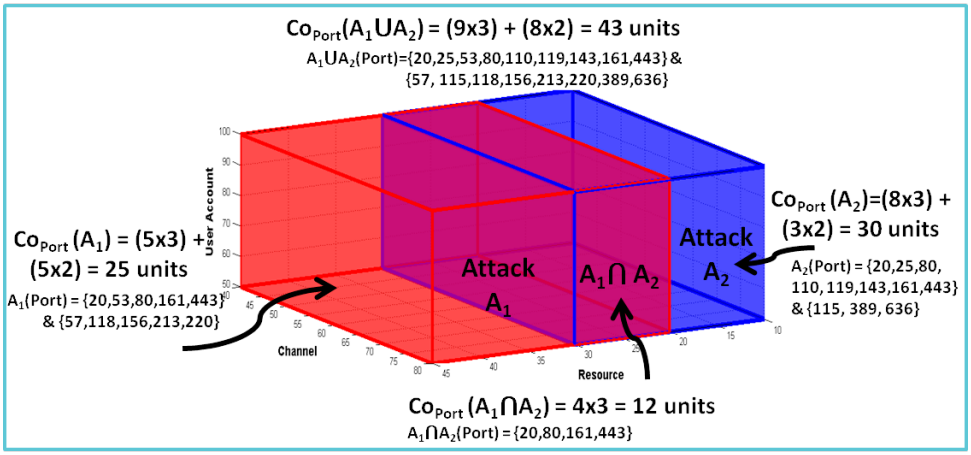

FIGURE 7.3 - Port-based volume union and intersection

In Figure 7.3, Attack A_1 affects the execution of 10 ports, five of which belong to Class 1 (i.e. 20, 53, 80, 161, 443), and the rest belong to Class 2 (i.e. 57, 118, 156, 213, 220); whereas Attack A_2 has 11 target ports, 8 of which belong to Class 1 (i.e. 20, 25, 80, 110, 119, 143, 161, 443) and the rest belong to Class 2 (i.e. 115,389, 636). The union of both attacks covers 17 ports, nine of which belong to Class 1, and the rest belong to Class 2; the port volume contribution based on the union of A_1 and A_2 is therefore calculated as $Co_{Port}(A_1 \cup A_2) = (9 \times 3) + (8 \times 2) = 43$ units (using the weighting factor from Table 6.9). The intersection of attacks A_1 and A_2 covers 4 ports, all of them belong to Class 1, the port volume contribution based on the intersection of both attacks is therefore calculated as $Co_{Port}(A_1 \cap A_2) = (4 \times 3) = 12$ units.

Using Definition 1.2, we are able to determine the total volume for the union and intersection of the channel dimension (IP-Port). In the previous example, the contribution Ip-Port for attack A_1 is equal to 292 units, the contribution Ip-Port for attack A_2 is equal to 168 units. The channel dimension contribution based on the union of both attacks is calculated as $Co_{Ip-Port}(A_1 \cup A_2) = 363 + 43 = 406$ units; and the channel dimension contribution based on the intersection of both attacks is calculated as $Co_{Ip-Port}(A_1 \cap A_2) = 42 + 12 = 54$ units

7.2.3.3 Resource:

Given two attacks (A_1, A_2), a set of resources R_1=$\{Res_1, Res_2, ...Res_n\}$ that is targeted by A_1, and a set of resources R_2=$\{Res_a, Res_b, ...Res_x\}$ that is targeted by A_2, the surface intersection of A_1 and A_2 based on the resource dimension is calculated as the sum of the elements that belong to both set of resources (R_1, R_2) times their corresponding weighting factor, as shown in Definition 6.4.

Definition 6.4: Resource-based volume intersection
$Co_{Res}(A_1 \cap A_2) = \sum_{R \in R_1 \cap R_2} WF(R)$

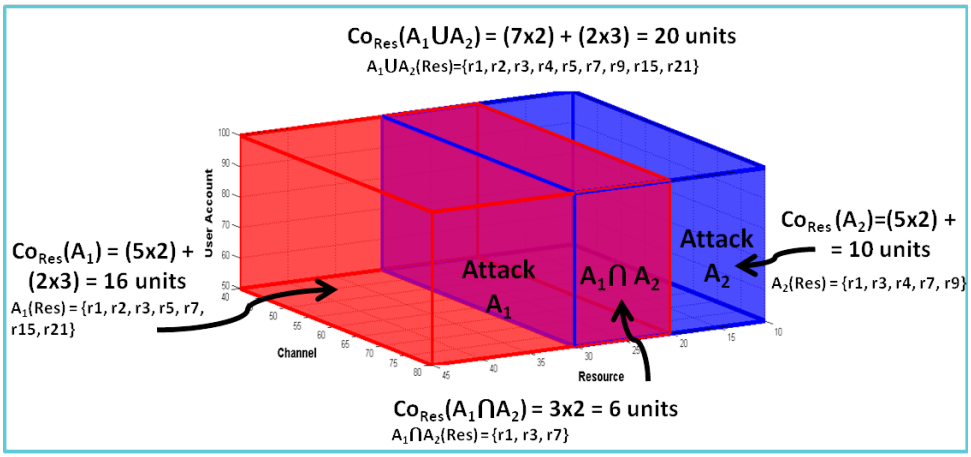

FIGURE 7.4 - Resource-based volume union and intersection

In Figure 7.4, Attack A_1 compromises 5 machines with user privilege, and Read-Write-Execute (RWX) permissions (i.e. r1, r2, r3, r5, r7), and 2 machines with kernel privilege and Write/Execute (W/X) permissions (i.e. r15, r21), which represents a volume contribution based on the resource dimension of $Co_{Res}(A_1) = (5 \times 2) + (2 \times 3) = 16$ units (using the weighting factor from Table 6.10). Attack A_2 compromises 5 machines with user privilege, and Read-Write-Execute permissions (i.e. r1, r3, r4, r7, r9), which represents a volume contribution based on the resource dimension of $Co_{Res}(A_2) = (5 \times 2) = 10$ units. The union of both attacks covers 7 machines with user privilege & RWX permissions, and 2 machines with kernel privilege & W/X permissions; the resource volume contribution based on the union of A_1 and A_2 is therefore calculated as $Co_{Res}(A_1 \cup A_2) = (7 \times 2) + (2 \times 3) = 20$ units. The intersection of attacks A_1 and A_2 covers 3 machineswith user privilege & RWX permissions, the resource volume contribution based on the intersection of both attacks is therefore calculated as $Co_{Res}(A_1 \cap A_2) = (3 \times 2) = 6$ units.

7.3 Countermeasure Volume for multiple attacks

Three cases are distinguished in the calculation of the countermeasure volume for a combined attack (i.e. totally joint, totally disjoint, and partially joint volumes).

7.3.1 Totally Joint Volumes:

If the volume of attack A_1 is totally covered by the volume of attack A_2, countermeasures against A_2 are also used against A_1 (Figure 7.5).

For instance, let us suppose that attack A_1 has as target the IP address range [172.20.4.195 - 172.20.4.240], and attack A_2 has as target the IP address range [172.20.0.1 - 172.20.15.254]. Attack A_2 targets a wider range of IP addresses (including the target of attack A_1), therefore, only Attack A_2 is analysed and countermeasures for this latter are proposed to face both attacks. The following

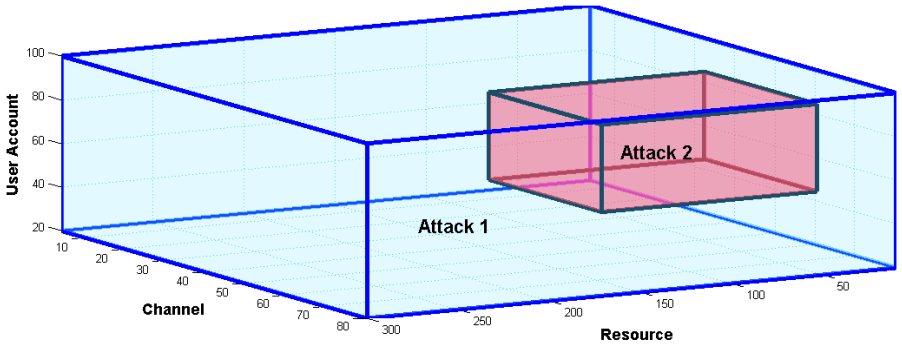

FIGURE 7.5 - Totally joint volumes

definition applies for totally joint surfaces:

Given two attacks (A_1, A_2), Attack A_1 is a subset of Attack A_2 iff the volume of A_1 is a subset of the volume of A_2, therefore, countermeasures(C) for attack A_1 are also a subset of the countermeasures for attack A_2, as shown in Definition 7.

Definition 7: Countermeasures for totally joint volumes
$A_1 \subseteq A_2$, *then* $C(A_1) \subseteq C(A_2)$

7.3.2 Totally Disjoint Volumes:

Attacks A_1 and A_2 are totally disjoint, their volumes are completely different, e.g. they have different targets (Figure 7.6).

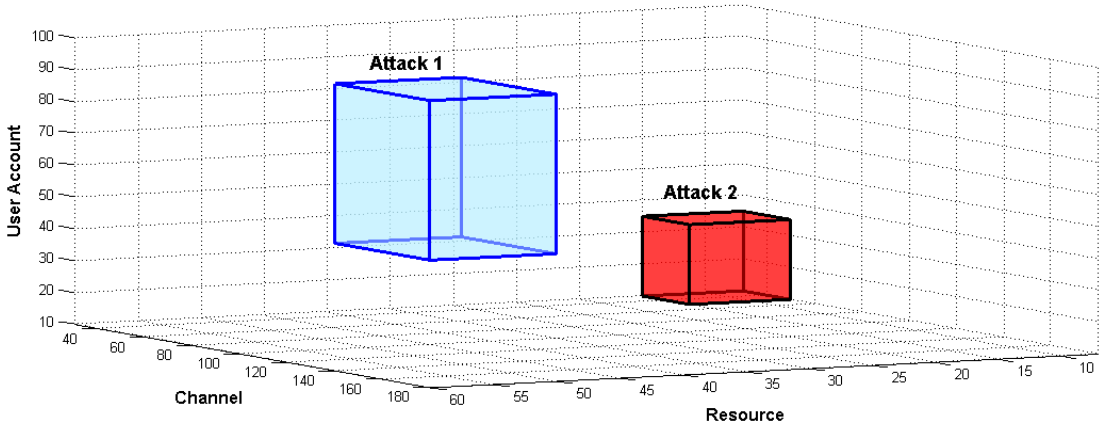

FIGURE 7.6 - Totally disjoint volumes

For instance, given an attack A_1 that targets the IP address range [172.20.4.120 - 172.20.4.194], and attack A_2 that targets the IP address range [172.20.15.197 - 172.20.15.240], their volumes are disjoint since they do not target the same IP addresses. Since both attacks have no target in common, they are treated individually, assuming that countermeasures for A_1 do not generate conflicts with those for A_2. The volume of attack A_1 is therefore independent of the volume of attack A_2. The following definition applies for totally disjoint surfaces:

Given two attacks (A_1, A_2), Attack A_1 is independent of Attack A_2 iff their individual volumes do not have elements in common, therefore, countermeasures (C) for attack A_1 are different from those of attack A_2, assuming that no conflict originates from their implementation, as shown in Definition 8.

Definition 8: Countermeasures for totally disjoint volumes
$A_1 \cap A_2 =$ *then,* $C(A_1) \neq C(A_2)$

7.3.2.1 Partially Joint Volumes:

If the volume of attack A_1 is partially covered by the volume of attack A_2, then one countermeasure is not sufficient to mitigate both attacks (Figure 7.7).

For instance, let us suppose that attack A_1 has as target the IP address range [172.20.4.195 - 172.20.4.240], and attack A_2 has as target the IP address range [172.20.4.120 - 172.20.4.208]. Only the IP addresses range [172.20.4.195 - 172.20.4.208] is common for both attacks and therefore one countermeasure will not cover the total attack surface area.

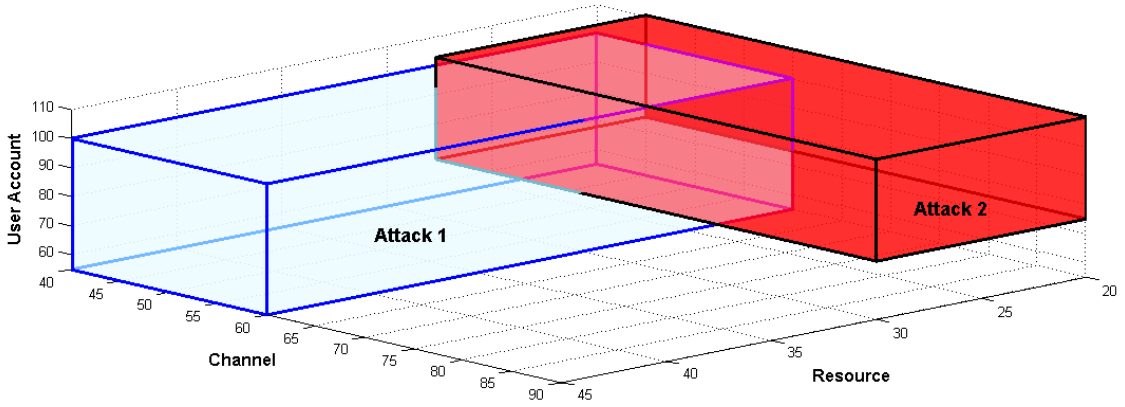

FIGURE 7.7 - Partially joint volumes

The following definition applies for partially joint surfaces:

Given two attacks (A_1, A_2), Attack A_1 is partially joint with Attack A_2 iff (i) their combined surface is not independent; (ii) A_1's volume is not a subset of A_2's volume; and (iii) A_2's volume is not a subset of A_1's volume, therefore, countermeasures (C) against one attack is not a proper subset of the countermeasures against the other attack, as shown in Definition 9.

Definition 9: Countermeasures for partially joint volumes
iff (i) $A_1 \cap A_2 \neq$ *and (ii)* $A_1 \nsubseteq A_2$ *and (iii)* $A_2 \nsubseteq A_1$ *then,*
$C(A_1) \nsubseteq C(A_2) \wedge C(A_2) \nsubseteq C(A_1)$.

7.4 Use Case: Multiple Attacks

Telecom SudParis and Telecom Management are two French state-funded schools for engineers and managers with more than 3000 student accounts (as of September 2013), and over 600 personnel and administrative accounts, as shown in Table 7.1.

Table 7.1 - Telecom SudParis User Accounts

Account Group	**Role**	**Quantity**	**Weight Factor**
Administrative personnel and technicians	Super administrator	263	4
Professors / Researchers	System administrator	165	3
PhD Students	Standard user	188	2
Interns	Standard user	17	2
Students	Internal users	3,058	1

In addition, the schools have been assigned a class B network composed of an IP address range 157.159.0.0/16, which represents a total of 65,536 public IP addresses (from which only 4,500 are active), and use class 1 ports (a total of 12 port numbers) in their communications. The main types of resources (i.e. machines (hosts), database servers, printers, and websites, shared by these two institutions are shown in Table 7.2.

Table 7.2 - Telecom SudParis Resources

Resource	Access Privilege	**Quantity**	**Weight Factor**
Host*	Kernel & WRX	30	5
Database	Kernel & WRX	10	5
Website	Kernel & WR/WX/RX	3	4
Printer	Kernel & W/X	50	3
Host	User & WRX, User & WR/WX/RX, Kernel & R	900	2

The authentication service is controlled and maintained by the Lightweight Directory Access Protocol (LDAP). In a normal operation, students, and personnel access the school resources by providing a combination of user login and password. Each element from the user accounts, channels and resources is assigned a weighting factor, as discussed in Section 6.2.3. Table 7.3 summarizes the weighting factor results.

TABLE 7.3 - Telecom SudParis Resources

Dimension	Range	Description	Quantity	Weight Factor
User Account	U1:U263	super administrator	263	4
	U264:U428	system administrator	165	3
	U429:U633	standard user	205	2
	U664:U3721	internal user	3058	1
Channel	Ch1:Ch4500	active public IP address	4500	3
	Ch4501:Ch4512	Port Class 1	12	3
Resource	R1:R40	kernel&WRX	40	5
	R41:R43	kernel&WR/WX/RX	3	4
	R44:R93	kernel&W/X	50	3
	R94:R993	user&WRX, user& WR/WX/RX, kernel&R	900	2

System Volume Calculation (SV):

1. We need to calculate the contribution (Co) of each dimension. For this, we apply Definition 1:

   $Co_{Acc}(S_1) = (263 \times 4) + (165 \times 3) + (205 \times 2) + (3,058 \times 1)$

   $Co_{Acc}(S_1) = 5,015$ units (Definition 1.1)

   $Co_{Ip}(S_1) = (4,500 \times 3) = 13,500$ units (Definition 1.2.1)

   $Co_{Port}(S_1) = (12 \times 3) = 36$ units (Definition 1.2.2)

   $Co_{Ip-Port}(S_1) = 13,536$ units (Definition 1.2)

   $Co_{Res}(S_1) = (40 \times 5) + (3 \times 4) + (50 \times 3) + (900 \times 2)$

   $Co_{Res}(S_1) = 2,212$ units (Definition 1.3)

2. We need to calculate the system volume. For this, we apply Definition 2:

   $SV(S_1) = (5,015 \times 2) \times (13,536 \times 1) \times (2,112 \times 1.5)$

   $SV(S_1) = 430,106,901,440\ units^3$

7.4.1 Attack Scenario

Telecom SudParis and Telecom Management schools have been targeted with sophisticated attacks. The attacks originate after a group of students visited a compromised website redirected through Facebook. The compromised website hosted exploits which then allowed malware to be installed in the students PCs. This section describes two separated attacks that originate simultaneously in the system, one of which is performed as a sequence of events.

7.4.1.1 Attack 1: Zeus

Zeus works in social networks, under the assumption that people will click links disguised as fan pages, social shares, and even friend profiles [Hon, BS12]. Students at Telecom SudParis and

Telecom Management are prompted to install a malware after clicking a link appearing in their Facebook accounts. The malware sits dormant on the system until users access their bank account, at which point it makes a copy of their user-names and passwords.

Zeus targets 38 system admin accounts (i.e. U340:U377), 21 public IP addresses (i.e. Ch100:Ch120), and 21 hosts with user privilege (i.e. R110:R130). The geometric figure of this attack is depicted in Figure 7.8.

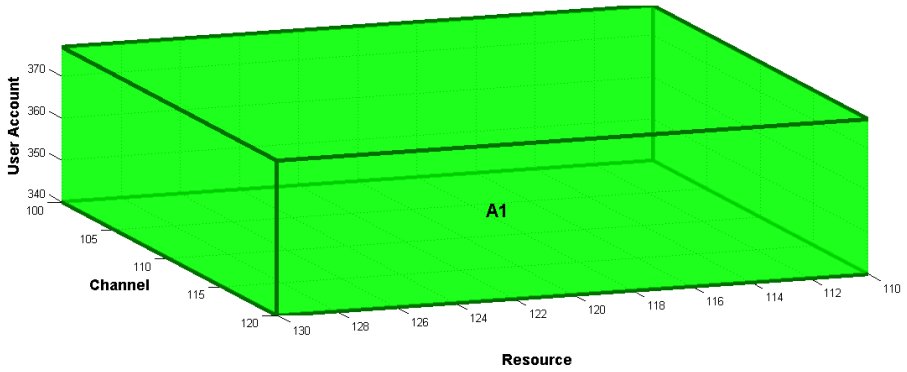

FIGURE 7.8 - Graphical Representation of the Zeus Attack

The volume of the Zeus attack is calculated as the product of the three axes (Definition 3).

$$\text{AV(A1)} = [(38 \times 3) \times 2] \times [(21 \times 3) \times 1] \times [(21 \times 2) \times 1.5] = 904{,}932\ units^3$$

The Zeus attack contribution in the system volume is determined using Definition 3.1:

$C(A_1)/(S_1) = \frac{904{,}932}{430{,}106{,}901{,}440} \times 100 = 0.0002\%$

7.4.1.2 Attack 2 (Step 1): Conficker

A second attack, (i.e. Conficker), occurs simultaneously in the system. Conficker is a computer worm targeting the Microsoft Windows operating system, which uses flaws in Windows software and dictionary attacks on administrator passwords to propagate while forming a botnet [LW09, Pis10].

The attack targeted 30 system admin accounts (i.e. U320:U349), 50 student accounts (i.e. U1110:U1159), 80 public IP addresses (Ch70:Ch149), 5 hosts* with kernel privilege (i.e. R5:R9) and 13 hosts with user privilege (i.e. R115:R127), all of them through port 80. The volume of this attack is represented by the union of 4 parallelepipeds, as depicted in Figure 7.9.

Since the four parts of the Conficker attack are disjoint, the volume of this attack is calculated as the union of attacks A2.1, A2.2, A2.3, and A2.4 (Definition 5.1). Following Definitions 3 and 6, we compute the attack volume of each individual portion of the attack.

$$\text{AV(A2.1)} = [(50 \times 1) \times 2] \times [(80 \times 3) \times 1] \times [(5 \times 5) \times 1.5] = 900{,}000\ units^3$$
$$\text{AV(A2.2)} = [(50 \times 1) \times 2] \times [(80 \times 3) \times 1] \times [(13 \times 2) \times 1.5] = 936{,}000\ units^3$$
$$\text{AV(A2.3)} = [(30 \times 3) \times 2] \times [(80 \times 3) \times 1] \times [(5 \times 5) \times 1.5] = 1{,}620{,}000\ units^3$$
$$\text{AV(A2.4)} = [(30 \times 3) \times 2] \times [(80 \times 3) \times 1] \times [(13 \times 2) \times 1.5] = 1{,}684{,}800\ units^3$$

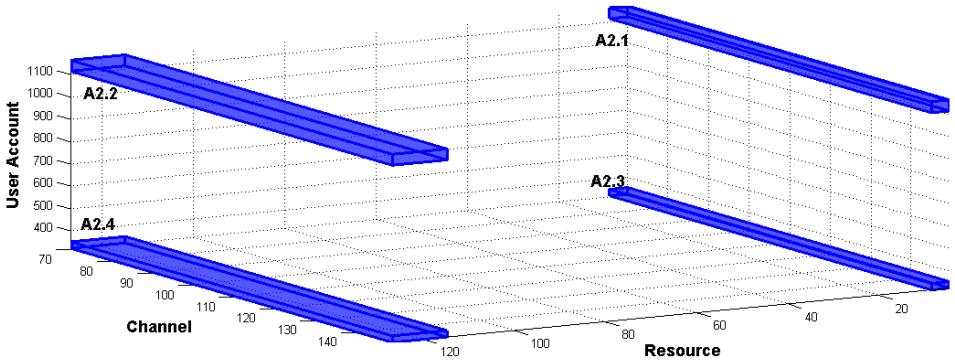

FIGURE 7.9 - Graphical Representation of the Conficker Attack

Then, we are able to calculate the total volume of the Conficker attack (Definition 5.1):

AV(A2) = (900,000 $units^3$) + (936,000 $units^3$) + (1,620,000 $units^3$) + (1,684,800 $units^3$)
AV(A2)= 5,140,800 $units^3$

The Conficker attack contribution in the system volume is determined using Definition 3.1:

$C(A_2)/(S_1) = \frac{5{,}140{,}800}{430{,}106{,}901{,}440} \times 100 = 0.0012\%$

As a result, the Conficker attack represents less than 1% of the total system volume. In addition, A2 is 5 times bigger than A1, both attacks are partially covered (Attack A1 covers a portion of Attack A2.4), we need, therefore, to calculate the volume of the combined attack (i.e. A1 & A2). For this, we use Definition 5.2 which allows us to calculate the union and intersection for joint attacks.

Let us first calculate the intersection portion of the attacks. By interposing the geometric representation of both attacks, we identify that 10 system admin accounts (i.e. U340:U349), 21 public IP addresses (i.e. Ch100:Ch120), and 13 hosts with user privilege (R115:R127), are present in both, Zeus and Conficker attacks (Figure 7.10).

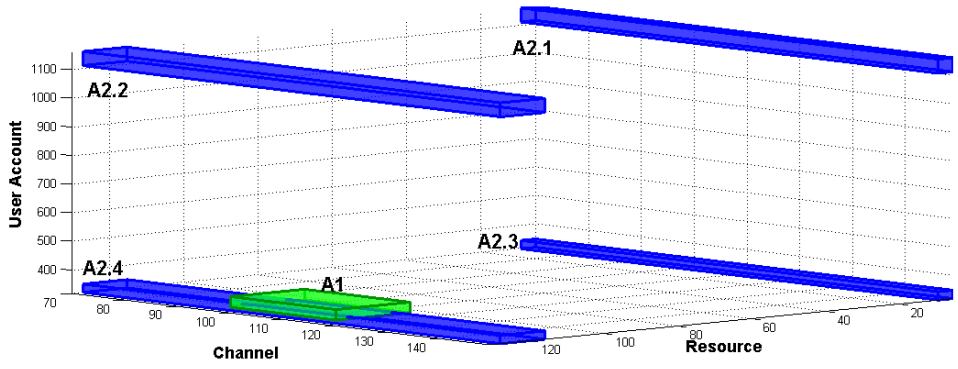

FIGURE 7.10 - Graphical Representation of Zeus and Conficker Attacks

The intersection of these two attacks is calculated as follows:

$AV(A_1 \cap A_2) = [(10 \times 3) \times 2] \times [(21 \times 3) \times 1] \times [(13 \times 2) \times 1.5] = 147{,}420\ units^3$

This is the priority area to mitigate. Countermeasures should be first implemented taking into account users, channels and resources in this region.

Using Definition 5.2, we determine the union of the joint attacks as:

$AV(A_1 \cup A_2) = AV(A_1) + AV(A_2) - AV(A_1 \cap A_2)$
$AV(A_1 \cup A_2) = 904{,}932\ units^3 + 5{,}140{,}800\ units^3 - 147{,}420\ units^3 = 5{,}898{,}312\ units^3$

7.4.1.3 Attack 2 (Step 2): Sequential Conficker

After the Conficker attack compromised the user accounts, channels and resources previously described, the system detected that the compromised administrator accounts (i.e. U320:U349) performed a brute force attack over the 10 databases of the system (i.e. R31:R40). This sequential attack is also performed through channels Ch70:Ch149. Since the attack has changed, it is necessary to recalculate the attack volume.

AV(A2.5) = $[(50 \times 1) \times 2] \times [(80 \times 3) \times 1] \times [(10 \times 5) \times 1.5] = 1{,}800{,}000\ units^3$
AV(A2.6) = $[(30 \times 3) \times 2] \times [(80 \times 3) \times 1] \times [(10 \times 5) \times 1.5] = 3{,}240{,}000\ units^3$

Since this portion of the attack is disjoint with the other 4 portions (i.e. A1.1, A1.2, A1.3, and A1.4), we apply Definition 5.1 to calculate the volume of the sequential conficker attack:

AV(A2) = $(900{,}000\ units^3) + (936{,}000\ units^3) + (1{,}620{,}000\ units^3) + (1{,}684{,}800\ units^3) + (1{,}800{,}000\ units^3) + (3{,}240{,}000\ units^3) = 10{,}180{,}800\ units^3$

The union of the sequential Conficker and the Zeus attacks is calculated as:

$AV(A_1 \cup A_2) = 904{,}932\ units^3 + 10{,}180{,}800\ units^3 - 147{,}420\ units^3 = 10{,}938{,}312\ units^3$

The graphical representation of the sequential Conficker attack and the Zeus attack is depicted in Figure 7.11

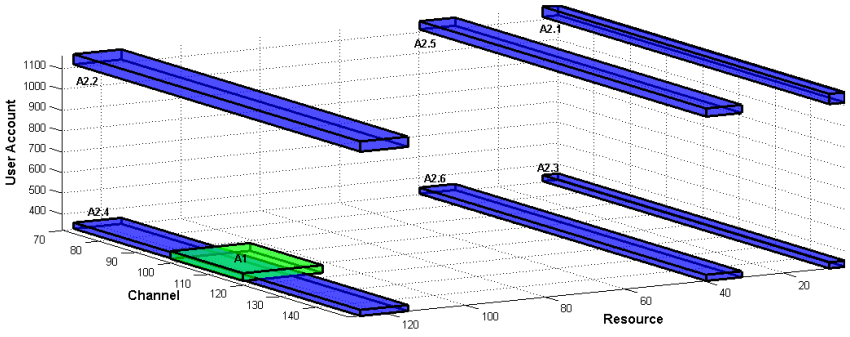

FIGURE 7.11 - Graphical Representation of Sequential Conficker & Zeus Attacks

The Zeus and the sequential Conficker attack contribution in the system volume is determined as:

$C(A1 \cap A_2)/(S_1) = \frac{10{,}938{,}312}{430{,}106{,}901{,}440} \times 100 = 0.0025\%$

7.4.2 Countermeasure Analysis

Several countermeasures are proposed to react over the Zeus and the Conficker attacks [Kri09,Gud]. However, due to space constraint, we evaluate the three most common solutions for these attack. The most common countermeasures used to mitigate the Zeus attack are described as follows:

C1.1 Use behavioral detection techniques (e.g. buffer overflow protection in Host IPS)

C1.2 Download and install antivirus/antimalware on all the machines of the network

C1.3 Make all shares "read-only", the trojan easily spreads via shares

The most common countermeasures used to mitigate the Conficker attack are described as follows:

C2.1 Download and install patches for windows (e.g. KB958644, KB957097, KB958687) on all the machines of the network and install antivirus/antimalware (e.g. McAfee VirusScan, ToPS for endpoints), or tools for identification of local infections and for vaccination (e.g. Netflow)

C2.2 Block access to a list of domains by using a proxy server (e.g. Squid)

C2.3 Create signatures for matching against the shellcode pattern and use them with IDS/IPS (e.g. Snort)

Table 7.4 summarizes the total coverage and effectiveness information of each countermeasure.

TABLE 7.4 - Countermeasure Information

Countermeasure	User Account	Channel	Resource	Effectiveness
C1.1	U300:U349	Ch1:Ch149	R121:R123	60%
C1.2	U301:U433	Ch100:Ch179	R94:R193	70%
C1.3	U330:U360	Ch1:Ch110	R1:R119	50%
C2.1	U229:U550	Ch51:Ch110	R94:R130	70%
C2.2	U270:U449	Ch70:Ch149	R1:R30	80%
C2.3	U1030:U1160	Ch40:Ch90	R1:R123	75%

7.4.2.1 Countermeasure Volume

Following Definition 4, we compute the volume of each individual countermeasure.

CV(C1.1) = $[(50 \times 3)\times 2] \times [(149 \times 3)\times 1] \times [(3 \times 2)\times 1.5] = 1{,}206{,}900\ units^3$
CV(C1.2) = $[(133 \times 3)\times 2] \times [(80 \times 3)\times 1] \times [(100 \times 2)\times 1.5] = 57{,}456{,}000\ units^3$
CV(C1.3) = $[(31 \times 3)\times 2] \times [(110 \times 3)\times 1] \times [((40 \times 5)+ (3 \times 4)+ (50 \times 3)+ (26 \times 2))\times 1.5] = 25{,}411{,}320\ units^3$

CV(C2.1) = [((35×4)+ (165×3)+ (122×2))×2] × [(60×3)×1] × [(37×2)×1.5] = 35,124,840 $units^3$
CV(C2.2) = [((159 × 3)+ (21 × 2))× 2] × [(80 × 3)× 1] × [(30 × 5) × 1.5] = 56,052,000 $units^3$
CV(C2.3) = [(131 × 1)× 2] × [(51 × 3)× 1] × [((40 × 5)+ (3 × 4)+ (50 × 3)+ (30 × 2))× 1.5] = 14,551,218 $units^3$

Table 7.5 summarizes the results of the volume calculation for system S1, attacks A1, A2, and each countermeasure.

TABLE 7.5 - Resulting Volumes

Element	User Account	Channel	Resource	Volume ($units^3$)
S1	U1:U3691	Ch1:Ch4512	R1:R993	430,106,901,440
A1	U320:U349& U1110:U1159	Ch70:Ch149	R5:R9& R31:R40& R115:R127	8,380,800
A2	U340:U377	Ch100:Ch120	R110:R130	904,932
C1.1	U300:U349	Ch1:Ch149	R121:R123	1,206,900
C1.2	U301:U433	Ch100:Ch179	R94:R193	57,456,000
C1.3	U330:U360	Ch1:Ch110	R1:R119	25,411,320
C2.1	U229:U550	Ch51:Ch110	R94:R130	35,124,840
C2.2	U270:U449	Ch70:Ch149	R1:R30	56,052,000
C2.3	U1030:U1160	Ch40:Ch90	R1:R123	14,551,218

The graphical representation of the attack scenario with the countermeasure candidates is depicted in Figure 7.12.

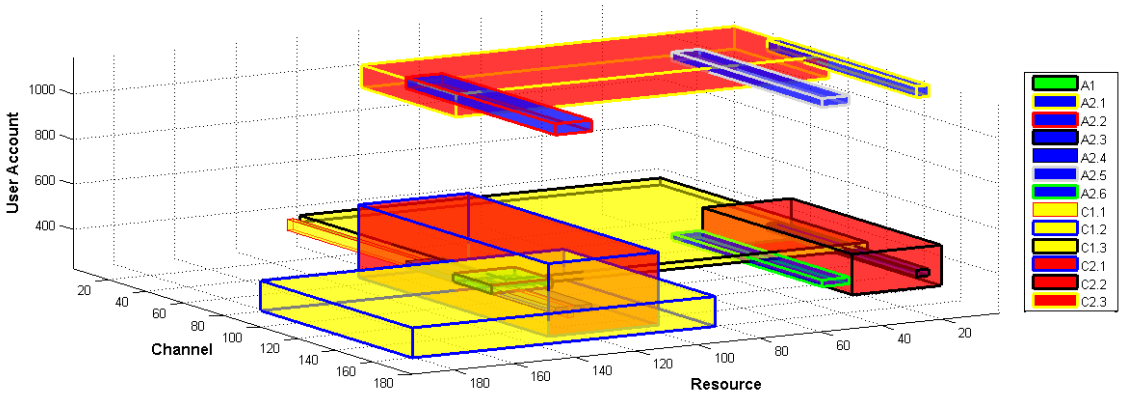

FIGURE 7.12 - Graphical Representation of attacks and countermeasures

In Figure 7.12 the red and yellow figures represent the countermeasure candidates, the blue parallelepipeds represent the five parts of the conficker attack, and the green parallelepiped represents the zeus attack. All countermeasure volumes are higher than the attack volumes, except for C1.1, whose volume is smaller than the conficker attack, but higher than the zeus attack. The implementation of such countermeasures may result in collateral damages (when the countermeasure volume is higher than the attack volume) and/or residual risk, otherwise. For instance, let us suppose that candidate C2.1 is selected as the optimal countermeasure. Its implementation will

cover a volume of 35,124,840 $units^3$, which is higher than the combined attack (whose volume is equivalent to 5,898,312 $units^3$). If the countermeasure covers the total attack volume, there will be around 30,000,000 $units^3$ that will cover other user accounts, channels, and resources. This latter can be considered as a collateral damage, if its implementation affects negatively such users, channels, and resources.

7.4.2.2 Countermeasure Coverage

Each countermeasure is represented as a parallelepiped that covers a set of user accounts, channels and resources from System S1. Such coverage represents only a portion of the combined attack (i.e. Conficker + Zeus). The calculation of the coverage of each countermeasure is performed using Definition 4.1. For this, we need to identify the elements (i.e. user accounts, channels, and resources) that belong to both, the selected countermeasure and the attack. In this case we want to determine the coverage of each countermeasure over the combined attack. Table 7.6 summarizes this information.

TABLE 7.6 - Countermeasure coverage elements

Counter-measure	**User Account**	**Channel**	**Resource**	**Coverage Volume ($units^3$)**
C1.1	U320:U349	Ch70:Ch149	R121:R123	388,800
C1.2	U320:U377	Ch100:Ch149	R110:R130	3,288,600
C1.3	U330:U360	Ch70:Ch110	R5:R9& R31:R40& R110:R119	3,260,115
C2.1	U320:U377	Ch70:Ch110	R110:R130	2,696,652
C2.2	U320:U377	Ch70:Ch149	R5:R9	3,132,000
C2.3	U1110:U1130	Ch70:Ch90	R5:R9& R31:R40 & R110:R123	408,807

Results from Table 7.6 show the elements (i.e. user account, channel, resource) from a given attack, that are controlled by a given countermeasure. For instance, we determine that none of the countermeasures covers 100% of the combined attack volume (i.e. $AV(A_1 \cup A_2) = 10{,}938{,}312$ $units^3$). Their individual implementation only covers a portion of both attack volumes. This coverage is calculated using Definition 4.1, as follows:

$$\text{Cov}(\text{C1.1}/A_1 \cup A_2) = \frac{388{,}800}{10{,}938{,}312} \times 100 = 3.55\%$$
$$\text{Cov}(\text{C1.2}/A_1 \cup A_2) = \frac{3{,}288{,}600}{10{,}938{,}312} \times 100 = 30.06\%$$
$$\text{Cov}(\text{C1.3}/A_1 \cup A_2) = \frac{3{,}260{,}115}{10{,}938{,}312} \times 100 = 29.80\%$$
$$\text{Cov}(\text{C2.1}/A_1 \cup A_2) = \frac{2{,}696{,}652}{10{,}938{,}312} \times 100 = 24.65\%$$
$$\text{Cov}(\text{C2.2}/A_1 \cup A_2) = \frac{3{,}132{,}000}{10{,}938{,}312} \times 100 = 28.63\%$$
$$\text{Cov}(\text{C2.3}/A_1 \cup A_2) = \frac{408{,}807}{10{,}938{,}312} \times 100 = 3.74\%$$

As a result, the individual application of any of the countermeasures represents a partial mitigation of the combined attack, that in the best of the cases reaches 30.06% of the total risk, meaning

that part of the attack is not treated at all. The residual risk is, therefore, calculated as the difference between the attack volume and the countermeasure coverage volume. In our example, the residual risk ranges from 60% to 95% as shown in Table 7.7.

TABLE 7.7 - Residual risk and collateral damage

Counter-measure	**Coverage (%)**	**Residual Risk ($units^3$)**	**Residual Risk (%)**	**Collateral Damage ($units^3$)**	**Collateral Damage (%)**
C1.1	3.55%	10,549,512	96.45%	818,100	67.79%
C1.2	30.06%	7,649,712	69.94%	54,167,400	94.28%
C1.3	29.80%	7,678,197	70.20%	22,151,205	87.17%
C2.1	24.65%	8,241,660	75.35%	32,428,188	92.32%
C2.2	28.63%	7,806,312	71.37%	52,920,000	94.41%
C2.3	3.74%	10,529,505	96.26%	14,142,411	97.19%

In addition, since only one part of the countermeasure volume is useful to mitigate the attack, the remaining countermeasure volume is considered as a collateral damage, meaning that the application of any of the countermeasures implies that the treatment is going to be performed in all associated user accounts, channels, and resources associated to the given countermeasure, even for those elements that are not covered by any of the attack volumes. The collateral damage is, therefore, calculated as the difference between the total countermeasure volume and the attack-based countermeasure coverage. For instance, the total volume of C2.1 is determined as 35,124,840 $units^3$ and the volume coverage of C2.1 is 2,696,652 $units^3$, the collateral damage for C2.1 is therefore 32,428,188 $units^3$, which represents 92.32% of the total countermeasure volume.

7.5 Conclusion

In this chapter we introduced a geometric approach to calculate the volume of the union and intersection of multiple attacks. The approach considers joint and/or disjoint attacks, and proposes equations for their corresponding evaluation. The contribution of each attack vector (i.e. user account, channel, resource) is analysed in order to model the required operations to calculate their union and intersection. Examples on the calculation of the attack volume based on each attack vector are provided accordingly.

The proposed approach provides a clear representation of attacks and countermeasures in a given system, and the possibility to identify priority areas (i.e. those with the highest attack volume, or where multiple attacks intersect). Consequently, it is possible to detect the users, channels, and resources that are the most vulnerable in the system, in order to define the reaction strategies to apply.

Another important aspect to discuss is the fact that our approach proposes an accurate and quantitative methodology to evaluate countermeasures for multiple attack scenarios. Three cases are considered in the countermeasure evaluation: totally joint attacks, totally disjoint attacks, and partially joint attacks. As a result, countermeasure volumes are analysed and compared, making it possible to determine the coverage of each countermeasure in each of the studied attacks.

The case study discussed at the end of this chapter considers specific information regarding user accounts, channels, and resources provided by an educational institution in France. We did not reuse one of the scenarios discussed in Part I of the dissertation, since they do not provide

enough information to deploy our model. As a consequence, we can observe that applying only one countermeasure is not enough to mitigate the combined attack (Conficker & Zeus). None of the countermeasures implemented individually protect 100% of the users, channels, and resources affected by both attacks. It is, therefore, necessary to implement multiple countermeasures as described in Definition 8. However, it is worth noting that in this example, even the implementation of all countermeasures does not provide a mitigation of 100% of the combined attacks.

As a result, mitigation strategies should be considered according to the objectives and goals of each organization. For instance, some organizations could be very concerned about residual risks. Strategies should consider the implementation of countermeasures that mitigate the maximum volume of attacks, regardless of the collateral damage they may cause. On the contrary, for those organizations that care about collateral damage, the mitigation strategy should consider the implementation of the least number of countermeasures, as well as the implementation of very punctual countermeasures.

Chapter 8

Conclusion

> The open mind never acts: when we have done our utmost to arrive at a reasonable conclusion, we still - must close our minds for the moment with a snap, and act dogmatically on our conclusions.
>
> George Bernard Shaw - 1856-1950

Throughout this thesis we proposed a quantitative approach to select optimal security countermeasures based on the Return On Response Investment (RORI) index. The proposed solution is twofold. In the first part of the dissertation, we explored individual attack scenarios to evaluate the expected losses that result from a particular attack versus the benefits that can be obtained if a countermeasure is implemented. Countermeasures are evaluated individually and then a combination is performed among all candidates in order to select the countermeasure or group of countermeasures that provides the highest benefit to the organization. In the second part of the dissertation, we studied the notion of attack surface in order to propose an approach that calculates the volume of each attack, and through geometrical operations, we determined the union and intersection of multiple attacks occurring simultaneously in a given system. As a result, optimal countermeasures are selected for concurrent or multi-step attack scenarios, based on the the volume they cover on the system.

8.1 Contributions

In response to *Objective 1.1*, we proposed the Return On Response Investment (RORI) model for evaluating and ranking individual countermeasures in a single attack scenario. The model evaluates not only the cost and benefit of a given security solution, but also the impact of the attack over a given system, and the organization infrastructure value. As a result, our model selects countermeasures from a pool of candidates, and ranks them based on a trade-off between their efficiency in stopping the attack observed by the SIEM, and their ability to preserve, at the same time, the best service to normal users. In response to *Objective 1.2*, we proposed a combination approach to evaluate and select combined countermeasures to mitigate the effects of individual attacks. The approach is divided into two phases: the evaluation of the parameters associated to the intrusion or attack (e.g., ALE and AIV); and the evaluation of the parameters

associated to the combined solutions (e.g., ARC and RM). As a result, the system is able to rank and select the countermeasure and/or combined solution that provides the highest cost-effectiveness ratio, thus the highest benefit to the organization.

In order to fulfil *Objective 1.3*, we proposed in the first part of the dissertation, a complete methodology to estimate each parameter of the RORI model and select the countermeasure(s) that provides the highest RORI index, thus the highest benefit to the organization. In the second part of the dissertation, we proposed a notion of attack volume, extending the notion of attack surface, to evaluate multiple attacks and multiple countermeasures. The approach models the volume of each attack based on a three-dimensional coordinate system (i.e. user, channel, and resource). Each axis from the coordinate system is weighted using the CARVER methodology. As a result, we are able to connect volumes with risks (e.g. big volume equals high risk, small volume equals low risk).

In response to *Objective 1.4*, we provided in the first part of the dissertation, a deployment of the RORI index and the operations required to evaluate and select optimal countermeasures over two real case scenarios: the first one, a Mobile Money Transfer Service (provided by a large telecommunication company based in France), and the second one, a Critical Infrastructure Control Process (Dam) (provided by a medium-sized telecommunication company based in Italy). Both use case deployments show the applicability of the RORI model and demonstrates that in most cases, combined countermeasures provide a higher RORI index and thus a higher benefit to the organization. In the second part of the dissertation, we proposed a methodology to evaluate multiple attacks occurring simultaneously on the system. The methodology considers the volume each attack covers on a given system and provides a geometric approach to calculate the countermeasure coverage based on the union and intersection of the individual volumes. A case study with two simultaneous attacks is deployed in order to show the applicability of our proposed model. As a result, countermeasures are proposed for complex and/or combined attacks.

In response to *Objective 1.5*, an implementation of the RORI model using the OrBAC formalism is provided over the Mobile Money Transfer use case. The implementation models the transactions performed by mobile users (e.g. connect, authenticate, receive and emit money) in two contexts: default context (when no attack is detected), and attack context (when the system detects a given attack). As a result, security policies are activated/deactivated according to the defined context. In order to solve conflicts, we assign a priority to each security policy, based on the RORI index. Thus, the higher the RORI index, the higher the priority assigned.

8.2 Perspectives

Perspectives for future work concentrate in two main aspects: extension of the RORI model, and improvement of the attack volume formalism.

8.2.1 RORI Model Extensions

The RORI model should consider other parameters such as reaction time, recovery time, vulnerability, and effort, in order to perform a dynamic and automated response to intrusions and attacks. The use of the OrBAC model could allow the definition of such parameters in temporal, spacial and user defined contexts. For instance, the evaluation of the RORI model should consider the time at which a given user is connected into the system (e.g. operational hours, weekends, night time) as well as his/her geographical location (remote access, in-site, etc), making the reaction process an active and dynamic task.

In addition, the RORI model could be greatly improved by considering service dependencies

in the selection of countermeasures. In this sense, the work of Kheir et al. [KCBCD10, Khe10] has demonstrated that the use of service dependencies for assessment of intrusion and response impacts provides enough evidence to compare response alternatives and to select optimal intrusion responses. Merging the notion of service dependencies with the proposed RORI model could be an interesting subject of further research.

One research that could be fully integrated with our model is the work performed by Ben Mustapha et al. [MBD13]. The approach considers the Responsibility Domain of a Policy Enforcement Point (RD(PEP)), that is, the capability and ability for a PEP to enforce security policies. The main objective is to build a consistent view of the deployed policy enforcement capabilities (through approximations) that may contribute in defining the appropriate response strategy. As a result, alerts are managed and correlated according to the PEP capabilities, and countermeasures could be easily evaluated in each responsibility domain.

Another aspect that should be considered for future work is to improve detection in order to enhance the reaction process. A first approach has been done within the MAnagement of Security information and events in Service Infrastructures (MASSIF) Project[1]. The work of Coppolino et al. [CDER09] and Romano et al. [RDFC] demonstrate that processing events at the edge of the platform significantly reduces event volume for scalability and allows to bound the analysis of sensitive data at the edge of the SIEM architecture. As a result, different levels of mitigation for the detected attack (e.g., simple, strong, enhanced, etc) could be implemented, which in turn translates into different levels of countermeasures.

8.2.2 Attack Volume Improvements

A validation and implementation of the attack volume formalism into a real case study should be important for its continuity and improvement. One validation approach could be the use of expert user survey proposed by Manadhata [Man08], where data is collected and analysed based on a survey questionnaire that measures the level of agreement or disagreement that selected subjects have with the measurement method. Another validation approach would be to implement the method in controlled environments that can provide real data regarding losses, attack, and countermeasures. Comparing the results obtained by applying our proposed model with the ones obtained in controlled scenarios could indicate how close or far our model is from reality.

Another aspect that should be considered in the attack volume model is the integration of other axes (e.g. time, contexts, etc) into the coordinate system. We have proposed in Chapter 6 to evaluate the attack volume using three axes (i.e. user account, channel, resource). However, the number of axes in our coordinate system may change. The proposed system is flexible enough to model the information retrieved by a URI into two or more dimensions, resulting in a variety of geometrical figures (e.g. squares, cube, hypercube) that are not initially considered in the calculation of the attack volume.

In Part II of this dissertation we proposed a method to relate volume and risk. However, we have not provided a methodology to relate volume with monetary cost, which in the end, provide the rules that security organizations could use to charge their clients according to the desired risk level. For instance, if a given organization requires a basic level of security (only a minimum protection), the contracted service should represent a lesser cost than the one provided to other organizations that require a higher level of security.

[1] http://www.massif-project.eu/

8.3 Final Word

To conclude, this research has been an opportunity to investigate a wide variety of concepts, models and technologies in the information and network security fields. Our objective was to analyse an issue that has not been addressed before: the selection of optimal countermeasures for multiple and complex attack scenarios. We provide a novel and systematic approach in response to the aforementioned issue, and we propose new ideas for further research.

List of Figures

List of Tables

Glossary of Acronyms

ADAS	Automated Data Acquisition System
AIV	Annual Infrastructure Value
ALE	Annual Loss Expectancy
ARC	Annual Response Cost
ARO	Annual Rate of Occurrence
AT	Account Takeover
Ci	Contracted insurance
CoI	Cost of Implementation
CoM	Cost of Maintenance
CP-net	Conditional Preference Networks
DMZ	Demilitarized Zone
DNP3	Distributed Network Protocol 3.0
DoS	Denial of Service
DS&R	Decision Support and Reaction
Ec	Equipment costs
EF	Effectiveness Factor
HIDS	Host-based Intrusion Detection System
Ic	Indirect costs
ICCP	Inter-Control Center Communications Protocol
ICT	Information and Communication Technology
IDS	Intrusion Detection System
IP	Internet Protocol
IPS	Intrusion Prevention System
IRR	Internal Rate of Response
IT	Information Technology
La	Loss of assets
LAN	Local Area Network
Ld	Loss of data
Lr	Loss of reputation
Lp	Legal procedures
Lrc	Loss of revenues from clients
MASSIF	MAnagement of Security information and events in Service InFrastructures
MMTS	Mobile Money Transfer Service

NIDS	Network-based Intrusion Detection System
NPV	Net Present Value
Oc	Other costs
Odc	Other direct costs
Ol	Other losses
OrBAC	Organization-based Access Control
Pc	Personnel costs
PEP	Policy Enforcement Point
PyOrBAC	Python implementation of the OrBAC model
RM	Risk Mitigation
ROA	Return On Attack
ROI	Return On Investment
RORI	Return On Response Investment
ROSI	Return On Security Investment
RTM	Remote Terminal Unit
Rv	Resell value
Sc	Service costs
SC	Surface Coverage
SCADA	Supervisory Control And Data Acquisition
SIEM	Security Information and Event Management
SLE	Single Loss Expectancy
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TR	TRansaction
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WSN	Wireless Sensor Network
XML	Extensible Markup Language
XOrBAC	XML file for PyOrBAC communications

Glossary of Terms

Account Takeover	A password-based attack that exploits vulnerabilities on the user's side (e.g., social engineering, key-loggers, etc.) and steals the mobile user account to perform transactions in favour of the attacker.
Administration Password Theft	An unfaithful employee of the dam, with a non administrator role but that is enabled to access to the control station uses stolen administrator credentials to open the dam's gates.
Attack	An attempt to gain unauthorized access to system services, resources, or information, or an attempt to compromise system integrity.
Attack Surface	Refers to the subset of resources used to attack a system. An attacker may use the system's entry and exit points (methods e.g., get, read, write, print,... that receive/send data items directly from/to the system's environment), channels (e.g., TCP, SSL, Unix socket,... used to connect to the system and invoke a method), and untrusted data items (e.g., files, cookies, database records, registry entries,... used by the system'users to send/receive data indirectly into/from the system) to attack the system.
Attack Surface Action	Refers to the system's total area exposed to a given attack. This surface includes tangible assets (e.g., PCs, mobile phones, network components, etc.), as well as intangible assets (e.g., confidential information, business reputation, etc).
Botnet	A number of Internet computers that, although their owners are most time unaware of it, have been set up to forward transmissions (including spam or viruses) to other computers on the Internet.
Collateral Damage	Costs that are added by a new response, and which are not related to intrusion costs. For instance, by modifying a configuration, the response affects users of the target system, and thus resulting in collateral and yet, unavoidable damages.
Combined Countermeasure	A combined countermeasure results from the simultaneous implementation of two or more countermeasures to mitigate a given attack. A combined solution is therefore analysed as a single countermeasure with a combined cost and a combined effectiveness.

Control Station Hacking	A malicious user takes the control of some machines in the control station from the visualization station, making it possible to modify sensor settings and commands to the actuators.
Controlled Faux Accounts	Accounts designed to lead intruders to believe that they are executing within a compromised standard account, when instead they are locked into a special limited access account. This technique eliminates the need for the separate hardware resources required by a faux system, but must rely on the target operating system security to ensure isolation from protected system resources.
Countermeasure	Actions, devices, procedures, or techniques that meet or oppose (i.e., counters) a threat, a vulnerability, or an attack by eliminating or preventing it, by minimizing the harm it can cause, or by discovering and reporting it so that corrective action can be taken.
Dam	It is an infrastructure used for water supplying, hydroelectric power generation, irrigation, water activities and wildlife habitat granting.
Denial of Service	An attempt against the availability of a particular system. The main objective is to disrupt service and network availability by attempting to reduce a legitimate user's bandwidth, or preventing access to service or system.
Discount Rate	The rate used to calculate the present value of future cash flows.
Discounted Cash Flow	Equal to future cash flows divided by discount rate factors to obtain present value.
Employee Complicity	An employee or a retailer is complicit with Money-Laundering activities and facilitates opening of account despite knowing that the account will be loaded with funds coming from criminal activities.
Financial Statement	Enterprise balance sheet, income statement, cash-flow statement, and/or statements of changes in owner's equity.
Hazardous Water Release	An entity commands water releasing operations on the dam. The discharging operation causes water movements that are hazardous for water recreational activities held on the reservoir.
Hiding User Identity	An account in the Mobile Monet Transfer Service system that only executes remote transactions.
Honey-pot	Systems or accounts used to lure the intruder into pursuing a decoy controlled environment directly of his own volition. Honeypots are placed near assets requiring protection, are made attractive, and are fully instrumented for intrusion detection and back tracking.

Host-based intrusion detection system (HIDS)	IDSs which operate on information collected from within an individual computer system. This vantage point allows host-based IDSs to determine exactly which processes and user accounts are involved in a particular attack on the Operating System.
Hydroelectric Power Plant Anti-Islanding Hacking	The hydroelectric power plant requests for anti-islanding support to the dam, in that the dam has to stop the water feeding the hydraulic turbines. An attacker intercepts the requests to the dam control station and hides the requests. The dam continues in feeding the turbine with water, causing their failure.
Infectious threats	It corresponds to viruses, worms, trojan horses and other similar threats.
Intrusion detection system (IDS)	Hardware or software product that gathers and analyzes information from various areas within a computer or a network to identify possible security breaches, which include both intrusions (attacks from outside the organizations) and misuse (attacks from within the organizations.)
Intrusion prevention systemv(IPS)	Systems which can detect an intrusive activity and can also attempt to stop the activity, ideally before it reaches its targets.
Mobile money (mMoney)	An electronic unit of monetary denominated in local currency and issued by the Bank.
Money Creation/ Destruction	Within the Mobile Money Transfer System, a certain amount of virtual money is authorized to circulate. This amount is strictly controlled to avoid any money creation and/or destruction. The amount should remain constant at all time and any change is considered as a fraud.
mWallet	Account hosted in the Mobile Money Transfer system, allowing the mWallet holder to carry out various actions with mMoney. An mWallet can also be referred to as an account.
Net Cash Flow	Projected revenues and cost savings less costs during a given period of time of a project.
Network-based intrusion detection systems (NIDS)	IDSs which detect attacks by capturing and analyzing network packets.
NOOP	No Operational action executed to mitigate a given attack.
Payback period	Time taken to recoup the original investment with the new revenue and/or cost savings from the project.
Policy Enforcement Point	Logical entity or place on a server that enforces policies for admission control and decisions in response to a request from a user wanting to access a resource on a computer or network server.

Polymorphic Virus	A virus that includes a scrambled body and decryption routine that first gain control of the computer, then decrypts the virus body, and finally adds a mutation engine that generates randomize decryption routines that change each time a virus infect a new program.
Quarantined Faux Systems	A system designed to lead intruders (primarily the unfamiliar outsider) to believe that they are logged into the target system, when they are actually locked into a separate controlled system. An effective quarantined faux system encourages an intruder to remain long enough for a response team to determine the intruder's identity and motive.
Risk	The level of impact on organizational operations (e.g., functions, image, or reputation), organizational assets, or individuals resulting from the operation of an information system given the potential impact of a threat and the likelihood of such threat occurring.
Security Policy	A set of criteria for the provision of security services. It defines and constrains the activities of a data processing facility in order to maintain a condition of security for systems and data.
Security Information and Event Management (SIEM)	Integrated information security oriented platform that offer the following services: Log management (log collection, storage, organization and retrieval); IT regulatory compliance (audit, validation or violation identification); Event correlation (normalization, fusion, verification, analysis); Active response (decision analysis, counter-measure response, prioritization); and Endpoint security (monitoring, updating, configuration).
Supervisory Control And Data Acquisition SCADA	Industrial control systems targeted to monitor and control infrastructure, industrial, and facility based processes, with the objective of supporting critical infrastructure protection processes.
Threat	A circumstance or event with the potential to adversely impact organizational operations (e.g., mission, functions, reputation), as well as organizational assets, individuals, other organizations, or even the Nation through unauthorized access, destruction, disclosure, modification of information, and/or denial of service.
Tiltmeter Compromise	The attacker takes control of the tiltmeter sensors and uses them to send false measurements to the monitoring station. In addition, the real status of the tilt of the dam's walls is hidden to the dam administrator. As a result, the dam's displacements and tilt trends are indefinitely hidden covering anomalous behaviours, preventing the monitoring system from raising alarms.
Time Value of Money	The idea that cost savings or revenue received today is more valuable than the same cost savings or revenue received some time in the future.

Trafficking Collection	An mWallet that is credited by many different entities with several loading means (e.g., credit card, bank transfer, mWallet transfer, payment intermediary, etc.) which is used by a fraudster to collect payments related to his/her trafficking.
Visualization Station Misuse	A station that is used to monitor the behavior of the system is able to send fake commands to the dam actuators or sensors.
Water Level Sensor Compromise	The attacker takes control of the water level sensors and uses them to send spoofed measurements to the monitoring station. The real status of the reservoir is hiden to the dam administrator, therefore, the dam can be overflown without alarms being raised by the monitoring system.
Zombies	A compromised computer connected to the Internet that is used to perform malicious tasks under remote control.

Bibliography

[AEG+10] P. Agarwal, A. Efrat, S. Ganjugunte, D. Hay, S. Sankararaman, and G. Zussman. Network Vulnerability to Single, Multiple and Probabilistic Physical Attacks. In *Military Communications Conference*, 2010.

[Aki10] O. Akindeinde. *Attack Simulation and Threat Modeling.* GNU Free Documentation License, 2010.

[And] T. Andreescu. Number Theory Trivia, Amicable Numbers. http://britton.disted.camosun.bc.ca/NumberTrivia.pdf.

[ANS10] ANSSI. EBIOS 2010 - Expression of Needs and Identification of Security Objectives. Website, 2010. `http://www.ssi.gouv.fr/en/the-anssi/publications-109/methods-to-achieve-iss/ebios-2010-expression-of-needs-and-identification-of-security-objectives.html`.

[ASM06] A. Abbas, A. El Saddik, and A. Miri. A Comprehensive Approach to Designing Internet Security Taxonomy. In *Canadian Conference on Electrical and Computer Engineering*, 2006.

[AV07] F. Amiel and K. Villegas. Passive and Active Combined Attacks-Combining Fault Attacks and Side Channel Analysis. In *Workshop on Fault Diagnosis and Tolerance in Cryptography*, 2007.

[BFP07] S. Bistarelli, F. Fioravanti, and P. Peretti. Using CP-nets as a Guide for Countermeasure Selection. In *ACM Symposium on Applied Computing*, page 300, 2007.

[BFPT08] S. Bistarelli, F. Fioravanti, P. Peretti, and I. Trubitsyna. Modeling and selecting countermeasures using CP-nets and Answer Set Programming. In *23rd Italian Convention of Computational Logic*, 2008.

[BGS+11] P. Bedi, V. Gandotra, A. Singhal, H. Narang, and S. Sharma. Optimal Countermeasures Identification Method: A New Approach in Secure Software Engineering. *European Journal of Scientific Research*, 55(4):527–537, 2011.

[BLFM05] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform Resource Identifier (URI): Generic Syntax. Technical report, RFC3986, 2005.

[BS12] V. Baumhof and A. Shipp. Zeus P2P Advancements and MitB Attack Vectors. http://threatmetrix.com/docs/ThreatMetrix-Labs-Report-July-2012.pdf, 2012.

[BSB07] J. Brocke, G. Strauch, and C. Buddendick. Return on Security Investment - Design Principles of Measurement System Based on Capital Budgeting. In *Proceedings of the 6th International Conference of Information Systems Technology and its Applications (ISTA)*, pages 21–32, 2007.

[BW00] L. Bernstein and J. Wild. *Analysis of Financial Statements.* McGraw-Hill, 2000.

[CCB07] F. Cuppens and N. Cuppens-Boulahia. Modelling Contextual Security Policies. *International Journal of Information Security*, 7(4):285–305, 2007.

[CCBM04] F. Cuppens, N. Cuppens-Boulahia, and A. Miege. Inheritance Hierarchies in the OrBAC Model and Application in a Network Environment. In *In the 2nd Foundation of Computer Security Workshop*, 2004.

[CCCR07] A. Cilardo, L. Coppolino, F. Campanile, and L. Romano. Adaptable Parsing of Real-Time Data Streams. In *15-th Euromicro Conference on Parallel, Distributed and Network-based Processing*, 2007.

[CDER09] L. Coppolino, S. D'Antonio, M. Esposito, and L. Romano. Exploiting Diversity and Correlation to Improve the Performance of Intrusion Detection Systems. In *International Conference on Network and Service Security*, 2009.

[CET+11] M. Cotton, L. Eggert, J. Touch, M. Westerlund, and S. Cheshire. Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry. Technical report, http://tools.ietf.org/html/rfc6335, 2011.

[CFGR10] C. Clavier, B. Feix, G. Gagnerot, and M. Roussellet. Passive and Active Combined Attacks on AES-Combining Fault Attacks and Side Channel Analysis. In *Workshop on Fault Diagnosis and Tolerance in Cryptography*, 2010.

[Clu10] Clusif. MEHARI 2010 - Risk Analysis and Treatment Guide. Website, 2010. `http://www.clusif.asso.fr/fr/production/ouvrages/pdf/MEHARI-2010-Risk-Analysis-and-Treatment-Guide.pdf`.

[CM05] M. Cremonini and P. Martini. Evaluating Information Security Investment from Attackers Perspective: the Return-on-Attack (roa). In *Proceedings of the 4th Workshop on the Economics on Information Security*, 2005.

[CMR04] H. Cavusoglu, B. Mishra, and S. Raghunathan. A Model for Evaluating It Security Investment. *Communications of the AMC*, 47(7):87–92, 2004.

[Con04] Lockstep Consulting. A Guide for Government Agencies Calculating ROSI. Technical report, http://lockstep.com.au/library/return_on_investment, 2004.

[Con11] The MASSIF Consortium. D2.1.1 Scenario Requirements. Technical report, MASSIF European Project from the 7th Framework Program, as part of the ICT MASSIF project (grant no. 257644), 2011.

[Cru08] F. K. Crundwell. *Finance for Engineers. Evaluation and Funding of Capital Projects.* Springer-Verlag London Limited, 2008.

[CVBH13] M. Cotton, L. Vegoda, R. Bonica, and B. Haberman. Special-Purpose IP Address Registries. Technical report, http://tools.ietf.org/html/rfc6890, 2013.

[CW08] A. Calder and S. Watkins. *IT Governance. A Manager's Guide to Data Security and ISO 27001/ISO 27002.* Kogan Page Limited, 2008.

[DCH06] C. Duan and J. Cleland-Huang. Automated Safeguard Selection Strategies. In *CTI Research Symposium*, 2006.

[DCMS06] P. Defibaugh-Chavez, S. Mukkamala, and A. Sung. Efficacy of Coordinated Distributed Multiple Attacks (A Proactive Approach to Cyber Defense). In *20th International Conference on Advanced Information Networking and Applications*, pages 10–14, 2006.

[DH95] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. Technical report, available at: http://tools.ietf.org/html/rfc1883, 1995.

[DP06] Y. Deswarte and D. Powell. Internet Security : An Intrusion-Tolerance Approach . *Proceedings of the IEEE, Special Issue on "Cryptography and Security"*, 92:432–441, 2006.

[DTCCB07] H. Débar, Y. Thomas, F. Cuppens, and N. Cuppens-Boulahia. Enabling Automated Threat Response through the Use of Dynamic Security Policy. *Journal in Computer Virology*, 3(3):195–210, 2007.

[EHP00] M. Evans, N. Hastings, and B. Peacock. *Triangular Distribution, Ch. 40 in Statistical Distributions, 3rd ed.* New York: Wiley, 2000.

[Ent05] Siemens Enterprise. The Logic behind CRAMM's Assessment of Measures of Risk and Determination of Appropriate Countermeasures, Technical report. Website, 2005. `http://www.cramm.com/downloads/techpapers.htm`.

[Fau06] T. Faucett. An introduction to ROC analysis. *Pattern Recognition Letters - Special issue: ROC analysis in pattern recognition*, 27(8):861–874, 2006.

[FCR09] M. Ficco, L. Coppolino, and L. Romano. A Weight-Based Symptom Correlation Approach to SQL Injection Attacks. In *Fourth Latin-American Symposium on Dependable Computing*, 2009.

[FGV11] J. Fan, B. Gierlichs, and F. Vercauteren. To Infinity and Beyond: Combined Attack on ECC Using Points of Low Order. In *13th International Conference on Cryptographic Hardware and Embedded Systems*, pages 143–159, 2011.

[Fis12] D. Fisher. Microsoft Releases Attack Surface Analizer Tool. Technical report, Available at: http://threatpost.com/en_us/blogs/microsoft-releases-attack-surface-analyzer-tool-080612, 2012.

[FS09a] Y. Ferenc and Y. Salim. A Game Theory Based Risk and Impact Analysis Method for Intrusion Defense Systems. In *International Conference on Computer Systems and Applications (AICCSA)*, pages 975–982, 2009.

[FS09b] P. Flajolet and R. Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009.

[Fuj10] J. Fujimoto. Speculative Attacks with Multiple Targets. Technical report, CARF Working Papers, available at: http://www.carf.e.u-tokyo.ac.jp/pdf/workingpaper/fseries/278.pdf, 2010.

[Ger07] J.L. Gersting. *Mathematical Structures for Computer Science. A Modern Approach to Descrete Mathematics*, volume 6. Freeman & Company, 2007.

[GHA+12] C. Gaber, B. Hemery, M. Achemlal, M. Pasquet, and P.Urien. A Synthetic Logs Generator for Fraud Detection in Mobile Transfer Services. In *In Financial cryptography*, 2012.

[GL02] L. Gordon and M. Loeb. Return on Information Security Investments: Myths vs. Realities. *Strategic Finance*, 84(5), 2002.

[GPGL11a] S. Gastéllier-Prevost, G. Gonzalez Granadillo, and M. Laurent. A Dual Approach to Detect Pharming Attacks at the Client-Side. In *4th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 2011.

[GPGL11b] S. Gastéllier-Prevost, G. Gonzalez Granadillo, and M. Laurent. Decisive Heuristics to Differentiate Legitimate from Phishing Sites. In *6th Conference on Network Architectures and Information Systems Security (SAR-SSI)*, 2011.

[Gri85] R. Grimaldit. *Discrete and Combinatorial Mathematics. An Applied Introduction.* Addison - Wesley Publishing Company, 1985.

[GS97] C. Grinstead and J. Laurie Snell. *Introduction to Probability, Second Revised Edition.* American Mathematical Society, 1997.

[Gud] K. Gudgion. McAfee Avert Labs Finding W32/Conficker.worm. McAfee.

[Hal] J. Halton. A Retrospective and Prospective Survey of the Monte Carlo Method. Technical report, SIAM Review.

[Har10] Houghton Mifflin Harcourt. *The American Heritage Science Dictionary.* Houghton Mifflin Harcourt Publishing Company, 2010.

[HB95] L. R. Halme and R. K. Bauer. Intrusion Detection FAQ: AINT Misbehaving: A Taxonomy of Anti-Intrusion Techniques. In *Proceedings of the 18th National Information Systems Security Conference*, pages 163–172, 1995.

[HDGKJ06] K. Tae Hyun, H. Dong-Guk, O. Katsujuki, and L. Jongin. Generic Cryptanalysis of Combined Countermeasures with Randomized BSD Representations. In *7th International Federation for Information Processing (IFIP), Smart Card Research and Advanced Applications*, pages 119–134, 2006.

[HH04] S. Hansman and R. Hunt. Ataxonomy of network and computer attacks. *Journal of Computers and Security*, 24(1):31–44, 2004.

[Hon] L. Hongkiat. Five Notorious Facebook Attacks (Learn How To Protect Yourself). Website.

[How04] M. Howard. Mitigate Security Risks by Minimizing the Code You Expose to Untrusted Users. In *MSDN Magazine available at: http://msdn.microsoft.com/en-us/magazine/cc163882.aspx*, 2004.

[HTRM10] D. Harwood, D. Torbic, K. Richard, and M. Meyer. SafetyAnalyst: Software Tools for Safety Management of Specific Highway Sites. Technical report, Federal Highway Administration Publication No. FHWA-HRT-10-063, 2010.

[HW07] M. Howard and J. Wing. Measuring Relative Attack Surfaces. In *In Computer Security in the 21st Century*, pages 109–137, 2007.

[IL99] C. Irvine and T. Levin. Toward a Taxonomy and Costing Method for Security Services. In *Computer Security Applications Conference*, pages 163–168, 1999.

[Jef04] M. Jeffrey. *Return on Investment Analysis for e-Business Projects*, volume 3. Internet Encyclopedia, First Edition, Hossein Bidgoli Editor, 2004.

[JTT10] W. Jack, S. Tavneet, and R. Townsed. Monetary Theory and Electronic Money: Reflections on the Kenyan Experience. Technical report, Federal Reserve Bank of Richmond, 2010.

[JW04] I. Jacobs and N. Walsh. Architecture of the World Wide Web, Volume One. Technical report, W3C Recommendation, 2004.

[KBB+03] A. Abou El Kalam, R. El Baida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Miege, C. Saurel, and G. Trouessin. Organization Based Access Control. In *8th International Workshop on Policies for Distributed Systems and Networks*, 2003.

[KCBCD10] N. Kheir, N. Cuppens-Boulahia, F. Cuppens, and H. Débar. A Service Dependency Model for Cost-Sensitive Intrusion Response. In *Proceedings of the 15th European Symposium on Research in Computer Security (ESORICS)*, pages 626–642, 2010.

[Khe10] N. Kheir. *Response Policies and Countermeasures: Management of Service Dependencies and Intrusion and Reaction Impacts.* PhD thesis, Ecole Nationale Supérieure des Télécommunications de Bretagne, 2010.

[Kis11] R. Kissel. *Glossary of Key Information Security Terms.* National Institute of Standards and Technology. U.S. Department of Commerce, 2011.

[KLI08] D. Kim, T. Lee, and H. In. Effective Security Safeguard Selection Process for Return on Security Investment. In *Proceedings of the IEEE Asia-Pacific Services Computing Conference*, pages 21–32, 2008.

[KLK05] A. Kim, J. Luo, and M. Kang. Security Ontology for Annotating Resources. In *Research Lab, NRL Memorandum Report*, pages 1483–1499, 2005.

[Kos11] D. Kosutic. Is it possible to calculate the Return on Security Investment (rosi)? In *http://blog.iso27001standard.com/2011/06/13/is-it-possible-to-calculate-the-return-on-security-investment-rosi/*, 2011.

[Kri09] A. Kriegisch. Detecting Conficker in your Network. http://www.cert.at/static/downloads/papers/TR_Conficker_Detection.pdf, 2009.

[KRZ+98] M. Kaaniche, L. Romano, Z.Kalbarczyck, R. Iyer, and R. Karcich. A Hierarchical Approach for Dependability Analysis of a Cache-based RAID Storage Architecture. *28th IEEE International Symposium on Fault-Tolerant Computing (FTCS'28)*, pages 6–15, 1998.

[Kum95] S. Kumar. *A Classification and Detection of Computer Intrusion.* PhD thesis, Purdue University, 1995.

[LJ97] U. Lindqvist and E. Jonsson. How to Systematically Classify Computer Security Intrusions. In *In IEEE Security and Privacy*, pages 154–163, 1997.

[Loc05] C. Locher. Methodologies for Evaluating Information Security Investments - What Basel II Can Change in the Financial Industry. In *ECIS Proceedings*, 2005.

[Lou01] D. Lough. *A Taxonomy of Computer Attacks with Applications to Wireless Networks.* PhD thesis, Virginia Polytechnic Institute and State University, 2001.

[LSY11] Y. Liu, Y. Sun, and T. Yu. Defending Multiple-user-multiple-target Attacks in Online Reputation Systems. In *International Conference on privacy, Security, Risk and Trust, and International Conference on Social Computing*, pages 425–434, 2011.

[LW09] F. Leder and T. Werner. Know Your Enemy: Containing Conficker To Tame A Malware. http://www.honeynet.org/files/KYE-Conficker.pdf, 2009.

[Man08] P. Manadhata. *An Attack Surface Metric*. PhD thesis, School of Computer Science Carnegie Mellon University, 2008.

[Mar01] R. Martin. Managing Vulnerabilities in Networked Systems. *Journal Computer*, 34(11):32–38, 2001.

[MAZP99] L. Masinter, H. Alvestrand, D. Zigmond, and R. Petke. Guidelines for new URL Schemes. Technical report, RFC2718, 1999.

[MBD13] Y. Ben Mustapha, G. Blanc, and H. Debar. Policy Enforcement Points Responsibility Domain. In *Under reviw*, 2013.

[MHH+10] D. Miller, S. Harris, A. Harper, S. Van Dyke, and C. Blask. *Security Information and Event Management (SIEM) Implementation*. Mc Graw Hill, 2010.

[Mic12] SDL Team Microsoft. Attack Surface Analyzer 1.0. Website, 2012. `http://blogs.msdn.com/b/sdl/archive/2012/08/02/attack-surface-analyzer-1-0-released.aspx`.

[Mie05] A. Miege. *Definition of a Formal Framework for Specifying Security Policies. The OrBAC Model and Extensions*. PhD thesis, Ecole Nationale Supérieure des Télécommunications de Paris, 2005.

[Miz05] A. Mizzi. Return on Information Security Investment - Are you spending enough? Are you spending too much? In *http://www.adrianmizzi.com/ROISI-Paper.pdf*, 2005.

[Miz10] A. Mizzi. Return on Information Security Investment - The Viability Of An Anti-Spam Solution In A Wireless Environment. *International Journal of Network Security*, 10(1):18–24, 2010.

[MKBS07] S. Mukkamala, K.Yendrapalli, R. Basnet, and A. Sung. Detecting Coordinated Distributed Multiple Attacks. In *21st International Conference on Advanced Information Networking and Applications Workshops*, volume 01, pages 557–562, 2007.

[MKW08] P. Manadhata, Y. Karabulut, and J. Wing. Measuring the Attack Surfaces of SAP Business Applications. In *IEEE International Symposium on Software Reliability Engineering*, 2008.

[ML] A. Morris and W. Lopez. Thabit Number (version 2). http://planetmath.org/encyclopedia/ThabitNumber.html.

[MR04] J. Mirkovic and P. Reiher. A Taxonomy of DDoS Attack and DDoS Defense Mechanisms. *ACM SIGCOMM Computer Communication Review*, 34:39–53, 2004.

[MW10] P. Manadhata and J. Wing. An Attack Surface Metric. In *IEEE Transactions on Software Engineering*, 2010.

[MWFM06] P. Manadhata, J. Wing, M. Flynn, and M. McQueen. Measuring the Attack Surfaces of Two FTP Daemons. In *2nd ACM Workshop on Quality of Protection*, pages 3–10, 2006.

[NLSO11] D. Nakatsu, Y. Li, K. Sakijama, and K. Ohta. Combination of SW Countermeasure and CPU Modification on FPGA against Power Analysis. In *11th International Conference on Information Security Applications (WISA)*, pages 258–272, 2011.

[Nor10] T. Norman. *Risk Analysis and Security Countermeasure Selection.* CRC Press Taylor & Francis Group, 2010.

[NP10] T. Neubauer and M. Pehn. Workshop-based Security Safeguard Selection with AURUM. *International Journal on Advances in Security*, 3(3-4):123–134, 2010.

[NSW06] T. Neubauer, C. Stummer, and E. Weippl. Workshop-based Multiobjective Security Safeguard Selection. In *First International Conference on Availability, Reliability and Security (ARES)*, page 1, 2006.

[oAS91] Federation of American Scientists. Special Operations Forces Intelligence and Electronic Warfare Operations. Technical report, Appendix D: Target Analysis Process, Available at: http://www.fas.org/irp/doddir/army/fm34-36/toc.htm, 1991.

[Olo05] P. Olofsson. *Probability, Statistics, and Stochastic Processes.* John Wiley & Sons, Inc, 2005.

[Org08] International Standard Organization. International Standard ISO/IEC 27005: Information Technology - Security Techniques - Information Security Risk Management, 2008.

[oSC80] Information Sciences Institute University of Southern California. DOD Standard Internet Protocol. Technical report, available at: http://tools.ietf.org/html/rfc760, 1980.

[oST70] National Institute of Standards and Technologies. Guide for Conducting Risk Assessment. Website, 1970. `http://csrc.nist.gov/publications/nistpubs/800-30-rev1/sp800_30_r1.pdf`.

[Pis10] D. Piscitello. Conficker Summary and Review. http://www.icann.org/en/news/announcements/announcement-11may10-en.htm, 2010.

[PKTT11] S. Petajasoja, H. Kortti, A. Takanen, and J. Tirila. IMS Threat and Attack Surface Analysis using Common Vulnerability Scoring System. In *35th IEEE Annual Computer Software and Applications Conference Workshops*, 2011.

[Pua09] P. Puangsri. Quantified Return On Information Security Investment - A Model for Cost-Benefit Analysis. Master's thesis, Delft University of Technology, 2009.

[RDFC] L. Romano, S. D'Antonio, V. Formicola, and L. Coppolino. Protecting the WSN Zones of a Critical Infrastructure via Enhanced SIEM Technology. In *International Conference on Computer Safety, Reliability, and Security*, pages 222–234.

[RGF+10] R. Riveiro, E. Galvao, F. Freitas, C. Rodriguez, M. Siqueira, W. Campos, and R. Santos. An Automatic Ontology-Based Multiagent System for Intrusion Detection in Computing Environments. *International Journal for Informatics (IJI)*, 3(1), 2010.

[RH98] M.H. Rheinfurth and L.W. Howell. Probability and Statistics in Aerospace Engineering. Technical report, NASA Center for AeroSpace Information, 1998.

[Ros94] K. Rosen. *Discrete Mathematics and its Applications.* McGraw Hill, 1994.

[RS12] L. Rajbhandari and E. Snekkenes. Intended Actions: Risk Is Conflicting Incentives. In *15th International Conference ISC*, volume 7483, pages 370–386, 2012.

[RS13] L. Rajbhandari and E. Snekkenes. Using the Conflicting Incentives Risk Analysis Method. In *28th IFIP International Conference, Security and Privacy Protection in Information System*, pages 315–329, 2013.

[Saa93] T. L. Saaty. What is relative measurement? The ratio scale phantom. *Mathematical and Computer Modelling Journal*, 17(4-5):1–12, 1993.

[SAS06] W. Sonnenreich, J. Albanese, and B. Stout. Return On Security Investment (rosi) - A Practical Quantitative Model. *Journal of Research and Practice in Information Technology*, 38(1), 2006.

[SBC+07] P. Sousa, A. Neves Bessani, M. Correia, N. Ferreira Neves, and P. Verissimo. Resilient Intrusion Tolerance through Proactive and Reactive Recovery, Technical report. Website, 2007. `http://www.di.fc.ul.pt/tech-reports`.

[SBW07a] N. Stakhanova, S. Basu, and J. Wong. A Cost-Sensitive Model for Preemptive Intrusion Response Systems. In *Proceedings of the 21st International Conference on Advanced Networking and Applications*, 2007.

[SBW07b] N. Stakhanova, S. Basu, and J. Wong. A Taxonomy of Intrusion Response Systems. *International Journal of Information and Computer Security*, 1(1/2):169–184, 2007.

[Sch03] M. Schumacher. *Security engineering with patterns: origins, theoretical model, and new applications*. Springer-Verlag New York Inc, 2003.

[Sch04] S. E. Schechter. *Computer Security Strength & Risk: A Quantitative Approach*. PhD thesis, Harvard University, 2004.

[Sch11] M. Schmidt. Return on Investment (ROI): Meaning and Use. In *Encyclopedia of Business Terms and Methods*. http://www.solutionmatrix.com/return-on-investment.html, 2011.

[Seg13] J. Segura. Zero-Day Java vulnerability wreaks havoc on computers worldwide. In *http://blog.malwarebytes.org/intelligence/2013/01/zero-day-java-vulnerability-wreaks-havoc-on-computers-worldwide/*, 2013.

[SND09] A. Saidane, V. Nicomette, and Y. Deswarte. The Design of a Generic Intrusion Tolerant Architecture for Web Servers. *IEEE Transactions on Dependable and Secure Computing*, 6:45–58, 2009.

[Sof09] GFI Software. Targeted Cyber Attacks, the danger faced by your corporate network. Technical report, White paper available at: http://www.gfi.com/whitepapers/cyber-attacks.pdf, 2009.

[TAAA12] A. Talib, R. Atan, R. Abdullah, and M. Azmi. Security Ontology Driven Multi Agent System Architecture for Cloud Data Storage Security: Ontology Development. *International Journal of Computer Science and Network Security*, 12(5):63–72, 2012.

[Tho07] Y. Thomas. *Policy-Based Response to Intrusions Through Context Activation*. PhD thesis, Ecole Nationale Supérieure des Télécommunications de Bretagne, 2007.

[TK00] H. F. Tipton and M. Krause. *Information Security Management Handbook, Four Volume Set*. Boca Raton Auerbach Publications, 2000.

[TKL+13] J. Touch, M. Kojo, E. Lear, A. Mankin, K. Ono, M. Stiemerling, and L. Eggert. Service Name and Transport Protocol Port Number Registry. Technical report, http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml, 2013.

[Tou13] J. Touch. Updated Specification of the IPv4 ID Field. Technical report, available at: http://tools.ietf.org/html/rfc6864, 2013.

[VE03] H. Venter and J. Eloff. A Taxonomy for Information Security Technologies. *Computers and Security*, 22:299–307, 2003.

[VF10] E. Vetillard and A. Ferrari. Combined Attacks and Countermeasures. In *International Federation for Information Processing*, 2010.

[VNC03] P. E. Verissimo, N. Ferreira Neves, and M. Pupo Correia. Intrusion-Tolerant Architectures: Concepts and Design. *Lecture Notes in Computer Science, Architecting Dependable Systems*, 32677:3–36, 2003.

[VNC+06] P. Verissimo, N. Neves, C. Cachin, D. Powell, Y. Deswarte, R. Stroud, and I. Welch. Intrusion-Tolerant Middleware: the Road to Automatic Security. *IEEE Security & Privacy*, 4:54–62, 2006.

[Wan04] S. Waner. Probability and Statistics. http://people.hofstra.edu/Stefan_ Waner/RealWorld/pdfs/241Notes.pdf, 2004.

[War07] K. Ward. Series Binomial Theorem. http://www.trans4mind.com/personal_ development/mathematics/series/binomialTheorem.htm, 2007.

[WSS+00] A. Westerinen, J. Schnizlein, J. Strassner, M. Scherling, B. Quinn, S. Herzog, A. Huynh, M. Carlson, J. Perry, and S. Waldbusser. Terminology for Policy-Based Management. Technical report, RFC3198, 2000.

[WW03] H. Wang and C. Wang. Taxonomy of Security Considerations and Software Quality. *Communications of the ACM*, 46(6):75–78, 2003.

[YPG00] R. Yavatkar, D. Pendarakis, and R. Guerin. A Framework for Policy-based Admission Control. Technical report, RFC2753, 2000.

[ZKSY09] A. Zonouz, H. Khurana, W. Sanders, and T. Yardley. A Game-Theoretic Intrusion Response and Recovery Engine. In *International Conference on Dependable Systems and Networks*, 2009.

Author's publications

Publications in international peer-reviewed journals

[GMHD12] G. Gonzalez Granadillo, Y. Ben Mustapha, N. Hachem, and H. Débar. An Ontology-driven Approach to Model SIEM Information and Operations Using the SWRL Formalism. *International Journal of Electronic Security and Digital Forensics*, 4(2/3):104–123, 2012.

[GBDJ13] G. Gonzalez Granadillo, M. Belhaouane, H. Débar, and G. Jacob. RORI-based Countermeasure Selection using the OrBAC Formalism. *International Journal of Information Security*, 2013.

[GGF13] G. Gonzalez Granadillo, C. Gaber, and V. Formicola. Enhanced Detection and Reaction at the Edge of SIEM Systems for a Mobile Money Transfer Service Infrastructure. *Journal of Network and Systems Management (in review)*, 2013.

Publications in international peer-reviewed conferences

[GMHD11] G. Gonzalez Granadillo, Y. Ben Mustapha, N. Hachem, and H. Débar. An Ontology-based Model for SIEM Environments. In *7th International Conference in Global Security, Safety and Sustainability*, volume 99, pages 148–155, 2011.

[GDJ+12] G. Gonzalez Granadillo, H. Débar, G. Jacob, C. Gaber, and M. Achemlal. Individual Countermeasure Selection based on the Return On Response Investment Index. In *International Conference Mathematical Methods, Models and Architectures for Computer Network Security*, pages 156–170, 2012.

[GDJC12] G. Gonzalez Granadillo, H. Débar, G. Jacob, and L. Coppolino. Combination Approach to Select Optimal Countermeasures based on the RORI Index. In *Second International Conference on the Innovative Computing Technology*, pages 38–45, 2012.

Contributions to European projects

[Con11b] The MASSIF Consortium. D3.2.2 Specification of Event and Alert Description. Technical report, MASSIF European Project from the 7th Framework Program, as part of the ICT MASSIF project (grant no. 257644), 2011.

[Con11c] The MASSIF Consortium. D5.2.1 Decision Support, Simulation, and Deployment Software Components. Technical report, MASSIF European Project from the 7th Framework Program, as part of the ICT MASSIF project (grant no. 257644), 2012.

[Con12a] The MASSIF Consortium. D5.2.2 Preliminary Software Implementation for Decision Support, Simulation, and Deployment Software Components. Technical report, MASSIF European Project from the 7th Framework Program, as part of the ICT MASSIF project (grant no. 257644), 2013.

[Con12b] The MASSIF Consortium. D5.2.3 Final Software Implementation for Decision Support, Simulation, and Deployment Software Components. Technical report, MASSIF European Project from the 7th Framework Program, as part of the ICT MASSIF project (grant no. 257644), 2013.

Part III

Appendixes

Appendix A

French Summary

Optimisation de la Réponse aux menaces basée sur les coûts dans des systèmes pour la Sécurité de l'Information et la Gestion des Evénements (SIEMs)

Les SIEMs (systèmes pour la Sécurité de l'Information et la Gestion des Evénements) sont le coeur des centres opérationnels de sécurité actuels. Les SIEMs corrèlent les événements en provenance de différents capteurs (anti-virus, pare-feux, systèmes de détection d'intrusion, etc), et offrent des vues synthétiques pour la gestion des menaces ainsi que des rapports de sécurité.

La recherche dans les technologies SIEM a toujours mis l'accent sur la fourniture d'une interprétation complète des menaces, en particulier pour évaluer leur importance et hiérarchiser les réponses. Toutefois, dans de nombreux cas, la réponse des menaces a encore besoin de l'homme pour mener l'analyse et aboutir à la prise de décisions, p.ex. compréhension des menaces, définition des contremesures appropriées ainsi que leur déploiement. Il s'agit d'un processus lent et coûteux, nécessitant un haut niveau d'expertise, qui reste néanmoins sujet à erreurs. Ainsi, des recherches récentes sur les SIEMs ont mis l'accent sur l'importance et la capacité d'automatiser le processus de sélection et le déploiement des contremesures.

Certains auteurs [DTCCB07, RGF+10] ont proposé des mécanismes automatiques de réponse, comme l'adaptation des politiques de sécurité pour dépasser les limites de réponses statiques ou manuelles. Bien que ces approches améliorent le processus de réaction (en le rendant plus rapide et/ou plus efficace), ils restent limités car ces solutions n'analysent pas l'impact des contremesures choisies pour atténuer les attaques.

Dans cette thèse, nous proposons une nouvelle approche systématique qui sélectionne la contremesure optimale au travers d'un ensemble de candidats, classés sur la base d'une comparaison entre leur efficacité à arrêter l'attaque et leur capacité à préserver, simultanément, le meilleur service aux utilisateurs légitimes.

Afin d'analyser quantitativement l'impact de l'attaque, notre modèle prend en compte deux aspects de la politique de sécurité liées aux réponses des menaces: premièrement, le changement cumulé à long terme des politiques de sécurité suite à des attaques antérieures et, deuxièmement, le fait que les politiques de sécurité peuvent nécessiter d' être automatiquement adaptées au contexte actuel.

Prenant en compte les précédents modèles quantitatifs [Jef04, Sch11, CM05, BSB07, SAS06, SBW07a, KLI08, KCBCD10, Khe10], cette thèse propose un modèle pour sélectionner la contremesure qui offre le plus haut niveau de bénéfice à l'organisation. Nous adaptons la proposition faite dans [KCBCD10, Khe10] pour l'utilisation des métriques de coûts visant à évaluer l'impact

de chaque contremesure, ce qui permet au système de sélectionner celle qui fournit l'indice RORI maximal.

A.1 État de l'art

Plusieurs auteurs ont proposé des métriques de coûts pour équilibrer dommages d'intrusion et coûts d'intervention tout en garantissant le choix de la réponse la plus appropriée sans sacrifier les fonctionnalités du système. Un résumé de ces modèles est présenté dans le Tableau A.1.

L'approche la plus simple et la plus utilisée pour évaluer les conséquences financières des investissements dans les entreprises est le retour sur investissement (ROI). Le ROI compare le bénéfice par rapport aux coûts obtenus d'investissement donnés [Jef04, Sch11].

Le retour sur l'attaque (ROA) a été défini comme le gain espéré par l'attaquant pour une attaque réussie face aux pertes qu'il subit suite à l'adoption de mesures de sécurité par sa cible [CM05]. Les auteurs affirment par ailleurs que l'efficacité des investissements dans les technologies de sécurité pourrait se dégrader en raison de changements de contexte, sans impact sur l'indice ROI. En outre, le retour sur investissement seul ne peut pas prévoir les différents impacts que les solutions ont sur les comportements de l'attaquant.

Le retour sur investissement de la sécurité (ROSI) [BSB07, SAS06], a été proposée comme une métrique qui compare les différences entre dommages causés par les incidents de sécurité (avec ou sans contremesures) et coût de la solution. Les auteurs conviennent que même en ayant des résultats inexacts, si la méthode produit des résultats cohérents et répétitifs, le modèle peut être utile pour comparer les solutions de sécurité sur la base de valeurs relatives.

Plus récemment, le retour sur Investissement de la Réponse (RORI) [KCBCD10, Khe10] a été introduit comme un modèle de dépendance de service pour la réponse des coûts sensibles, en fonction d'une comparaison financière des options de réponse. L'indice RORI considère non seulement les dommages collatéraux de réponse, mais aussi les effets sur la réponse des intrusions.

TABLE A.1 - Résumé des modèles sensibles aux coûts

Modèles	**ROI** [Jef04, Sch11]	**ROA** [CM05]	**ROSI** [BSB07, SAS06, SBW07a, KLI08]	**RORI** [KCBCD10, Khe10]
Objectif Principal	Coût de solutions de sécurité (SecCost), Efficacité	Gain de l'attaque (Att_Gain), Coût de l'Attack (Att_Cost), Pertes dues à la sécurité (SecLoss)	Investissement de sécurité (SecInv), Bénéfices et Coûts	Dommage collatéral(CD), Coûts Opérationnels (OC) et Coûts de Réponse (RC)
Formule	$\frac{Benefit-SecCost}{SecCost}$	$\frac{Att\_Gain}{Att\_Cost+SecLoss}$	$\frac{Returns-SecInv}{SecInv}$	$\frac{Losses-RC-OC}{CD+OC}$
Solution Optimale	La valeur ROI la plus élevée	La valeur ROA la plus basse	La valeur ROSI la plus élevée	La valeur RORI la plus élevée
Caractéristiques	Évaluer les conséquences financières de l'investissement privé	Évaluer l'impact des solutions de sécurité basées sur le comportement de l'attaque	Comparer la différence entre les dommages informatiques (avec ou sans contremesures) contre le coût de la solution	Déterminer le bénéfice obtenu dans un scénario de menace particulière qui applique une contremesure donnée
Contraintes	- Il ne peut pas être utilisé pour évaluer le fait de ne rien faire - Impossible d'évaluer l'impact de la solution en raison du comportement de l'attaquant - Il ne tient pas compte des dommages collatéraux ni les coûts d'exploitation	- Difficile d'être précis tout en prédisant le comportement de l'attaquant - Il ne tient pas compte des coûts liées à la solution de sécurité - Il ne peut pas être utilisé pour évaluer le fait de ne rien faire	- Il ne tient pas compte des dommages collatéraux ni les coûts d'exploitation - Il ne peut pas être utilisé pour évaluer le fait de ne rien faire - Impossible d'évaluer l'impact de la solution en raison du comportement de l'attaquant	- Il ne considère pas le comportement de l'attaquant - Il ne peut pas être utilisé pour évaluer le fait de ne rien faire - Il n'est pas normalisé à la taille de l'infrastructure

A.2 Modèle de Sélection de Contremesures

Notre solution considère l'approche proposée par Kheir et al. [KCBCD10, Khe10], où les auteurs définissent le Retour Sur l'Investissement de Réponse (RORI) par la formule représentée dans l'Équation A.1.

$$RORI = \frac{[ICb - RC] - OC}{CD + OC} \tag{A.1}$$

où,

- ICb est l'impact d'intrusion attendu en l'absence des mesures de sécurité. Il mesure le coût des dommages du aux intrusions ou attaques.
- RC est l'impact combiné tant pour l'intrusion que pour la réponse. RC représente la somme des impacts d'intrusion attendus après mise en place d'une réponse et coût engendré par cette réponse.
- OC est le coût opérationnel qui inclut les coûts de déploiement et de mise en place de réponses, tel que la main d'oeuvre et le sur-provisionnement.
- Le CD est le dommage de réponse collatérale qui représente le coût engendré par la mesure de sécurité.

Le déploiement de l'indice du Retour sur l'investissement de réponse (RORI) dans des scénarios réels a mis en évidence les limitations suivantes :

- La valeur absolue de paramètres comme l'ICb et le RC est difficile à estimer. Néanmoins, un indice de ces paramètres est plus facile à déterminer, ce qui réduit à son tour des erreurs de magnitude.
- L'indice RORI n'est pas défini lorsqu'aucune contremesure n'est choisie. Comme le coût opérationnel (OC) est associé à la mesure de sécurité, l'indice RORI donnera lieu à une indétermination lorsqu'aucune solution n'est choisie.
- L'indice RORI n'est pas normalisé par rapport à la largeur et la complexité de l'infrastructure d'une organisation.

A.2.1 RORI Amélioré

Nous proposons une amélioration de l'indice RORI en tenant compte le coût de la contremesure et son niveau de mitigation associée ainsi que la valeur de l'infrastructure et les pertes qui peuvent se produire comme conséquence d'une intrusion ou d'une attaque. L'indice RORI amélioré considère le choix de n'est pas mettre en place de contremesure, et le compare avec les résultats obtenus par la mise en oeuvre de solutions de sécurité (individuelles ou combinées). L'indice amélioré du retour sur l'investissement de réponse (RORI) fournit également une réponse relative à la taille de l'infrastructure. Son calcul utilise les paramètres représentés dans l'Equation A.2.

$$RORI = \frac{(ALE \times RM) - ARC}{ARC + AIV} \times 100 \tag{A.2}$$

où,

- ALE (Perte annuelle attendue) est le coût de l'impact obtenu en l'absence de mesures de sécurité. ALE est exprimée en devises par an (p.ex. \$/an) et dépendra directement de la sévérité de l'attaque et de sa probabilité d'occurrence.

- RM désigne le niveau d'atténuation du risque associé à une solution particulière. Il prend une valeur comprise entre 0 et 100. En l'absence de contremesures, RM est égal à 0%.
- ARC est le coût annuel de réponse qui est engagé par la mise en oeuvre d'une nouvelle politique de sécurité. De l'équation A.1, ARC = OC + CD. ARC est toujours supérieur ou égal à 0, et il est exprimé en devise par an (p.ex. $/an)
- AIV est la valeur annuelle de l'infrastructure (p.ex. le coût du matériel, des services pour les opérations régulières, etc) que l'on attend du système, quelles que soient les contremesures mises en oeuvre. ARC est supérieur à zéro (AIV $>$ 0), et il est exprimé en devise par an (p.ex. $/an).

A.2.2 Améliorations

- Les paramètres '$ICB - RC$' sont substitués par '$ALE \times RM$' afin dévaluer plus facilement la réponse de solutions individuelles et combinées, tout en réduisant l'ampleur d'erreur.
- Le paramètre AIV fournit une réponse par rapport à la taille de l'infrastructure. AIV est corrélé à la perte annuelle attendue (ALE) du système, et permet de comparer les résultats RORI des différents systèmes quelles que soient leurs tailles.
- L'introduction du paramètre de valeur annuelle de l'infrastructure (AIV) permet de gérer le cas où aucune contremesure n'est sélectionnée, ce qui se traduit par la valeur 0, et signifie qu'aucun gain n'est prévu si aucune solution n'est mise en oeuvre.

A.2.3 Analyse de Sensibilité

RORI est un indice relatif indiquant le pourcentage de bénéfice perçus si une contremesure est mise en oeuvre. Lors de l'analyse de l'investissement dans la sécurité de l'information, il ne faut pas s'attendre à une augmentation des profits, mais plutôt au contraire, à une atténuation des risques auxquels l'organisation est exposée.

L'indice RORI varie de $\frac{-ARC}{ARC+AIV}$ (dans sa limite inférieure) jusqu'à $\frac{ALE}{AIV}$ (dans sa limite supérieure). Un RORI positif signifie que nous nous attendons à diminuer le risque jusqu'à un certain niveau et qu'il est donc approprié d'appliquer la solution de sécurité. p.ex. un RORI de 50% signifie que nous nous attendons à atténuer la moitié du risque auquel l'organisation est exposée. Toutefois, lors de l'évaluation de l'option de ne rien faire (pas de contremesure évaluée), il faut s'attendre à 0% de mitigation du risque.

Le pire scénario (le coût de la contremesure est supérieur aux avantages qu'elle procure) aura $ALE \times RM << ARC$, donc $RORI \rightarrow \frac{-ARC}{ARC+AIV}$. Le meilleur scénario (mitigation parfaite) aura RM = 1, ARC = 0, donc $RORI = \frac{ALE}{AIV}$. Si le bénéfice attendu est égal au coût de la contremesure, RORI s'approchera à zéro. Toutefois, si le bénéfice attendu est inférieur au coût de la contremesure, RORI atteindra une valeur négative. Ce n'est que dans les cas où le bénéfice est supérieur au coût de la mise en oeuvre d'une mesure de sécurité que RORI atteindra une valeur positive.

Afin d'évaluer les effets sur les résultats RORI, nous avons effectué une série d'analyses de sensibilité où deux variables ont été modifiées, tandis que les autres ont gardé leurs valeurs du scénario de référence. Les résultats obtenus sont décrits comme suit:

- Si l'ARC est des ordres de magnitude en dessous de l'AIV ($ARC \ll AIV$), l'impact de l'ARC sur RORI est très faible. Dans ce cas, $ARC + AIV \cong AIV$, donc $RORI \cong \left(\frac{ALE \times RM}{AIV}\right)$. Toutefois, si l'ARC est des ordres de magnitude au-dessus de l'AIV ($ARC \gg AIV$), l'impact de l'ARC sur l'indice RORI est très forte. Dans ce cas, $ARC + AIV \cong ARC$, donc $RORI \cong \frac{(ALE \times RM) - ARC}{ARC}$.

- Si l'ALE est des ordres de magnitude en dessous de l'AIV ($ALE \ll AIV$), l'ALE impact négativement l'indice RORI, car $ALE \times RM \cong 0$, donc $RORI \cong \frac{-ARC}{ARC+AIV}$. Toutefois, si l'ALE est des ordres de magnitude au-dessus de l'AIV ($ALE \gg AIV$), l'indice RORI est positivement impacté. Dans ce cas, $AIV \cong 0$, donc $RORI \cong \frac{(ALE \times RM)-ARC}{ARC}$.
- Si l'ALE est des ordres de magnitude en dessous de l'ARC ($ALE \ll ARC$), l'ALE impact négativement l'indice RORI, car $ALE \cong 0$, donc $RORI \cong \frac{-ARC}{ARC+AIV}$. Toutefois, si l'ALE est des ordres de magnitude au-dessus de l'ARC ($ALE \gg ARC$), l'indice RORI est positivement impacté. Dans ce cas, l'$ARC \cong 0$, donc $RORI \cong \frac{ALE \times RM}{AIV}$.
- Si RM augmente par rapport à l'AIV, l'ALE et l'ARC, l'indice RORI dépendra de la valeur de l'ALE. Dans ce cas, $ALE \times RM \cong ALE$, donc $RORI \cong \frac{ALE-ARC}{ARC+AIV}$, ce qui rend la solution plus attractive au mesure que l'ALE augmente.

A.2.4 Limitations restantes

L'évaluation et la sélection des contremesures dépendent de l'estimation appropriée de la valeur de l'infrastructure, de l'impact des attaques et de la définition des politiques de sécurité nécessaires pour atténuer l'attaque. Une telle définition devrait inclure les coûts et les bénéfices associés à une contremesure particulière dans un scénario d'attaque donnée.

La principale limitation du mod'ele RORI est la précision dans l'estimation des différents paramètres qui composent la formule. L'estimation de la perte annuelle attendue (ALE) d'une attaque qui se produirait sur un système donné, et le niveau d'atténuation des risques (RM) d'une contremesure particulière est difficile à déduire et nécessite un effort important. Une mesure précise de ces deux paramètres est quasi-impossible car elle nécessite des prédictions d'un événement qui n'a pas encore eu lieu. Cependant, une estimation qualitative de la RM et ALE reste possible, ce qui fait de l'évaluation RORI un outil intéressant dans l'aide à la décision pour les analystes de sécurité.

Le modèle RORI présenté dans ce document ne considère pas l'interdépendance entre contremesures (c.-à-d. comment l'application d'une contremesure peut affecter l'efficacité des autres), mais il décrit les restrictions et/ou les confits qui peuvent se produire lors de la mise en oeuvre de la contremesure choisie (p.ex. contremesures partiellement ou totalement restrictives).

Finalement, le modèle évalue les contremesures individuelles et combinées pour attenuer les effets d'une attaque donnée, mais il ne tient pas compte de l'évaluation et de la sélection des contremesures dans un scénario d'attaques multiples.

A.3 Processus de Sélection de Contremesures Individuelles

Le processus de sélection des contremesures optimales est effectuée en deux étapes: le calcul RORI et l'évaluation de contremesures. Cette section détaille chaque partie du modèle.

A.3.1 Calcul du RORI

Le retour sur l'investissement de Réponse proposé dans la section A.2.1 est utilisé comme une approche quantitative pour évaluer et classer un ensemble de contremesures, ce qui permet de choisir celle qui convient le mieux pour atténuer les effets d'une attaque donnée. Les paramètres d'entrée pour le calcul RORI sont de deux types: des paramètres invariables (p.ex. ALE, AIV), qui dépendent de l'intrusion ou de l'attaque, et des paramètres variables (p.ex. RM, ARC), qui dépendent de la contremesure.

A.3.1.1 Paramètres Invariables

A.3.1.1.1 La Perte Annuelle Estimée (ALE): désigne le coût d'impact obtenue en l'absence des mesures de sécurité. Il inclut la perte de biens (La), la perte de données (Ld), la perte de la réputation (Lr), les procédures juridiques (Lp), la perte de chiffre d'affaires avec des clients existants (LREC), la perte de revenus provenant des clients potentiels (LRPC), d'autres pertes (Ol), le contrat d'assurance (Ci), et le taux annuel d'occurrence (ARO), comme le montre l'Équation A.3.

$$ALE = (La + Ld + Lr + Lp + Lrc + Ol - Ci) \times ARO \tag{A.3}$$

où,

- La Perte de biens (La), se réfère à la valeur des actifs physiques qui seraient affecté par une intrusion ou attaque (p.ex. le matériel, le mobilier, les infrastructures physiques, etc).
- La Perte de données (Ld), se réfère à la valeur des actifs non-physiques qui souffrent de dommages ou modifications à la suite d'une intrusion ou attaque (p.ex. logiciels, bases de données, documents électroniques, etc).
- La Perte de réputation (Lr), se réfère à la perte d'image ou crédibilité dès lors qu'il est impossible de fournir les produits ou services aux niveaux attendus.
- Les Procédures judiciaires (Lp), se réfère à des sanctions pénales qui peuvent survenir dès lors que les obligations contractuelles avec les clients ne peuvent être remplies.
- La perte de chiffre d'affaires avec des clients existants (LREC), représente les pertes qui se sont produites lors de l'incident empêchant la fourniture des produits ou services aux niveaux attendus par les clients existants.
- La Perte de revenus des clients potentiels (LRPC), se réfère aux pertes engendrées par l'impossibilité d'acquérir de nouveaux clients suite à l'incident.
- Les autres pertes (Ol), prennent en compte toutes les autres pertes qui peuvent survenir lors de l'incident (p.ex. la récupération du temps, les services externes, etc).
- L'Assurance contractée (Ci), cette valeur peut être soustraite de la quantité totale des pertes si l'organisation dispose d'une assurance qui couvre une partie des pertes.
- Le Taux annuel d'occurrence (ARO) se réfère à la fréquence estimée de l'attaque (p.ex. une fois par jour, une fois par an, etc).

A.3.1.1.2 La Valeur Annuel de l'Infrastructure (AIV): correspond aux coûts que le système est susceptible d'avoir en une année, quelle que soit la mise en oeuvre de contremesures, p.ex. les coûts d'équipement (Ec), les frais du personnel (Pc), les frais de service (SC), d'autres frais (Oc), et la valeur de revente (Rv), comme l'indique l'Équation A.4.

$$AIV = Ec + Pc + Sc + Oc - Rv \tag{A.4}$$

où,

- Le Coût d'Équipement (Ec), correspond au coût annuel des équipements de sécurité, des produits ou des matériaux (p.ex. l'achat, la location, le crédit-bail, les licences, etc) requises pour les opérations régulières de l'infrastructure.
- Le Frais du personnel (Pc), correspondent aux coûts des employés qui effectuent les opérations régulières de l'infrastructure (p.ex. les salaires, les primes, le paiement des heures supplémentaires, etc.)

- Le Coût des services rendus (Sc), correspondent aux coûts des services réguliers (p.ex. les frais d'électricité, d'infrastructures, de l'assurance contractée, etc), liés à l'organisation pour ses activités régulières.
- D'autres coûts (Oc), se réfèrent à tous les autres frais nécessaires au fonctionnement régulier de l'infrastructure du système.
- Les coûts de revente (Rv), se réfèrent à la valeur des équipements de sécurité après leur utilisation.

A.3.1.2 Paramètres Variables

A.3.1.2.1 La Mitigation du Risque (RM): considère le pourcentage de réduction du coût total de l'incident qui est donné à partir de la mise en oeuvre d'une contremesure. RM est calculée comme étant le produit entre la surface de couverture des contremesures (SC) et son Factor d'Efficacité (EF), tel qu'illustré dans l'Équation A.5.

$$RM = SC \times EF \tag{A.5}$$

où,

- La Surface de Couverture (SC) est le pourcentage de la surface d'attaque qui est couvert et contrôlé par une contremesure donnée. Par exemple, le blocage d'un utilisateur pourrait couvrir 85% de la surface d'attaque, et l'augmentation de la surveillance pour bloquer ponctuellement des opérations ne couvre que 60% de la surface d'attaque. Ces valeurs sont inhérentes à la nature de l'attaque et le système auquel il est affecté, ce qui permet d'obtenir des valeurs différentes de couverture de surface pour la même contremesure appliquée dans différents scénarios d'attaque.
- Le Facteur d'Efficacité (FE) considère le pourcentage de réduction du coût total de l'incident qui est donné à partir de l'application d'une mesure de sécurité. Selon la méthode de Norman [Nor10], le risque d'une attaque (R) est déterminé comme étant le produit de la vulnérabilité (V), la probabilité d'occurrence (P), et la conséquence (C), (c.-à-d., $R = V \times P \times C$). Un risque peut être atténué par la diminution de la vulnérabilité, la probabilité et / ou la conséquence. Par conséquent, le Factor d'Efficacité (EF) est calculé comme le pourcentage de réduction des risques qui résulte de l'application d'une contremesure donnée. p.ex. si le risque d'une attaque donnée avant contremesure est $R_1 = 10 \times 7 \times 7 = 490$ et le risque résultant après l'application d'une contremesure est $R_2 = 7 \times 6 \times 6 = 252$, $EF = 100 - (\frac{R_2 \times 100}{R_1}) = 51,43\%$

A.3.1.2.2 Le Coût Annuel de Réponse (ARC): se réfère au coût de la réponse associé à une contremesure donnée. Il inclut les coûts directs: p.ex. les coûts de mise en oeuvre (CoI), les coûts de maintenance (CoM), d'autres coûts directs (ODC) et les coûts indirects (Ic), comme indiqué dans l'Équation A.6.

$$ARC = CoI + CoM + Odc + Ic \tag{A.6}$$

où,

- Le Coût de mise en oeuvre (CoI), se réfère au coût du déploiement, d'installation et /ou de mise en oeuvre des mesures de sécurité afin d'atténuer une attaque donnée.
- Le Coût de maintenance (CoM), comprend le coût des services réguliers (p.ex. électricité, conseil, analyse, tests, etc) qui sont nécessaires pour l'implémentation des contremesures.
- Les autres coûts directs (Odc), se réfèrent à tous les autres coûts directs (p.ex. fournisseurs et partenaires) qui sont nécessaires pour mettre en place des contremesures.

- Les coûts indirects (Ic), se réfèrent aux conséquences qui peuvent parvenir lors de l'adoption d'une contremesure particulière aux utilisateurs légitimes.

Les paramètres ALE et AIV sont définis statiquement et dépendent de l'attaque détectée. Les paramètres RM et ARC dépendent des contremesures proposées. Leur calcul nécessite que le système détermine la couverture de surface discutée dans la section A.4.2.

A.3.2 Évaluation de Contremesures Individuelles

Le processus d'évaluation de contremesures commence en sélectionnant la première contremesure de la liste et en calculant son indice RORI. Le RORI obtenu est comparé contre celui par défaut, tel qu'illustré dans la Figure A.1.

Avant d'évaluer la première contremesure nous avons établie une valeur RORI par défaut égale à zéro (paramètres RM et ARC positionnés à 0, car aucune contremesure n'est mise en oeuvre). Si le RORI résultant est différent de celui par défaut, le système vérifie si le RORI évalué est supérieur à la valeur par défaut, dans ce cas-là, la contremesure évaluée est sélectionnée, et elle remplace la valeur du RORI par défaut. A contrario, si le RORI résultant est inférieur à celui par défaut, le système évalue une autre contremesure et le RORI par défaut reste inchangé.

Enfin, si le RORI résultant est égal à la valeur par défaut, le système vérifie le coût annuel de réponse (ARC) et sélectionne celui avec la valeur de coût la plus faible (il est toujours préférable de mettre en oeuvre une contremesure qui coûte le moins cher et fournit le plus grand bénéfice). Il peut arriver que, lorsque l'on compare les coûts de deux contremesures, ils sont exactement identiques. Dans ce cas-là, le système maintient la valeur du RORI par défaut et cherche une autre solution à évaluer.

Le processus est répété pour évaluer la deuxième contremesure sur la liste, puis le troisième, et ainsi de suite, jusqu'à ce qu'il ne reste plus de contremesures à évaluer. Le système sélectionne la dernière contremesure prise par défaut, car elle fournit l'indice RORI le plus élevé.

A.4 Processus de Sélection de Contremesures Combinées

Le processus de sélection des contremesures combinées prend compte des axiomes et approches combinatoires, pour estimer les paramètres liés aux contremesures p.ex. La Mitigation du Risque (RM) et Le Coût Annuel de Réponse (ARC). Cette section détaille chaque partie du modèle.

A.4.1 Axiomes Combinatoires

Les axiomes suivants ont été définis afin de calculer le coût et le niveau d'atténuation des risques qui résulte de la combinaison de deux ou plusieurs contremesures:

Axiome 1 : Le coût d'une contremesure combinée est égal à la somme de tous les coûts individuels des contremesures, p.ex. $ARC(C_1 \cup C_2) = ARC(C_1) + ARC(C_2)$, où l'ARC est le coût de la contremesure, et C_1, C_2 sont les contremesures individuelles. L'Équation A.7 illustre le cas le plus général.

$$ARC(C_1 \cup ... \cup C_n) = ARC(C_1) + ... + ARC(C_n) \tag{A.7}$$

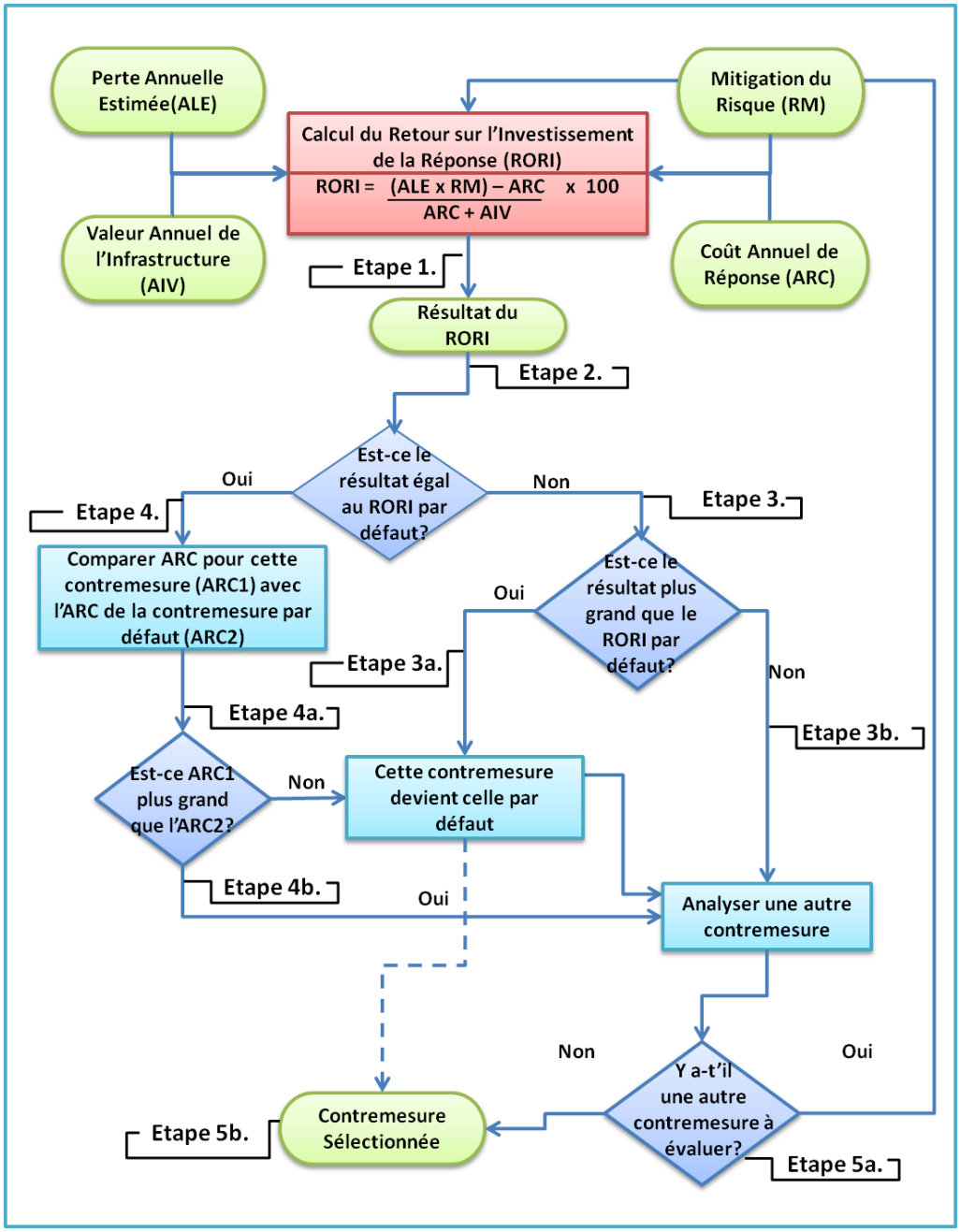

FIGURE A.1 - Processus d'Évaluation de Contremesures Individuelles

Cet axiome est utilisé pour des contremesures conjointes et disjointes, car l'application des multiples contremesures implique généralement des coûts plus élevés de mise en oeuvre, de consultation, de maintien, etc, par rapport aux coûts engagés par l'application d'une contremesure individuelle. De plus, dans une approche pessimiste qui consisterait à considérer le pire des cas, nous sommes en mesure d'estimer le retour sur investissement de la réponse pour une solution

combinée, ce qui signifie que si la solution est efficace pour un coût plus élevé, elle sera toujours efficace si le coût est inférieur à celui estimé.

Axiome 2 : L'Atténuation des risques (RM) pour une solution combinée est calculé comme la probabilité de leur recouvrement de la surface combinée $SC(C_1 \cup C_2 \cup ... \cup C_n)$ multiplié par le facteur d'efficacité (EF). Pour deux contremesures mutuellement exclusives (C_1, C_2), l'atténuation du risque est calculé comme $RM(C_1 \cup C_2) = SC(C_1) \times EF_1 + SC(C_2) \times EF_2$. Cependant, pour deux contremesures non mutuellement exclusives, l'atténuation du risque peut être calculée de deux façons :

1. Soit comme la somme de leurs surface de couverture individuelles multiplié par leurs facteurs d'efficacité respectifs, ensuite é de la surface de couverture de leur intersection multiplié par le facteur d'efficacité le plus faible, c.-à-d. $RM(C_1 \cup C_2) = SC(C_1) \times EF_1 + SC(C_2) \times EF_2 - SC(C_1 \cap C_2) \times min(EF_1, EF_2)$. L'Équation A.8 illustre le cas le plus général.

$$\begin{aligned} RM(C_1 \cup ... \cup C_n) = SC(C_1) \times EF_1 + ... + SC(C_n) \times \\ EF_n - \sum SC(\cap twoevents) \times min(EF_1, ..., EF_n) \\ + \sum SC(\cap threeevents) \times min(EF_1, ..., EF_n)... \\ +(-1)^{n+1} SC(C_1...C_n) \times min(EF_1, ..., EF_n) \end{aligned} \tag{A.8}$$

2. Soit comme la différence entre les deux surfaces de contremesures partiellement couvertes multiplié par leurs facteurs d'efficacité respectifs, ajouté de la surface de couverture de leur intersection multiplié par le facteur d'efficacité le plus élevé, c.-à-d. $RM(C1 \cup C2) = [SC(C_1) - (SC(C_1) \times SC(C_2))] \times EF_1 + [SC(C_2) - (SC(C_1) \times SC(C_2))] \times EF_2 + SC(C_1 \cap C_2) \times max(EF_1, EF_2)$.

A.4.2 Surface de Couverture de Contremesures

La surface de couverture est un travail en développement dans de nombreux domaines de recherche (p.ex. les réseaux de capteurs sans fil, les systèmes chimiques, le calcul de propriétés physiques, biologiques, etc.). En sécurité de l'information, la notion de recouvrement de la surface est liée à la surface de l'attaque, définie par Manadhata [MW10] comme le sous-ensemble des ressources du système qu'une entité malveillante peut utiliser pour envoyer/recevoir des données vers/à partir du système pour attaquer le système. Ainsi, plus la surface du système est grande, plus il y a d'opportunités d'attaque [HW07].

Intuitivement, la surface de couverture de contremesures représente le niveau d'actions que la solution de sécurité peut avoir sur une surface d'attaque d'un système. Plus clairement, la surface de couverture est le pourcentage de la surface d'attaque qui est couverte et contrôlée par une contremesure donnée. La surface de couverture est nécessaire pour calculer le niveau d'atténuation du risque (RM) pour une contremesure individuelle et combinée.

L'union de deux ou plusieurs surfaces est un indice qui varie entre la surface de couverture maximale de l'ensemble de contremesures (c.-à-d. max $SC(C_1, ..., C_n)$) dans sa limite inférieure, et la somme des surfaces individuelles (c.-à-d. $\sum SC(C_1), ..., SC(C_n)$) dans sa limite supérieure. L'intersection de deux ou plusieurs surfaces est un indice qui varie de zéro dans sa limite inférieure, à la surface de couverture minimal du groupe de contremesures dans sa limite supérieure (c.-à-d. min $SC(C_1, ..., C_n)$). Deux cas peuvent se distinguer dans le calcul de la surface de couverture

pour une contremesure combinée (c.-à-d. les surfaces jointes et disjointes). La Figure A.2 illustre ces cas.

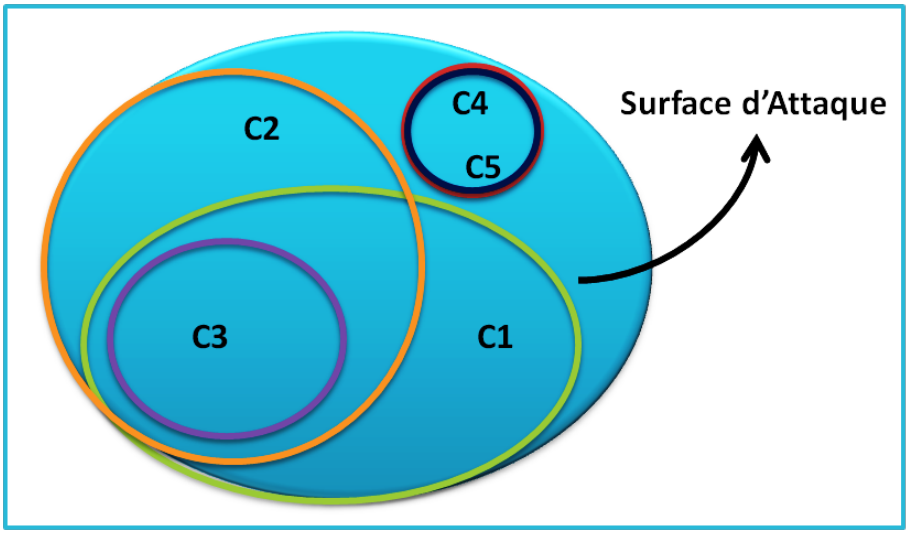

FIGURE A.2 - Surface de Couverture de Contremesures

- **Surfaces disjointes:** La surface de couverture d'une contremesure est disjointe de la surface de couverture d'une autre contremesure si elles n'ont pas d'éléments en commun. Par conséquent, avec deux contremesures disjointes (p.ex. C_1, C_4), la surface de couverture de l'union est calculé telle que $SC(C_1 \cup C_4) = SC(C_1) + SC(C_4)$, et la surface de couverture de l'intersection est égale à zéro.

- **Surfaces Jointes:** La surface d'une contremesure est jointe si elle est partiellement ou totalement recouverte par celle d'une autre contremesure. Pour des contremesures partiellement jointes (p.ex. C_1, C_2), l'union de la solution combinée est calculée telle que la somme des surfaces individuelles moins leur intersection (Équation A.9), et la surface de couverture de la l'intersection est calculée comme la moyenne des limites inférieures et supérieures (Équation A.10).

$$SC(C_1 \cup C_2) = SC(C_1) + SC(C_2) - SC(C_1 \cap C_2) \tag{A.9}$$

$$SC(C_1 \cap C_2) = \frac{SC(C_1 \cap C_2)_{LOW} + SC(C_1 \cap C_2)_{UP}}{2} \tag{A.10}$$

La limite inférieure de l'intersection entre deux contremesures $SC(C_1 \cap C_2)_{LOW}$ est égale à 0 si la somme de leurs surfaces est inférieure ou égale à 1, sinon, elle reçoit la valeur $(SC(C_1) + SC(C_2) - 1)$. La limite supérieure de l'intersection entre deux contremesures est égale à la couverture totale de la surface la plus petite.

Un cas plus général pour le calcul de l'union et l'intersection des contremesures combinées est présenté dans l'Équation A.11, où $x = SC(C_1) + ... + SC(C_n) - (n-1)$, et n= nombre maximal des contremesures à combiner.

$$\begin{aligned} SC(C_1 \cup ... \cup C_n) = &\quad SC(C_1) + ... + SC(C_n) - SC(C_1 \cap ... \cap C_n) \\ SC(C_1 \cap ... \cap C_n) = &\quad \tfrac{SC(C_1 \cap ... \cap C_n)_{LOW} + SC(C_1 \cap ... \cap C_n)_{UP}}{2} \\ SC(C_1 \cap C_n)_{LOW} = &\quad \begin{cases} 0 & if\ SC(C_1) + ... + SC(C_n) \leq n-1 \\ x & if\ SC(C_1) + ... + SC(_C n) > n-1 \end{cases} \\ SC(C_1 \cap ... \cap C_n)_{UP} = &\quad min\ \{SC(C_1), ..., SC(C_n)\} \end{aligned} \tag{A.11}$$

Pour les surfaces jointes en totalité (p.ex. C_1, C_3), l'union de la surface de couverture combinée est calculée telle que $SC(C_1 \cup C_3) = SC(C_1)$, et la surface couverture de l'intersection est calculée telle que $SC(C_1 \cap C_3) = SC(C_3)$. Une variante de ce cas se produit lorsque les deux contremesures ont la même valeur de surface, les surfaces de couverture de l'union et l'intersection sont alors calculées telles $SC(C_4 \cup C_5) = SC(C_4 \cap C_5) = SC(C_4) = SC(C_5)$.

A.4.3 Approches Combinatoires

Pour déterminer le nombre maximal de solutions combinées qui peuvent résulter d'un ensemble de contremesures individuelles, nous définissons un ensemble d'éléments où l'ordre n'est pas important et les répétitions ne sont pas autorisées. Cette section décrit les deux approches (c.-à-d. restrictive et non-restrictive) qui sont considérées dans la combinaison de contremesures de sécurité.

A.4.3.1 Approche non-restrictive

Des contremesures non restrictives sont celles qui peuvent être parfaitement combinées (c.-à-d. il n'existe aucune restriction dans leur combinaison). Le nombre maximal des contremesures est calculé comme la somme de la n-nième ligne (en partant de zéro) des coefficients binomiaux [Gri85], [Ros94], tel qu'exprimé dans l'Équation A.12, où "n" est le nombre total d'éléments à combiner et "k" est l'ensemble des éléments combinés (de zéro à "n").

$$\sum_{0 \leq k \leq n} \binom{n}{k} = 2^n - 1 - n \tag{A.12}$$

p.ex. pour un groupe de 4 contremesures, le nombre maximal de solutions combinées est $\binom{4}{k} = 2^4 - 1 - 4 = 11$. Le Tableau A.2 résume ces résultats.

A.4.3.2 Approche Restrictive

Trois cas peuvent apparaître lorsqu'une ou plusieurs contremesures sont restrictives et ne peuvent pas être combinées avec d'autres contremesures.

A.4.3.2.1 Contremesures Mutuellement Exclusives : En probabilité, deux événements sont mutuellement exclusifs s'ils ne peuvent pas se produire simultanément (c.-à-d. qu'ils n'ont pas de résultats en commun) [RH98], [Olo05]. Considérant le cas d'avoir quatre contremesures (C1, C2, C3, C4), et sachant que C1 et C3 sont mutuellement exclusives (elles ne peuvent pas être combinées), le nombre total de combinaisons possibles est donnée par l'Équation A.13 [And], [ML].

$$\sum_{0 \leq k \leq n} \binom{n}{k} = \left[(k+1)(2^{n-k})\right] - 1 - n \qquad for\ \ k \geq 2 \tag{A.13}$$

où "n" est le nombre total d'éléments à combiner et "k" le nombre de contremesures qui s'excluent mutuellement. Dans ce cas, nous avons n = 4 et k = 2, ce qui donne $\binom{4}{2} = \left[(2+1)(2^{4-2})\right] - 1 - 4 = 7$ combinaisons possibles. Le Tableau A.2 résume ces résultats.

A.4.3.2.2 Contremesures Partiellement Restrictives : Une contremesure peut être mise en oeuvre avec quelques autres contremesures mais pas avec leur totalité. Par exemple, supposons que de l'ensemble de 4 contremesures (C1, C2, C3, C4), C1 ne peut être combinée qu'avec C4 (les combinaisons de C1 avec C2 et C1 à C3 créent un conflit dans le système). Le nombre total de combinaisons possibles est alors donné par lÉquation A.14 [FS09b].

$$\sum_{0 \le k \le n} \binom{n}{k} = \left[(2^{k-1}+1)(2^{n-k})\right] - 1 - n \qquad for \;\; k \ge 2 \tag{A.14}$$

où "n" est le nombre total d'éléments à combiner et "k" le nombre de contremesures partiellement restrictives. Dans ce cas, nous avons n=4 et k=3, ce qui donne $\binom{4}{3} = \left[(2^{3-1}+1)(2^{4-3})\right] - 1 - 4 = 5$ combinaisons possibles. Le Tableau A.2 résume ces résultats.

A.4.3.2.3 Contremesures Totalement Restrictives : Parmi le groupe de contremesures sélectionnées, une ou plusieurs ne peuvent pas être combinées avec le reste des contremesures. Pour connaître le nombre exact de combinaisons possibles, on utilise alors l'Équation A.15 [Ros94], [War07].

$$\sum_{0 \le k \le n} \binom{n}{k} = 2^{n-k} + (k-1) - n \qquad for \;\; k \ge 1 \tag{A.15}$$

où "n" est le nombre total d'éléments à combiner et "k" le nombre de contremesures totallement restrictives. Par exemple, en ayant 4 contremesures (C1, C2, C3, C4) et en sachant que C1 est totallement restrictive, nous aurons $\binom{4}{1} = 2^{4-1} + (1-1) - 4 = 4$ combinaisons possibles. Le Tableau A.2 résume ces résultats.

TABLE A.2 - Combinaissons générées d'un groupe de 4 contremesures

Type	**Non-Restrictive**	**Mutuellement Exclusive**	**Partiellement Restrictive**	**Totalement Restrictive**
Double	C1+C2, C1+C3, C1+C4, C2+C3, C2+C4, C3+C4,	C1+C2, C1+C4, C2+C3, C2+C4, C3+C4,	C1+C4, C2+C3, C2+C4, C3+C4,	C2+C3, C2+C4, C3+C4,
Triple	C1+C2+C3, C1+C2+C4, C1+C3+C4, C2+C3+C4,	C1+C2+C4, C2+C3+C4	C2+C3+C4	C2+C3+C4
Quadruple	C1+C2+C3+C4			
Total	**11**	**7**	**5**	**4**

A.4.4 Évaluation des Contremesures Combinées

Une contremesure combinée résulte de la mise en oeuvre simultanée de deux ou plusieurs contremesures pour atténuer une attaque donnée. Une solution combinée est donc analysée comme

une contremesure individuelle, avec un coût combiné et une efficacité combinée. La combinaison des solutions de sécurité est possible seulement si les conditions suivantes sont remplies:

1. Les contremesures ne sont pas mutuellement exclusives,
2. La surface totale de couverture d'une contremesure n'est pas totalement recouverte par une autre contremesure avec une surface de couverture identique.

Le processus de sélection et de classement des contremesures combinées est représenté dans la Figure A.3. Le processus prend comme entrée les résultats obtenus à partir de l'évaluation des contremesures individuelles, tel que décrit dans la Section A.3.2 (étape 1). Le système élimine ensuite les contremesures pour lesquelles l'indice RORI est inférieur à la moyenne ou au-dessous d'un seuil prédéfini (étape 2). Cette action permet au système d'optimiser le processus d'évaluation.

Une fois que nous avions la liste des contremesures combinables (étape 3), on calcule le nombre total de combinaisons possibles. Ensuite, il est possible de générer des groupes de 2, 3, ..., n contremesures, où "n" est le nombre total d'éléments à combiner (étape 4).

La métrique RORI est ensuite calculée pour chaque groupe de combinaisons (étape 5), en tenant compte du fait que pour une solution combinée, le coût est calculé comme la somme de tous les coûts individuels des contremesures (Axiome 1) et l'atténuation du risque d'une solution combinée est calculée comme étant la probabilité de l'union des événements (Axiome 2). La valeur annuelle des infrastructures et la perte annuelle estimée demeure invariable pour toutes les solutions combinées.

Afin d'évaluer toutes les contremesures combinées, il est nécessaire d'utiliser les axiomes 1 et 2 décrites en Section A.4.1. Dans l'étape 6, la valeur RORI obtenue est comparée à la valeur par défaut (la valeur RORI la plus élevée obtenue dans l'évaluation individuelle des contremesures).

Si le RORI résultant est égal à la valeur par défaut, le système vérifie le coût de la contremesure (ARC) et sélectionne celle qui a la valeur la plus basse (étape 6a). Dans le cas contraire, le système compare si le RORI actuel est supérieur à la valeur par défaut (étape 6b), la contremesure est alors sélectionnée, et elle remplace la valeur par défaut (étape 7a).

Toutefois, si le RORI résultant est inférieure à la valeur par défaut (étape 7b), le système cherche une autre combinaison à évaluer (étape 8a).

Lorsqu'aucune autre combinaison est possible (étape 8b), le système conserve la contremesure par défaut, car c'est elle qui offre le plus haut indice RORI.

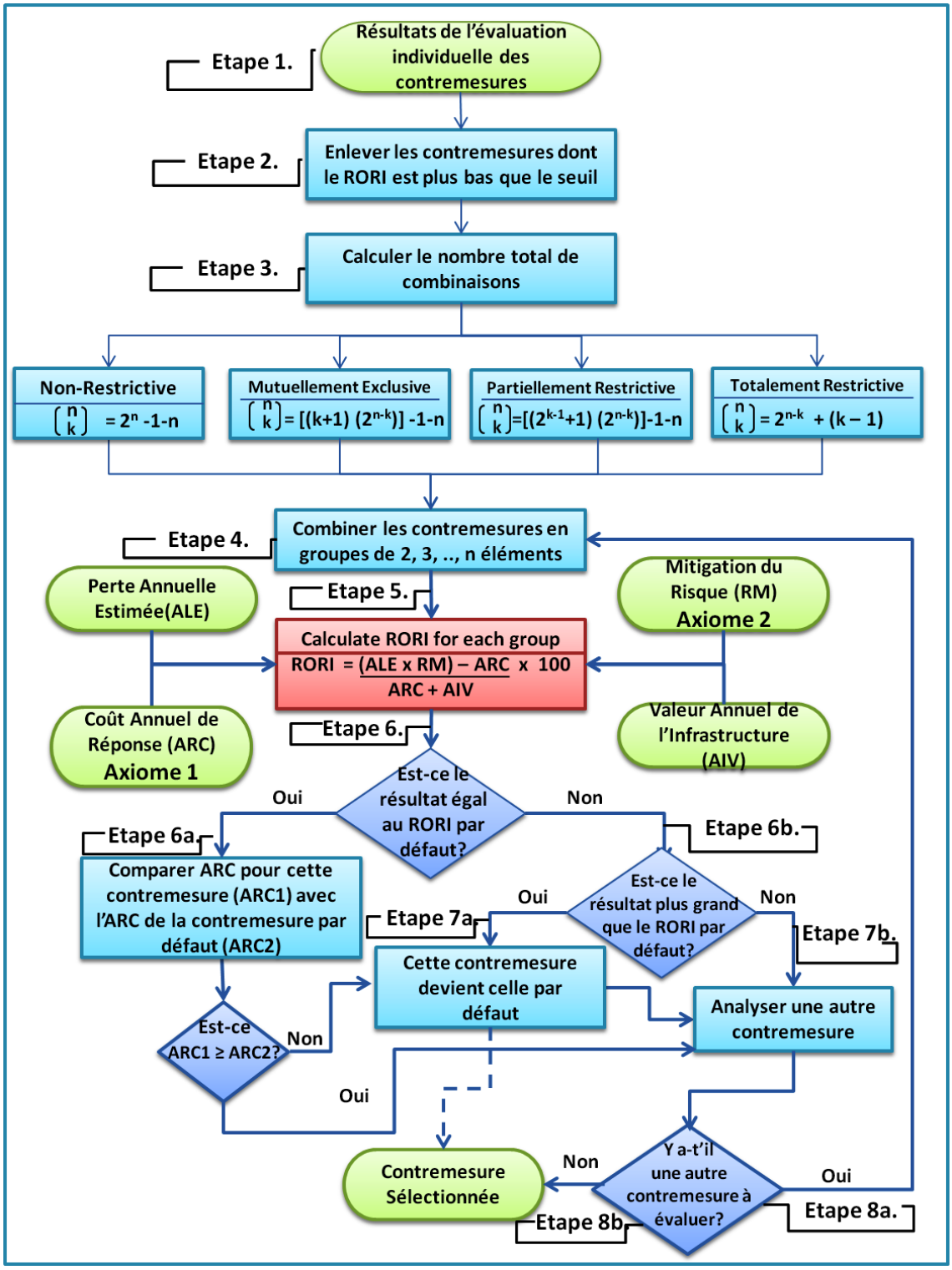

FIGURE A.3 - Processus d'Évaluation de Contremesures Combinées

A.5 Travaux Liés

La plupart des travaux existants concernant la sélection des contremesures se concentre uniquement sur les modèles qui utilisent l'analyse qualitative ou sur l'évaluation d'une solution individuelle pour une seule attaque. Par exemple, Cavusoglu et al. [CMR04] proposent un modèle pour évaluer des options d'investissement de sécurité qui utilise une approche de l'arbre d'attaque basée sur la théorie des jeux. A noter que leur modèle ne considère la mise en oeuvre de contremesures en cas d'une attaque individuelle exclusivement. Notre solution quand à elle peut être adaptée à des scénarios d'attaques multiples.

Duan et Cleand-Huang [DCH06] considèrent des méthodes heuristiques ainsi qu'une approche d'algorithmes génétiques pour le processus de sélection d'un ensemble de contremesures. Toutefois, en raison de la complexité de l'espace de recherche, l'approche heuristique n'est ni optimale, ni complète. Les principaux inconvénients de l'approche de l'algorithme génétique est la difficulté de mesurer avec précision la meilleure partie de chaque contremesure à combiner. En plus, l'utilisation de la valeur actualisée nette (VAN) comme mesure de comparaison n'est pas suffisante pour décider entre plusieurs contremesures. Notre solution utilise l'ALE au lieu de la VAN, puisque ce dernier n'est utile qu'avec des investissements de longs périodes. Les contremesures dans notre modèle sont proposées pour être mises en oeuvre pour une courte période de temps (à partir du moment où une intrusion est détectée jusqu'à ce que le système reviennet à son fonctionnement normal).

Par ailleurs, Neubauer et al. [NSW06] proposent l'utilisation de portefeuilles de sauvegarde efficaces qui sont évalués en fonction de plusieurs objectifs (p.ex. la valeur de l'image, la valeur monétaire, le coût accepté, le temps de configuration, etc.) Cependant, un modérateur est nécessaire pour des conseils pendant le processus, les calculs se fondent principalement sur la métrique de perte annuelle attendue, et des résultats incertains de milieux complexes ne sont pas considérés par l'outil. Notre modèle ne nécessite pas de modérateur au cours du processus d'évaluation et utilise non seulement la perte annuelle attendue, mais aussi le coût de fonctionnement, le coût de contremesures et le niveau d'atténuation des risques afin d'évaluer les dommages collatéraux de réponses et les effets d'intervention sur les attaques.

Bisterilli et al. [BFP07] présentent une approche qualitative pour la sélection des contremesures de sécurité en utilisant des arbres de défense (une extension des arbres d'attaque) et les préférences sur les contremesures en utilisant les réseaux de préférences conditionnelles (CP-net). Cependant, les conditions de sélection de contremesures sont fondées sur la connaissance d'experts, l'approche est statique et qualitative (pas de méthode mathématique pour évaluer et sélectionner des contremesures), et elle ne tient pas compte des attaques comme des variables aléatoires. Notre solution propose un modèle quantitatif (basé sur l'indice RORI) qui ne repose pas sur des connaissances expertes pour la sélection des contremesures appropriées.

Zonouz et al. [ZKSY09] proposent un moteur de réponse et récupération (RRE) qui utilise une approche graphique d'arbre pour analyser les événements et les contremesures sélectionnées sur la base de la logique booléenne. RRE modèle un scénario de jeu à deux joueurs (l'offensive et la défensive), et choisit les mesures d'intervention en résolvant des processus des décisions concurrentielles de Markov partiellement observables, qui proviennent des arbres de réponse d'attaque. Cependant, cette approche ne tient pas compte des avantages et des coûts associés à une action de réponse donnée. Et elle n'évalue pas quantitativement les différentes contremesures pour sélectionner celle qui offre le plus grand bénéfice à l'organisation.

Plus récemment, Bedi et al. [BGS+11] décrivent une approche qui utilise un algorithme pour générer des ensembles optimaux de contremesures. Cependant, cette approche est proposée pour être mise en oeuvre durant la phase de conception du logiciel uniquement, et elle ne tient pas compte des menaces non identifiées, ni de l'impact des contremesures dans le processus d'analyse et de sélection. A contrario, notre solution peut être mise en oeuvre dans le déploiement en temps

réel, et elle examine l'impact des contremesures (le coût, le niveau d'atténuation des risques), aussi bien durant la phase de conception que lors du processus d'évaluation et de sélection.

A.6 Conclusions et Travaux Futures

Dans cette thèse, nous introduisons une approche quantitative pour sélectionner des contremesures de sécurité optimales basées sur l'indice RORI, ce qui permet de fournir une réponse par rapport à la taille de l'infrastructure en évaluant des solutions individuelles et combinées contre l'absence d'action.

Notre solution est divisée en deux parties: dans un premier temps, l'évaluation des contremesures individuelles, qui détermine les paramètres associés à l'intrusion ou l'attaque (p.ex. l'ALE et AIV) et évalue l'indice RORI pour toutes les solutions individuelles; et dans un second temps, l'évaluation des contremesures combinées, qui détermine les paramètres associés aux solutions combinées (p.ex. l'ARC et le RM) et évalue l'indice RORI pour toutes les combinaisons possibles des mesures de sécurité. En conséquence, notre modèle est capable de classer et sélectionner la ou les contremesures qui offrent le plus grand bénéfice à l'organisation.

La combinaison de deux ou plusieurs politiques de sécurité représente généralement une plus grande couverture de la surface du risque et offre ainsi un niveau plus intéressant d'atténuation du risque. Cependant, il est important de noter qu'une valeur supérieure d'atténuation du risque ne débouche pas toujours sur un indice RORI plus élevé, ni que la solution la moins chère est toujours la plus intéressante. Plusieurs paramètres sont pris en compte afin de choisir la solution qui offre leur meilleur ratio coût-efficacité.

Différents cas d'usages ont été donné tout au long du mémoire de thèse pour montrer l'applicabilité du modèle (c.-à-d. un service de transfert d'argent mobile, où des attaques sont effectuées contre les utilisateurs du système ; un processus de contrôle d'infrastructure critique, où des attaques sont effectuées contre une barrage ; et un système de jeux olympiques, où des multiples attaques affectent le système d'information).

Les travaux futurs se concentreront à considérer l'interdépendance entre les contremesures (p.ex. comment l'application d'une contremesure peut affecter l'efficacité des autres), la gestion des conflits que peuvent apparaître lors de l'implémentation des contremesures, ou l'adaptation du modèle RORI à un environnement dynamique (incluant des paramètres dynamiques tel que le temps de réaction et de récupération, la puissance de la contremesure, etc).

Appendix B

MMTS Inference Rules

This appendix presents inference rules based on the Semantic Web Rule Language (SWRL), which are used to infer relationships among SIEM operations. These rules can be extended towards first order logic, and follow the formalism used in equation B.1.

The intended meaning can be interpreted as: whenever a condition defined in the antecedent holds, a condition defined in the consequent must also hold. Some conditions that can not be expressed in OWL, can be expressed by using SWRL rules or by deploying SWRL queries. These rules are written in terms of classes, properties and OWL individuals.

$$Antecedent(?x, ?y) \wedge Antecedent(?y, ?z) \rightarrow Consequent(?x, ?z) \tag{B.1}$$

The formalism uses parenthesis "()" to encapsulate the arguments of the expressions (e.g., individuals, data values or variables referring to them). The AND symbol "$\wedge$" is used to combine two or more atoms. The arrow "$\rightarrow$" separates the antecedent part of the rule from its consequent part and defines the resulting atom of the rule. variables are represented by question marks "?". Predicates that take one or more arguments and evaluates to true if they satisfy the condition are expressed by the qualifier "swrlb".

The following rules have been proposed to improve the detection and reaction process in the Mobile Money Transfer Service over different attack scenarios.

B.1 Trafficking Collection

In the Mobile Money Transfer System, an mWallet can be credited by using various loading means such as a credit card, a bank transfer, a transfer from another mWallet, a payment through a payment intermediary. An mWallet which is credited by many different entities with several loading means may be a dealer's account used to collect payments related to his trafficking. Such a case should be detected. Once this rule is implemented in a SIEM it should be possible for the user to adjust the considered parameters, such as the number of loading means, the number of transfer senders, the loading means considered, the slide of time considered.

$$Analyser(?a) \wedge User\_Account(?x) \wedge User\_Account(?y) \wedge User\_Account(?z)$$
$$Trans\_Amount(?ta) \wedge Trans\_Amount(?tb) \wedge is\_credited\_with(?x, ?ta) \wedge$$
$$is\_credited\_by(?ta, ?y) \wedge is\_credited\_with(?x, ?tb) \wedge is\_credited\_by(?tb, ?z)$$
$$Event\_Frequency(?ef) \wedge Event\_Threshold(?et) \wedge swrlb : greaterThan(?ef, ?et)$$
$$\wedge Traffic king\_Collection(?e) \rightarrow generate\_event(?a, ?e) \quad \text{(B.2)}$$

B.2 Hiding user

Generally, a personal account is used in face to face payments as well as for remote payments. An account used only for remote transactions can therefore be a sign of suspicious activity. This rule requires studying an account's behaviour during a certain period of time. It should be possible for the user to set this parameter.

$$Analyser(?a) \wedge User\_Account(?x) \wedge Remote\_Trans(?rt) \wedge is\_performed\_by(?rt, ?x)$$
$$Event\_Frequency(?ef) \wedge Event\_Threshold(?et) \wedge swrlb : greaterThan(?ef, ?et)$$
$$\wedge Hiding\_User\_Identity(?e) \rightarrow generate\_event(?a, ?e) \quad \text{(B.3)}$$

B.3 Scams

In some types of frauds, a crook tricks persons into sending him or her some money. Generally, the money is withdrawn very shortly after it is received. The repetition of this pattern can be a sign of fraudulent activity and it should be detected. The slide of time considered, the number of repetitions and the time separating reception and withdrawal of money are parameters that should be modifiable. The number of different money senders is also a relevant parameter as scams usually target a wide range of persons.

$$Analyser(?a) \wedge User\_Account(?x) \wedge Trans\_Amount(?ta) \wedge Trans\_Time(?t1)$$
$$Trans\_Time(?t2) \wedge is\_received\_by(?ta, ?x) \wedge is\_received\_at(?ta, ?t1) \wedge$$
$$is\_withdrawn\_by(?ta, ?x) \wedge is\_withdrawn\_at(?ta, ?t2) \wedge swrlb : greaterThan(?t2, ?t1)$$
$$Event\_Frequency(?ef) \wedge Event\_Threshold(?et) \wedge swrlb : greaterThan(?ef, ?et)$$
$$\wedge Scam\_Detected(?e) \rightarrow generate\_event(?a, ?e) \quad \text{(B.4)}$$

B.4 Virtual money creation/destruction

Within the Mobile Money Transfer System, a certain amount of virtual money is authorized to circulate. This amount has to be controlled strictly to avoid any money creation and destruction.

The amount should remain constant at all time and any change should be detected by the SIEM.

B.5 Account takeover

If a mobile phone is stolen from its legitimate user and is used by the thief to make money transfers it is very likely that the thief's behaviour will greatly differ from the original user's one. In order to detect such a case, it should be possible for a SIEM to learn during a certain period of time what are the user's habits and his general behaviour. This learning stage result is a profile which can be used to detect unusual behaviour. For example, the SIEM will raise an alarm if many transfers are made abroad while the client seldom travels.

B.5.1 Account Takeover by Number of Transactions

$$MMTS\_Account(?x) \wedge Analyser(?a) \wedge Trans\_Rate(?tr) \wedge has\_trans\_rate(?x, ?tr) \wedge Threshold\_Rate(?th) \wedge swrlb : greaterThan(?tr, ?th) \wedge Account\_Takeover\_Attack(?z) \rightarrow is\_affected\_by(?x, ?z) \wedge generate\_event(?a, ?z) \tag{B.5}$$

In Equation B.5, the system detects that a user account (x) has a current transaction rate (tr) greater than the predefined threshold rate (th) and therefore, it is assumed that the account (x) is affected by an account takeover attack (z) and an event is generated accordingly by an analyser (a).

B.5.2 Account Takeover by Amount of Transactions

$$MMTS\_Account(?x) \wedge Analyser(?a) \wedge Trans\_Amount(?ta) \wedge has\_trans\_amount(?x, ?ta) \wedge Threshold\_Amount(?th) \wedge swrlb : greaterThan(?ta, ?th) \wedge Account\_Takeover\_Attack(?z) \rightarrow is\_affected\_by(?x, ?z) \wedge generate\_event(?a, ?z) \tag{B.6}$$

Similarly, in Equation B.6, the system detects that a user account (x) has a current transaction amount (ta) greater than the predefined threshold transaction amount (th) and therefore, it is assumed that the account (x) is affected by an account takeover attack (z) and an event is generated accordingly by an analyser (a).

B.6 Employee complicity

It can also be interesting to detect whether employees are complicit with fraudsters. To detect such a case, we may consider a rule which identifies employees that have opened a certain number of accounts suspected in fraudulent activities.

B.7 Denial of Service

An increment on the number of connections performed in a period of time without sending an acknowledgement to complete the connection or a huge load of transactions for an irrelevant amount can be interpreted as a Denial of Service attack.

B.7.1 DoS without Acknowledgement

$$\begin{gathered} Analyser(?a) \wedge User\_Account(?x) \wedge User\_Account(?y) \wedge Denial\_Of \\ \_Service(?z) \wedge Trans\_Without\_Ack(?tr1) \wedge Trans\_Without\_Ack(?tr2) \\ \wedge Trans\_Time(?t) \wedge is\_performed\_by(?tr1, ?x) \wedge is\_performed\_at(?tr1, ?t) \\ \wedge is\_performed\_by(?tr2, ?y) \wedge is\_performed\_at(?tr2, ?t) \wedge Event\_ \\ Frequency(?ef) \wedge Threshold(?th) \wedge swrlb : greaterThan(?ef, ?th) \\ \rightarrow generate\_event(?a, ?z) \end{gathered} \tag{B.7}$$

In Equation B.7, the system detects that several user accounts (x,y) have initiated transactions (tr1, tr2) with the Mobile Money Transfer System at time (t), but none of the transactions have been acknowledged. Since the event has occurred in a frequency greater than a predefined threshold, it is assumed that the system is affected by a Denial of Service attack (y) and an event is generated by an analyser (a).

B.7.2 DoS with Transactions for less than 1 Euro

$$\begin{gathered} Analyser(?a) \wedge User\_Account(?x) \wedge User\_Account(?y) \wedge DoS\_Attack(?z) \\ \wedge Transaction(?tr) \wedge swrlb : lesserThan(?tr, 1) \wedge Trans\_Time(?t) \\ \wedge is\_performed\_by(?tr, ?x) \wedge is\_performed\_by(?tr, ?y) \wedge is\_performed \\ \_at(?tr, ?t) \wedge Event\_Frequency(?ef) \wedge Threshold(?th) \\ \wedge swrlb : greaterThan(?ef, ?th) \rightarrow generate\_event(?a, ?z) \end{gathered} \tag{B.8}$$

Similarly, in Equation B.8, the system detects that several MMTS user accounts (x,y) have initiated transactions (tr) with the Mobile Money Transfer System at time (t) for an amount of less than one euro. Since the event has occurred in a frequency greater than a predefined threshold, it is assumed that the system is affected by a Denial of Service attack (y) and an event is generated by an analyser (a).

B.8 Conclusions

The rules mentioned above rely on three common points. First of all, it is required to study transactions during a certain span of time. Secondly, for each event it is necessary to consider multiple parameters. Thirdly, they require a wide range of logical operators. Fourthly, these rules may require using data from different logs. Lastly, it should be possible for the operator in charge of surveillance to modify some parameters of rules in order to better target his or her analysis.

Appendix C

URI General Structure

C.1 URI Scheme

A URI always begins with a scheme name that refers to a specification for assigning identifiers within that scheme [BLFM05]. The URI syntax is therefore a federated and extensible naming system where each scheme's specification may further restrict the syntax and semantics of identifiers within that scheme. Scheme names consist of a sequence of characters beginning with a letter and followed by any combination of letters, digits, plus sing (+), period (.), or hyphen (-), as shown in Listing C.1

Listing C.1 - URI Scheme

```
scheme = ALPHA \*(ALPHA/DIGIT/+/./-)
e.g., ftp://example.org/aDirectory/aFile
      mailto:name@example.org
      ldal://ldap.example.org
      tel:+1-816-555-1212
      http://www.site.com
```

Schemes are case-insensitive, but its representation should only produce lower-case scheme names for consistency. More details on the designing of URI schemes can be found in [MAZP99].

C.2 URI Authority

The authority element is located after the URI scheme and is preceded by a double slash (//) and terminated either by the next slash (/), question mark (?), sign number (#), or by the end of the URI. The governance of the name space defined by the reminder of he URI is delegated to the URI authority. This latter has three components: user information, host and port, as shown in Listing D.4

Listing C.2 - URI Authority Syntax

```
authority = [userinfo @] host [: port]
```

C.2.1 User Information

The userinfo element consists of a user name, and optionally, scheme-specific information about the way to get authorization o access a resource. Such element, if present, is followed by a commercial at-sign() that delimits it from the host. Listing C.3 shows the user information generic syntax.

C.2.2 Host

The host element is identified by an IP literal encapsulated with square brackets, an IPv4 address in dotted-decimal form, or a registered name (Listing C.3).

C.2.3 Port

The port element is designated by an optional port number in decimal following the host and delimited from it by a single colom (:) character (Listing C.3). Default ports may be defined by schemes. For instance, http scheme defines a default port 80. However, in order to prevent redundancy, URI port components and its delimiter (:) should be omitted if port is empty or if its value is the same as that of the scheme's default.

LISTING C.3 - URI Authority Components

```
userinfo    = *(unreserved / pct-encoded / sub-delims / :)
host        = [IP-literal] / IPv4-address / regular-name
port        = :*DIGIT

e.g. ftp://[<user>[:<password>]@]<host>[:<port>]/<uri-path>
```

C.3 URI Path

The path element contains information that is usually organized in hierarchical form. Such information, along with data in the non-hierarchical query component, serves to identify resources within the scope of the URI's scheme and naming authority (if any). The path consists of a sequence of path segments separated by a slash (/), and terminated either by the first question mark (?), sign number (#), or by the end of the URI.

Depending upon the path in the URI, certain rules may apply. For instance: if a URI contains an authority component (path-abempty), the path component must begin with a slash (/) or be empty. Otherwise (path-absolute), the path cannot begin with two-slash characters (//). Listing C.4 shows these rules.

LISTING C.4 - URI Path Rules

```
path-abempty  = begins with / or is empty
path-absolute = begins with / but not with //
path-noscheme = begins with a non-colon segment
path-rootless = begins with a segment
path-empty    = zero characters
```

C.4 URI Query

The URI query element is generally used to carry identifying information in the form of key=value pairs. The query component is indicated by the first question mark (?) and terminated either by a sign number (#), or by the end of the URI. The general syntax of a URI query is shown in Listing C.5

Listing C.5 - URI Query Syntax

```
query = *(pchar / "/" / ?)
```

C.5 URI Fragment

The fragment element of a URI allows indirect identification of a secondary resource (e.g., subset of a primary resource, some view of representations of the primary resource, other resources) by reference to a primary resource and additional information. A fragment identifier is indicated by the presence of a sign number (#) and terminated by the end of the URI. Listing C.6 shows the general syntax of a URI fragment.

Listing C.6 - URI Fragment Syntax

```
query = *(pchar / "/" / ?)
```

Appendix D

Use Case Implementation: PyOrBAC for the MMTS Scenario

This section provides an implementation of the MMTS Scenario (described in Section 5.1) using the OrBAC formalism to define the roles that subjects, actions, and objects play in the MMTS infrastructure.

D.1 System Architecture

The MMTS demonstrator is composed of three modules: The MMTS Simulator module, the Detection & Correlation module, and the Decision Support & Reaction module; this latter implements our countermeasure selection process (Figure D.1).

The following subsections detail each module of the system architecture as well as a typical use-case scenario.

D.1.1 MMTS Simulator

It is a module composed of four elements: a model of the User Behaviour, the Front Office, the AMS, and the Security Database, whose mission is the management of the user accounts and the generation of artificial logs [Con11, GHA+12, JTT10].

- **User's Behaviour Model:** This element models the user's behaviour and creates labels to describe the reality about the fraudulent or non-fraudulent nature of a transaction.
- **Front Office (FO):** This element is in charge of the user authentication; profile management (password change, user information, etc.); interface between the AMS and the users (balance consultation); transfer of transaction requests to the AMS and logs generation.
- **Account Management System (AMS):** The main functionalities of this module are: Logs generation; database consult; acceptance and/or denial of transactions.
- **Security Database:** It contains all the security information related to MMTS users (thresholds, block- ed accounts, activated/deactivated accounts, number of transactions within a period of time, etc). The FO and AMS elements consult the security database to guarantee

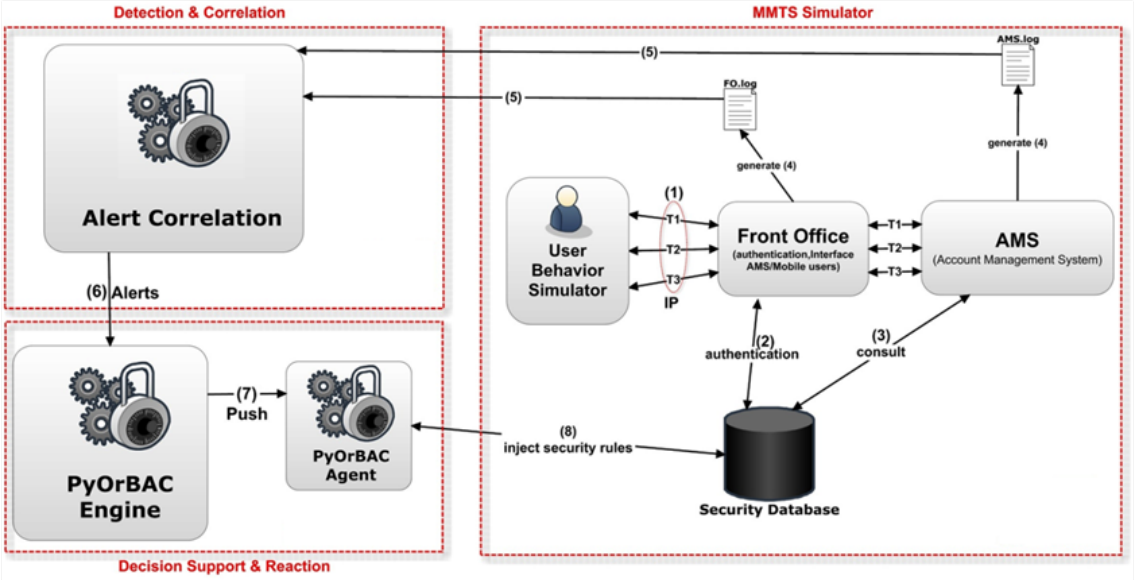

FIGURE D.1 - MMTS Use Case Service Architecture

the authentication functioning and the management of transactions.

D.1.2 Detection & Correlation Module

It is a framework in charge of receiving and analysing the logs provided by the detection module in order to correlate abnormal behaviours that can be interpreted as intrusions or attacks. The correlation engine feeds the Decision Support and Reaction module with correlated alerts that activate attack contexts.

This module is composed of two elements: the GEventeneric Event Translation (GET) Framework, and the Security Probes, that are in charge of receiving and analyzing the logs provided by the detection module in order to correlate abnormal behaviors that can be interpreted as intrusions or attacks [CCCR07, CDER09, FCR09] .

- **GET Framework:** It is in charge of gathering the data provided by the FO and by the AMS, of parsing their content (using the Adaptable Parser technology) and of normalizing the data fields to a common representation, to translate the messages to a format which is suitable for further processing;
- **Security Probes:** These components are capable of performing sophisticated cross-layer data correlation for the recognition of complex event patterns in the data flow. Patterns that are to be detected are described by means of Finite State Machines that are executed on top of an event-based platform. Alerts generated by the Security Probes are then fed to the PyOrBAC module, which is in charge of attack analysis.

D.1.3 Decision Support & Reaction Module (DS&R)

The DS&R module of five elements: The PyOrBAC Engine, PyOrBAC Agent, XOrBAC, Return On Response Investment (RORI) and the Policy Enforcement Point (PEP), whose main function is to evaluate, select and implement optimal security policies to react over intrusions and/or attacks

[Con11c, Con12a, GDJ+12, GDJC12]. Figure D.2 shows the general architecture of the DS&R module.

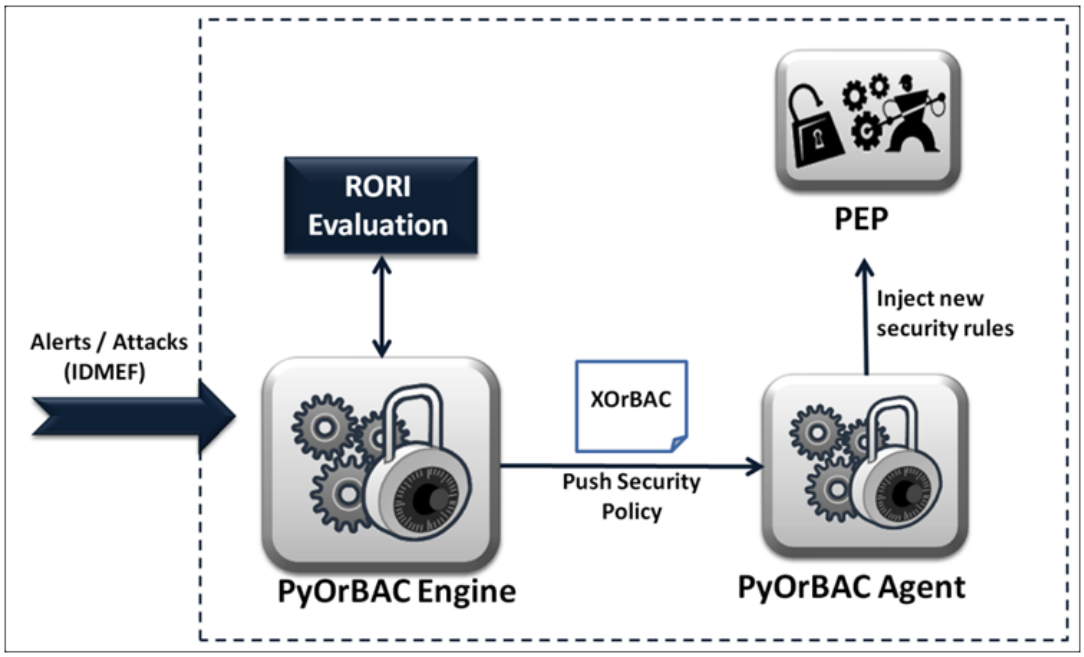

FIGURE D.2 - General Architecture of the DS&R Module

- **PyOrBAC Engine (Policy Management):** This element has the following functions: definition of security policies and different OrBAC elements (roles, activities, views, etc); dynamic generation of security policies through the use of contexts which allow the system to react more rapidly to any change; conflict management; and configuration of external systems called associated components.
- **PyOrBAC Agent (Policy Activation):** This element is in charge of translating the security policies that are received by the Engine and to configure these policies in the Policy Enforcement Points -PEP- (e.g., Security Database).
- **XOrBAC:** it is an XML modeling of the PyOrBAC implementation that provides XML schema to structure the data storage and communications between the PyOrBAC Engine and Agent. Through the XOrBAC file, the PyOrBAC Agent receives the commands sent by the Engine regarding the security policies that should be activated or deactivated in a given Organization.
- **Return On Response Investment (RORI):** it is a cost sensitive model that allows the evaluation of individual and multiple countermeasures by considering parameters associated to the intrusion or attack (e.g., Annual Loss Expectancy, Annual Infrastructure Value), as well as parameters associated to the proposed solutions (Countermeasure Cost, Risk Mitigation). As a result, the selected alternative is the one that provides the highest benefit to the organization (i.e., the highest RORI index).
- **Policy Enforcement Point (PEP):** it is implemented by the Security Database of the monitored scenario. The database is modified by the PyOrBAC Agent according to the security policies that are proposed by the PyOrBAC Engine in order to mitigate the attacks.

D.1.4 Use-Case Scenario

Figure D.1 depicts a typical use-case scenario where a regular user (a mobile subscriber who has an account with the Mobile Money Transfer Service) connects to the MMTS platform through the Front Office, using for this a phone number and its associated PIN code (1). The Front Office validates the user's credentials by checking up with the information stored in the security database (2). Similarly, the Account Management System (AMS) verifies the rights for users to perform transactions by consulting the database (3). A log is generated for each activity performed by each user (4).

The Alert Correlation Engine receives the logs generated in the MMTS Simulator (5) and provides alerts that indicate intrusions or attacks to the Decision Support and Reaction module (6). In case of attack, the alerts received by the PyOrBAC Engine are evaluated and the different security policies associated to the given attack are enforced by the PyOrBAC Agent (7). This latter injects the new security rules into the database, so that they are considered to authenticate users and grant permissions/prohibitions to perform transactions in the MMTS platform (8).

As a result, transactions are properly performed in a default scenario (when no attack is detected), and they are prohibited or restricted in a scenario of attacks.

D.2 Modelling MMTS Entities in OrBAC

Following the OrBAC formalism, we model 'MMTS' as the central organization, from which all the required definitions are done. Table D.1 summarizes the abstract entities (e.g., role, activity, view), as well as the concrete entities (e.g., subject, action, object) defined for the MMTS organization.

From Table D.1, a subject (e.g., s1) executes an action (e.g., connect) over an object (e.g., mmts_platform). In addition we define a set of contexts (conditions/constraints) that must be held to activate a given security rule.

In the MMTS organization, subjects (e.g., s1, s2, ..., s10) are empowered in the role "MMTS_USER". Similarly, the actions 'connect', 'd_auth', and 'account_surv' are considered in activities "CONNECT2MMTS", "DOUBLE_AUTH" and "SURVEILLANCE" respectively.

The actions 'n_emit_tr7', 'n_rcv_tr20', 'min_rcv_0', 'max_rcv_100' 'min_emit_0', and 'max_emit_100' are considered in the "NORMAL_TR" activity by the MMTS organization. In addition, the objects 'mmts_platform' and '*' are used respectively in views "PLATFORM" and "ANY".

D.3 MMTS Execution Context and Attack Reaction

The following subsections describe two MMTS contexts: default context and attack context.

D.3.1 Default Context

It is a nominal context in the MMTS organization, that is always activated by default (in the absence of attacks). A default context has the lowest priority on the system (equals to 1), meaning that an attack context is always treated with a higher priority.

Listing D.1 shows the abstract security rules that apply by default to all MMTS subjects. The negative sign (−) indicates a prohibition, while the positive sing (+) indicates a permission for a

TABLE D.1 - Definition of MMTS Entities

Entity	Description
Role	
MMTS_USER	It regroups MMTS users with the same rights and privileges
Activity	
CONNECT2MMTS	Connecting to the MMTS platform
SURVEILLANCE	Setting the user account under surveillance
DOUBLE_AUTH	Requesting double authentication
NORMAL_TR	Executing transactions in a normal scenario
N_NORMAL_TR	Executing transactions in a scenario different to the one by default
AT_TR	Executing transactions in an Account Takeover Attack (AT) scenario
View	
PLATFORM	It provides a view of the MMTS platform
ANY	It provides information of any entity
Subject	
s1,...,s10	Subjects of the MMTS platform
Action	
connect	It models the process of connecting to the MMTS platform
d_auth	It requests the user's PIN and birth date
account_surv	It activates account surveillance
n_emit_tr7	Maximum 7 transactions to emit per day
n_rcv_tr20	Maximum 20 transactions to receive per day
min_rcv_0	Minimum 0,01 euros to receive per transaction
max_rcv_100	Maximum 100 euros to receive per transaction
min_emit_0	Minimum 0,01 euros to emit per transaction
max_emit_100	Maximum 100 euros to emit per transaction
max_emit_30	Maximum 30 euros to emit per transaction
Object	
mmts_platform	It models the MMTS platform as an object accessible by users
*	It models the notion of 'all users'

role to perform an activity over a view in a given context.

LISTING D.1 - Abstract Security Rules in a Default Context

```
+,MMTS,MMTS_USER,CONNECT2MMTS,PLATFORM,Dft_ctx
+,MMTS,MMTS_USER,NORMAL_TR,ANY,Dft_ctx
-,MMTS,MMTS_USER,SURVEILLANCE,PLATFORM,Dft_ctx
-,MMTS,MMTS_USER,DOUBLE_AUTH,PLATFORM,Dft_ctx
```

The security rules shown in Listing D.1 grants the permission to all users of the MMTS to connect to the platform and perform normal transactions to any MMTS object. It also deactivates the surveillance and the double authentication of the user account. This derives the hold predicate shown in Listing D.2.

LISTING D.2 - Hold Predicate for a NORMAL_TR Activity

```
Hold(MMTS,-,-,-,Dft_ctx) <- True
```

The predicate shown in Listing D.2 holds for all MMTS subjects. From the abstract security rules shown in Listing D.1, we derive concrete security rules using the derivation rule defined in [Mie05]. An example of these rules is shown in Listing D.3.

LISTING D.3 - Concrete Security Rules for Subject 's1'

```
+,MMTS,MMTS_USER,CONNECT2MMTS,PLATFORM, Dft_ctx
Empower(MMTS,MMTS_USER, s1)
Consider(MMTS,CONNECT2MMTS, connect)
Use(MMTS,PLATFORM, mmts_platform)
Hold(MMTS, s1, connect, mmts_platform, Dft_ctx)<- True
-> Is_permitted (s1, connect, mmts_platform)
```

The example shown in Listing D.3 indicates that subject 's1' (empowered in the role MMTS_user), is granted the positive authorization to perform the action 'connect' (considered in the CONNECT2MMTS activity) to the object 'mmts_platform' (used in the PLATFORM view) in a default context.

D.3.2 Attack Context

It is a prerequisite context that triggers the selection of countermeasures. The Attack context is held if an MMTS subject has the AT attribute set to true. This context has some countermeasures associated to the type of attack with a priority defined by the RORI index (proposed in Equation 3.2), meaning that, the context with the highest priority is selected to react over a given attack. As presented in [GDJ+12], several attack scenarios may occur in the MMTS platform (e.g., Account Takeover, DoS, Money Creation/Destruction, etc); we select the Account Takeover Attack to explain the attack context in this section.

An account takeover is a password-based attack that exploits vulnerabilities on the user's side and steals the mobile user account to perform transactions in favour of the attacker. The complete evaluation process for this attack is described in Section 5.1. Results show that the optimal solution to implement is to reduce the transaction amount and to activate double authentication process (a combination of countermeasures C4 and C8).

Before defining the security rules to be applied in an Account Takeover Attack, we introduce two new OrBAC entities:

- **AT_TR:** An activity of the MMTS organization that regroups actions that activate a given countermeasure. In this use case, the action "max_emit_30" is considered in the AT_TR activity.
- **N_Normal_TR:** An activity of the MMTS organization that regroups actions that will deactivate some parameters of the NORMAL_TR activity in an attack scenario. In this use case, "max_emit_100" is an action considered in the N_Normal_TR activity.

Listing D.4 shows the security rules that apply to all MMTS users after detecting an Account Takeover Attack. Each security policy is associated to a particular countermeasure that results out of the previously described evaluation.

LISTING D.4 - Abstract Security Rules for an Account Takeover Attack

```
-,MMTS, MMTS_user, N_NORMAL_TR, ANY, AT_ctx (C4)
+,MMTS, MMTS_user, AT_TR, ANY, AT_ctx (C4)
+,MMTS, MMTS_user, DOUBLE_AUTH, PLATFORM, AT_ctx (C8)
```

The security rules shown in Listing D.4 deactivates particular parameters of a normal transaction by prohibiting the execution of some actions of the NORMAL_TR activity (this actions are grouped in a new activity N_NORMAL_TR). It also grants the permission to MMTS users to perform limited transactions (AT_TR) over any MMTS subject (C4) previous connecting to the platform using a double authentication method (C8) in an Account Takeover context (AT_ctx). The priority for the AT context is equal to 33,87 (the resulting RORI index for the combined countermeasure), which is higher than the one set by default.

The derived predicate for an AT_TR activity is shown in Listing D.5.

LISTING D.5 - Hold predicate for activity AT_TR

```
hold(MMTS, s, -, -, AT_ctx) <- s.attribute(AT)=True
```

If for instance, PyOrBAC Engine receives an alert indicating that subject 's2' is under an Account Takeover attack, the AT attribute of 's2' will be set to 'True'. As a result, 's2' is held by the "AT_ctx", which derives the concrete security rules expressed in Listing D.6.

LISTING D.6 - Concrete Security Rules for an Account Takeover Attack

```
Is_prohibited(s2, max_emit_100, *)  (C4)
Is_permitted(s2, max_emit_30, *)  (C4)
Is_permitted(s2, d_auth, mmts_platform)  (C8)
```

Notice that the default context is still held by subject 's2', therefore the concrete security rules expressed in Listing D.7 also apply to subject 's2':

LISTING D.7 - Concrete Security Rules for a default context

```
Is_permitted(s2, connect, mmts_platform)
Is_prohibited(s2, surveillance, mmts_platform)
Is_prohibited(s2, d_auth, mmts_platform)
Is_permitted(s2, min_emit_0, *)
Is_permitted(s2, max_emit_100, *)
Is_permitted(s2, min_rcv_0, *)
Is_permitted(s2, max_rcv_100, *)
Is_permitted(s2, n_emit_tr7, *)
Is_permitted(s2, n_rcvt_tr20, *)
```

As a result, transactions are restricted to a maximum of 30,00€ (C_4), previous a double authentication of subject 's2' (C_8), in an Account Takeover Attack.

D.4 Discussion

Combined attacks require a greater analysis and a higher amount of information to determine their volume and the coverage of the proposed countermeasures. The two case studies show the applicability of our model and the operations required to evaluate and select optimal countermeasures.

Our countermeasure selection model is integrated within an OrBAC formalism for the Mobile Money Transfer Service scenario. As a result, security policies are defined for the MMTS organization in a default and an attack context, which allows the definition of abstract entities (e.g., role, activity, view), as well as concrete entities (e.g., subject, action, object), making it possible to perform transactions in a default scenario (when no attack is detected), and to prohibit or restrict them in a scenario of attacks.

In a scenario of multiple attacks, it is possible that the optimal countermeasure for one attack enters in conflict with other countermeasures already deployed on the system or those proposed to mitigate other attacks. To solve conflict problems, we follow the approach proposed by Thomas [Tho07], using contexts to dynamically update security policies in particular circumstances. For instance, an operational context is said to be active in the absence of attacks or intrusion characterizing a threat. Other contexts define additional security rules to be applied when a threat or an intrusion is detected and specifies new security mechanisms to counter the detected threat. These security rules generally correspond to permissions (positive authorizations), prohibitions (negative authorizations) and obligations (constraints or restrictions).

Conflicts in the OrBAC model originate if a concrete permission (i.e., Is_permitted) and a concrete prohibition (i.e., Is_prohibited) are derived for the same subject, action and object [Mie05].

In the example described in Section D, two security rules are in conflict: the first one grants permission to all MMTS users to connect without double authentication (default context), however, the double authentication is activated for users that hold the AT context (AT_ctx = true). The second one, grants the positive authorization to emit transactions for a maximum of 100€ (action 'max_emit_100') to all MMTS subjects in a default context, however, this action is prohibited for users that hold the AT context, and a new action ('max_emit_30') is granted.

To solve these conflicts, the rule with the highest priority (highest RORI index) overrides those with the lowest priority context. As a result, since the AT context has the highest priority, we enforce the rules associated to this context and delete those that are in conflict and that have been implemented in the default context.