UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA


# A multi-level model for risk assessment in SIEM

Luis Miguel dos Santos Vilar Ferreira

**Mestrado em Segurança Informática**

Dissertação orientada por:
Profª. Doutora Ana Luisa do Carmo Correia Respício
e pelo Engenheiro Pedro Dias Rodrigues

2017

# Acknowledgments

First of all, I would like to thank Prof. Doc. Ana Luisa do Carmo Correia Respício and Engineer Pedro Dias Rodrigues because without your help this dissertation would not be possible to make. Both of you have taught me so much personally and professionally, from how it is to work in a company as big as EDP to how extensive and, at the same time exciting, the field of Risk is.

I would like to thank my colleagues from FCUL and EDP as well, you have helped me in so many ways that I cannot list them all. You have obligated me, directly and indirectly, to think differently, to improve myself daily, and to enjoy moments of relaxation.

The biggest and most special 'thank you' is for Mariana Donato. Your unconditional support, your patience, your way of being, your teaching and charisma were essential for me. Even so, we are no longer together, I will always be grateful for what you did for me and for this dissertation.

Finally, I am extremely grateful for having my family and their support. They have abdicated so much all over the years to be possible for me to have this opportunity, and I am grateful for the tolerance they showed during the most complicated times I had. They were impeccable.

Sincerely, Thank you all!

*Dedicated to my family... We finally did it!*

# Resumo

Os sistemas informáticos têm evoluído de forma tão exponencial que, hoje em dia, são considerados imprescindíveis para qualquer organização, pois são necessários para qualquer tipo de atividade dentro da mesma.

Uma vez que os sistemas informáticos estão longe de ser perfeitos, a nível de eficiência, eficácia e de segurança, a incorreção ou falha no seu funcionamento poderá provocar um impacto negativo nos negócios da organização.

Para auxiliar na detecção de problemas de segurança nos sistemas informáticos, existe uma tecnologia, designada de Security Information and Event Management (SIEM), que permite fazer uma monitorização e gestão dos eventos e utilizar essa informação com o intuito de identificar anomalias nos próprios sistemas.

Normalmente, estas anomalias são de segurança, desde ataques realizados à organização com intenções maliciosas, até problemas de configurações que poderão potenciar problemas severos de segurança. Para tais anomalias serem detetadas, os SIEMs utilizam filtros e regras para detecção de padrões de comportamento que a organização considera como um padrão malicioso ou um padrão fora do comportamento normal.

Embora os SIEMs já consigam detetar padrões complexos de forma eficaz, esta tecnologia tem ainda lacunas em certos aspectos, nomeadamente na apreciação do risco nos ativos da organização. Atualmente, os SIEMs apresentam processos de apreciação de risco muito elementares, dando apenas uma perspetiva ao nível dos ativos onde ocorrem os eventos e os incidentes. Geralmente, estes ativos são as máquinas e servidores físicos, o que implica que a visão do impacto dos eventos e incidentes na organização seja mais complicada de conceptualizar, principalmente a níveis mais abstratos, como por exemplo, qual o impacto sofrido pelas aplicações que essas máquinas ou servidores físicos suportam, e o consequente impacto no negócio.

O projeto europeu *DiSIEM*, *Diversity in Security Information and Event Management*, tem como objetivo desenvolver soluções que sejam adaptáveis a vários tipos de SIEM que existem no mercado, sem existir a necessidade de alterar esses SIEMs. Com este tipo de abordagem, é-nos permitido evoluir os SIEMs existentes em vez de os substituir ou alterar, reformulando toda a estrutura e mecanismos, e evitando novos ciclos repetitivos de desenvolvimento. A tecnologia desenvolvida neste projeto será adaptável a vários SIEMs

e de forma mais acessível, pois uma vez que as soluções SIEM são maioritariamente comercializadas.

Esta dissertação apresenta uma proposta de um modelo multi-nível para apreciação de risco em SIEMs. Para aplicação e validação do modelo em ambiente industrial, estabeleceu-se uma colaboração com a Energias De Portugal (EDP), que é um dos parceiros industriais no projeto DiSIEM.

Este modelo traz uma inovação ao nível da importância que os SIEMs poderão ter numa organização, através da análise dos incidentes que os SIEMs captam, das vulnerabilidades que os sistemas informáticos podem possuir, e das relações que existem entre os ativos dos sistemas informáticos. A inovação deve-se ao facto de o modelo ser multi-nível, considerando uma divisão hierárquica entre máquinas, aplicações, e serviços, de forma a ser possível obter diferentes visões do estado de risco atual de um sistema.

Ao nível das máquinas, todos os ativos que sejam considerados como computadores pessoais, servidores físicos, routers, firewalls, entre outros, e que tenham uma presença física, são colocados neste nível. Já ao nível das aplicações, representam-se os ativos aplicacionais que sustentam o negócio da organização, mesmo não estando estes diretamente acessíveis por clientes. Por fim, o nível dos serviços tem como objetivo representar ativos abstratos que caracterizam ações e funções de determinados conjuntos de aplicações, produzindo assim, uma visão holística do estado e do comportamento dos conjuntos de aplicações. Com esta visão holística do estado dos sistemas informáticos e serviços, é permitida uma melhor compreensão da parte de gestores de topo relacionados com o negócio da organização.

Os diferentes níveis são intrinsecamente ligados, onde o nível dos serviços é dependente do nível das aplicações, e consequentemente, as aplicações são dependentes do nível das máquinas, uma vez que estas suportam o funcionamento das aplicações. Podem ainda existir dependências dentro do mesmo nível, isto é, uma aplicação poderá depender de outra, tal como no nível das máquinas pois há máquinas que suportam outras. Contudo, no nível dos Serviços não se encontra este tipo de dependência, pois cada serviço tem um âmbito bem definido e independente.

A própria apreciação do risco é realizada por ativo com base num modelo comum e generalista a todos os tipos de ativos. O modelo considera três componentes: Vulnerabilidades, Dependências, e Incidentes.

A variável das Vulnerabilidades representa o impacto potencial dos problemas de segurança que um ativo pode ter no sistema em termos das vulnerabilidades conhecidas e as respectivas classificações de severidade. Já a variável das dependências permite integrar o impacto dos problemas de segurança que outros ativos relacionados com o ativo a ser avaliado, poderão provocar. Por fim, a variável dos Incidentes é a variável que quantifica o impacto de incidentes detetados a partir de eventos do SIEM.

Como cada variável pode ter um mecanismo de avaliação diferente, pois não existe

uma forma pré-estabelecida de o fazer, criámos três processos distintos de apreciação de risco com base no modelo generalizado e comparámos os seus resultados.

A ferramenta que implementa o modelo é constituída por três elementos: Base de Dados, Aplicação, e Dashboard.

A Base de Dados é o elemento onde é guardada toda a informação necessária para fazer a apreciação do risco, desde dos dados dos ativos e as suas características, detalhes das vulnerabilidades e incidentes, dependências entre ativos, até às configurações dos parâmetros para as várias versões do modelo. Já sobre a aplicação, esta irá proceder à apreciação do risco através da informação extraída da base de dados.

Na aplicação, são calculadas as fórmulas com informação recolhida da base de dados própria para o modelo. Toda a informação obtida para o preenchimento da base de dados foi assente em listas de ativos e vulnerabilidades, e análises detalhadas dos incidentes criados pelo SIEM que a EDP possui.

Por fim, o Dashboard é o elemento que permite visualizar a informação sobre o risco de todos os ativos, divididos consoante os seus níveis, e as respetivas dependências. Com esta nova abordagem de disponibilizar a informação, o especialista de segurança tem o seu trabalho facilitado ao analisar os resultados, o que até agora não era possível.

Muitos gestores de topo não têm conhecimentos técnicos sobre sistemas informáticos, não entendem o que realmente uma vulnerabilidade é, ou qual o custo para a organização de uma aplicação crítica não estar a funcionar, e, muito provavelmente, esses mesmos gestores não terão tempo para aprender.

A disponibilização deste modelo permite aos diferentes níveis de gestão, operational e de negócio, avaliar o risco em várias camadas para os respectivos gestores das mesmas conseguirem ter uma percepção dos risco dos ativos que são responsáveis. O modelo permite ainda facilitar a comunicação entre os diferentes gestores e a comunicação entre as equipas do centro de operações de segurança (*SOC*) e os donos dos ativos.

Ao apreciar o risco ao nível dos serviços, é estabelecido um *common ground* com os gestores de topo, visto que o seu foco são os serviços e que os mesmos têm uma importância na estratégia para o negócio da organização.

A aplicação permite identificar as máquinas, aplicações e serviços com um risco mais elevado, e desta forma, reportar os resultados já avaliados para uma tomada de decisão informada. Assim, é possível prioritizar os ativos que necessitem de uma correção mais urgente.

Também neste trabalho foram analisados, e comparados entre si, os resultados das várias versões do modelo, de modo a perceber quais as vantagens e desvantagens que cada uma tem.

**Palavras-chave:** SIEM, Análise de Risco, Modelos de risco, Avaliação de risco

# Abstract

Security Information and Event Management (SIEM) is a system to monitor IT elements of an organization and to detect security anomalies based on events produced by the same elements. This type of system has grown exponentially throughout the years and currently, they can detect complex patterns of behaviors and assess the impact of the anomalies detected. Nonetheless, this type of system is far from being perfect presenting a considerable number of flaws being one of them, the risk evaluation process.

Currently, risk evaluation in SIEMs has a perspective too operational and low-level, meaning that the evaluation is made for each event in assets such as physical servers. At this level, only the security operation centers can understand the possible consequences of the anomalies to the organization. Consequently, at a strategic and business levels of the organization, it becomes difficult to the C-level managers to realize the current state of the system, especially because they are focused on the business perspective.

Nowadays the business of an organization is too dependent on IT systems, which leads to the necessity to assess the risk that these systems have. However, this assessment needs to be done at a higher level of abstraction from the operational details to be understandable for the managers at different levels.

In order to establish a better communication between IT managers and C-Level managers, and to obtain a high-level assessment of the IT systems security status, we propose a multi-level model for risk assessment in SIEM. The model is divided into three layers: hosts, applications, and services. Each of these layers has a different perspective, being an operational perspective the hosts' layer and a business perspective the services' layer. The model also provides three versions to assess the risk which are analyzed and compared between each other. The risk assessment is done based on the assessment of vulnerabilities severity, the risk of dependencies, and incidents severity that each asset has.

**Keywords:** SIEM, Risk Management, Risk Models, Risk Evaluation

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Context

This work is part of the H2020 project Diversity in Security Information and Event Management (DiSIEM) funded by the European Commission [1]. This project addresses the limitations of current SIEM solutions, aiming to enhance them by extending their capabilities and features, instead of creating novel architectures or unnecessary modifications. The approach of extending existing SIEM technologies allows the materialization of a set of tools and components that can be used as plug-ins for the SIEMs themselves. This will provide a more diversified set of solutions for the inherent limitations of SIEMs. The main purpose of DiSIEM is to approach and improve five different topics of the state of the art of SIEMs. These topics are the integration of diverse OSINT (Open Source Intelligence) data sources available on the Internet, the creation of new security risk models and metrics, the development of new methods of data visualization, the integration of diverse, redundant and enhanced monitoring capabilities on SIEM technologies, and the use of cloud providers for archiving long-term data.

In particular, this work deals with the second topic, i.e., security models and metrics. We propose a multi-level risk assessment model to support security analysts on their decisions and to facilitate the communication between the Security Operations Center (SOC) of an organization with its top managers.

This work results from a collaboration between Faculdade de Ciências da Universidade de Lisboa (FCUL) [2] and EDP – Energias de Portugal [3], which are two of the parties in the DiSIEM project. Being one of the most advanced of such systems in Portugal, EDP's SOC was fundamental for the conceptualization of the model proposed since it provided not only examples of real systems and obstacles, but also of real processes of incident solving, vulnerability remediation, and risk management, which allowed us to create a realistic model.

## 1.2 Motivation

Throughout the years, SIEM systems have become an increasingly essential part of an organization. Before the deployment of this type of systems, security experts had to analyze every devices' logs manually, which was an extremely inefficient and time-consuming task.

Early SIEM systems provided the possibility of having all available information on collected logs stored in a unique place. Then, it became possible to filter such information, to create an investigation by correlating the gathered data with patterns of behavior defined by the organizations, and to store data for a vast period of time. Nowadays, SIEMs are capable of using Artificial Intelligence to detect incidents, correlate information, and perform risk evaluation. This shows how much SIEMs have improved.

Despite being already integrated with deployed SIEMs, current risk evaluation process still remains basic and in need of improvement. In most solutions, this process only assesses events that occur on low-level assets, e.g., a machine or a load balancer. This makes the SOC team the most suitable team to understand the impact that events can have on assets, and consequently, on the entire system, in an operational perspective.

Risk evaluation has become an essential task for organizations, mainly due to the increasing number of threats they are exposed to, caused by the so generalized dependency on technologies of information and on the Internet. This task requires collaboration between IT managers and business managers. Unfortunately, there's a gap between these two parts, mainly due to the existence of different ideas between IT stakeholders and business managers.

C-Level managers are managers with a high level of influence in the strategy of the organization, in the high-stakes decisions as well as in the responsibility to ensure the day-to-day operations aligned with the strategic goals. Once they have a strategic role in the organization and they deal with assets with a level of abstraction much higher than stakeholders related with IT matter, meaning that C-Level managers usually do not have a clear insight of the state of the information systems, especially if they have to rely on events occurred in a specific machine or load balancer in the organization.

It is important to consider that the general structure of an organization has several stakeholders for each level of management, which leads to a deep communication gap, a layered gap because the communication between IT staff is not direct with C-Level managers. Since it is a layered gap, in order to facilitate the communication between the IT department and the C-Level managers, all intermediary stakeholders have to understand their roles in the process as well.

On the other hand, the capacity of current SIEMs is sufficiently enough to improve the risk evaluation process of an organization, enhancing the importance of the SIEM technology in an organization as well. Since the capability of detecting abnormal patterns is no longer a problem, the necessity of not changing the structure and how SIEMs behave

becomes a matter of concern, once we try to create a solution that will diversify and to be adaptable with the current solutions of SIEMs.

Since the SIEM does not have the capacity to assess risk in multi-level and the gap of communication between IT staff and C-Level managers is notable, new approaches have to be considered to solve these problems.

## 1.3 Objectives

The objectives of this work are to conceive a multi-level model for risk assessment in SIEM and to create a tool in order to implement this model.

With this model and tool, we aim at: an independent decision-making process between levels, allowing to determine which assets have the biggest issues level by level creating a better insight of the organization status; facilitating the communication between the different levels of decision makers, including the IT staff and C-Level managers; a support to the decision-making process in each level, by supplying the risk score of each asset; could give more information for rules of SIEMs, allowing the monitoring of the traffic in applications, instead of individual machines; and, for each asset owner, an assessment of the respective assets.

## 1.4 Planning

This dissertation was divided into five main tasks. In task 1 - Review scientific literature, we analyze scientific literature and wrote the preliminary report.

The task 2 - Classify EDP's Information systems, where this task comprehends all sub-tasks related to identifying assets and their respective vulnerabilities and incidents from the EDP's systems in order to have real data to test the model.

In task 3 - Conceptualization of a risk model, we developed a model to assess the risk of the assets found on the previous stage. After the first four months of development of the model, we started the Test the model created stage simultaneously and adapting the tool accordingly with our conclusions about what should change in it.

Lastly, task 5 - the writing of the thesis was initiated, where we wrote this dissertation.

The tasks were fulfilled over 10 months and, during that period of time, no considerable changes had to be made to how the project was planned. Nevertheless, some tasks last longer than we expected due to several complications in terms of configurations and changes needed during the Testing of the model task.

Figure 1.1 exhibits the tasks and the respective expected time to finish it as well as the actual time needed.

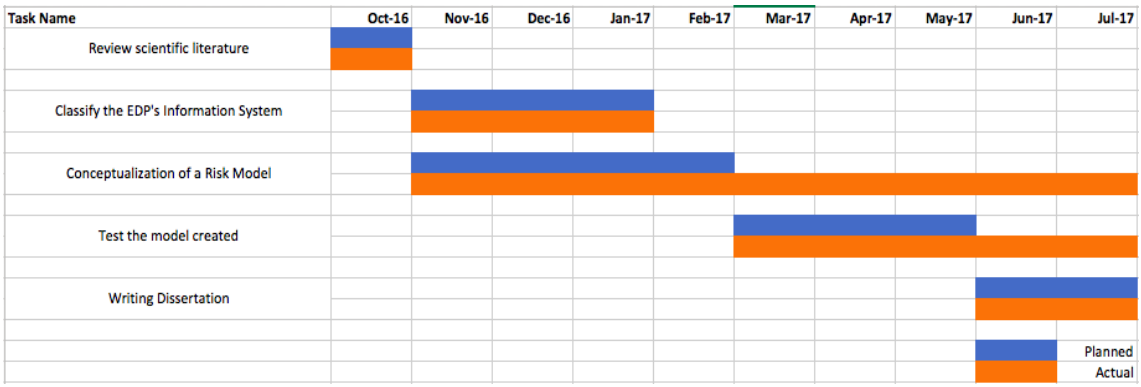



Figure 1.1: Planning of this dissertation

## 1.5 Structure of the dissertation

The structure of this dissertation is composed as follows.

Chapter 2, Related Work, details which technologies and methods have served as a base to conceptualize the model that will be presented further.

Chapter 3, A Multi-Level Model for Risk Assessment in SIEM, describes the Model for Risk Assessment in SIEM introducing its design and the computational risk scores.

Chapter 4, Model Implementation, describes a global vision of how the information is processed, how it is stored, how it is used to assess the scores of the assets. Also in this chapter, it is explained the structure of the application that assesses the risk, and the dashboard used to display all information about the assets and the respective scores.

Chapter 5, Results and Discussion, presents and discusses the results obtained in order to prove the concept of the model and its usefulness for the organizations.

Lastly, the Conclusion chapter contains a resume of all the dissertation, highlighting the main characteristics of the model and its implementation.

# Chapter 2

# Related Work

We begin this chapter by introducing the risk assessment process and how threats and assets of an organization are analyzed and evaluated, standards that structured this process as well as which organizations did create them. Then, we characterize the general structure of a SIEM and the current solutions, namely, AlienVault, HP ArcSight, and IBM QRadar. Finally, we review scientific literature related to the improvement of the communication among different stakeholders of an organization, the impact of an asset could have in other assets regarding their risk, and a framework to assess the severity of software vulnerabilities.

## 2.1 Risk Assessment Process

All organizations have their own purposes, but one thing that they have in common is the necessity of having assets. Those assets can have multiple characteristics such as tangible, intangible, intellectual, IT related, physical infrastructure, and so forth.

However, if the asset really has a value to the organization, it will always have a risk associated, which may or may not, cause severe problems to the interest of the organization. To be able to prevent, identify or react to situations that jeopardize the organization, it is mandatory to have a risk management process.

The Risk Management is a cyclic process that grants assessment and control strategies for potential threats, regardless of the type and nature, to a system or organization.

The process of assessing the risk is an overall process based on analyzing and evaluating both threats and adverse situations. Here, analyzing consists in identifying those hazards, determining how frequently they can happen and what consequences they may cause, conceiving a clean perception of the risk inherent to an asset or organization.

There are three methods of how to practice the risk analysis process. It can be in a qualitative, quantitative or a semi-quantitative way.

The difference between qualitative and quantitative methods is based on the scale. The Qualitative method uses a scale of qualifying attributes (e.g., Very Low, Low, Medium,

High, Very High), while the Quantitative uses a numerical scale (e.g., 0,1,2,3,4,5,6,7,8,9) to define the possible consequences and their likelihood of happening. A semi-quantitative method uses a scale in which a range of numerical values match to one single qualifying attribute (e.g., 0 and 1 match Low, 2 and 3 match Medium, 4 and 5 correspond to High) [4] [5].

Even with similarities, there are some advantages and disadvantages of each approach, as shown in Table 2.1.

| **Method** | **Advantage** | **Disadvantage** |
|---|---|---|
| Qualitative | Simple<br>Agile<br>More understandable | Inexact<br>Partial information treatment |
| Quantitative | More precise<br>Complete information treatment | More Complex<br>More vulnerable to errors in treating information |

Table 2.1: Advantages and disadvantages of qualitative and quantitative methods

Evaluating the threats and adverse situations is a process to prevent and tolerate, or accept, the risk, taking into account factors such as socioeconomic, and environmental promoting decisions based on a comparison with the risk acceptance criteria, that the organization considers the maximum level of risk. A set of multiple possibilities of risk-reducing measures is created to control the risk of hazards that may occur [6].

The Control process is where the decisions about what mechanisms and criteria will be used are selected, implemented, monitored and communicated. Those decisions are based on three distinct types, Deterministic, Risk-Based and Risk-informed. The first one is a decision type that does not consider the threat likelihood to happen. The second one is based on the quantification of the risk, associated costs and the benefits for comparison between all measures available to select the best one, especially with a low budget. The last one is a decision type represents a philosophy whereby risk insights are considered together with other factors.

In terms of types of controls, or mechanisms, they consist on an answer to determine a situation involving an exploitation of the vulnerability. There are five, Defense, Transferal, Mitigation, Acceptance and Termination [6].

The defense control aims to prevent the exploitation of the vulnerability. Transferal control has the objective to change the risk of an asset to another asset or not. Insurance policies are an example. Mitigation control focuses on reducing the damage caused when the vulnerability is exploited. Acceptance control is a decision to not do something about the risk taking into account the consequences of the outcome of an exploited vulnerability. Finally, termination control attempts to eliminate the vulnerable asset after an assessment of its importance.

The mechanisms chosen to be applied rarely eliminate the risk of a threat completely, leaving residual risk. The residual risk is a portion of risk which can not be eliminated by the mechanisms or measures selected due to several factors like cost, necessity, feasibility, or others.

For a better understanding of the risk management process, Figure 2.1 indicates the main phases of the process.

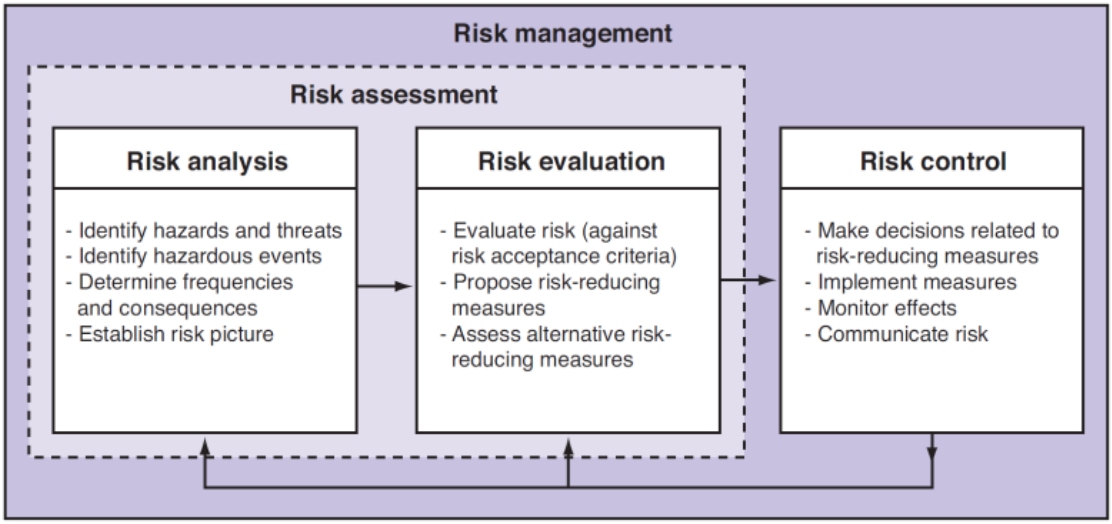



Figure 2.1: Risk Management Process extracted from [7]

The risk management process is crucial to be embedded in the culture and processes of an organization, especially now that is the digital era. Today, every organization relies on IT systems for treating information, for dealing with customers and other organizations, basically, for their purposes. Being that they have such dependencies, risk management has a critical role in protecting an organization's information assets, and therefore its mission, from IT-related risk.

Due to risk management being so complex and important, there are entities specialized in developing standards of how to implement such processes, being one of them the International Organization for Standardization (ISO) [8].

### 2.1.1 International Organization for Standardization (ISO)

The International Organization for Standardization (ISO) was founded in 1947 in Geneva, Switzerland and, since then, it creates international standards for Technical, Classifications and Procedural Norms in multiple fields like quality, project, incident, risk management, so on [8].

Specifically related to risk management and IT risk, ISO created several standards such as the ISO 31000 - Risk Management [9] and the ISO 27000 series [10].

**ISO 31000 - Risk Management**

The ISO 31000 - Risk Management standard provides general guidelines about risk management which can be applied to a private or public, association, or group and is not specific to a certain industry or sector at all [9]. It can be applied to all kinds of risk as well and has a main goal to harmonize the processes of risk management with present and future standards, for example, the ISO/IEC 27005 - Information Security Risk Management [4], a standard from the ISO 2700 series.

This standard introduces an architecture of the risk management process with three main components, the principles, the structure to manage the risk management process and the risk management process itself.

The Principles component has the responsibility of providing 11 principles that allow the risk management process to be more effective and present in and for the organization.

Once the principles are defined, it is possible to illustrate the structure to manage the risk management process. The risk management process has 5 composing phases.The first one is the communication and consultation phase. This phase has the purpose of not just communicating with all internal and external interested parts of the process but also has to occur in all other phases. In order to be present in all phases, the plans for communication and consulting have to be enacted in the structure of the risk management process or more specifically, in the conceptualization segment.

The establishing context is the second phase, where the organization states objectives for the risk management processes, parameters for external and internal context while managing risk, scope and risk criteria, even when criteria have to be imposed by laws and regulations belonging to the organization, district or country.

The third phase is the risk assessment process. This process, as mentioned before, is an overall process of identifying, analyzing and evaluating risk.

The risk treatment consists in selecting controls to be applied in order to reduce the risk, as mentioned previously.

Finally, the fifth phase is the monitoring and reviewing phase. In this phase, all the phases are monitored aiming to assume that the control is efficient, enhance risk assessment, learn from the events occurred to identify future risks and review the scope and parameters from the establishing context phase. Figure 2.2 describes the structure of the ISO 31000 - Risk Management according to [9].

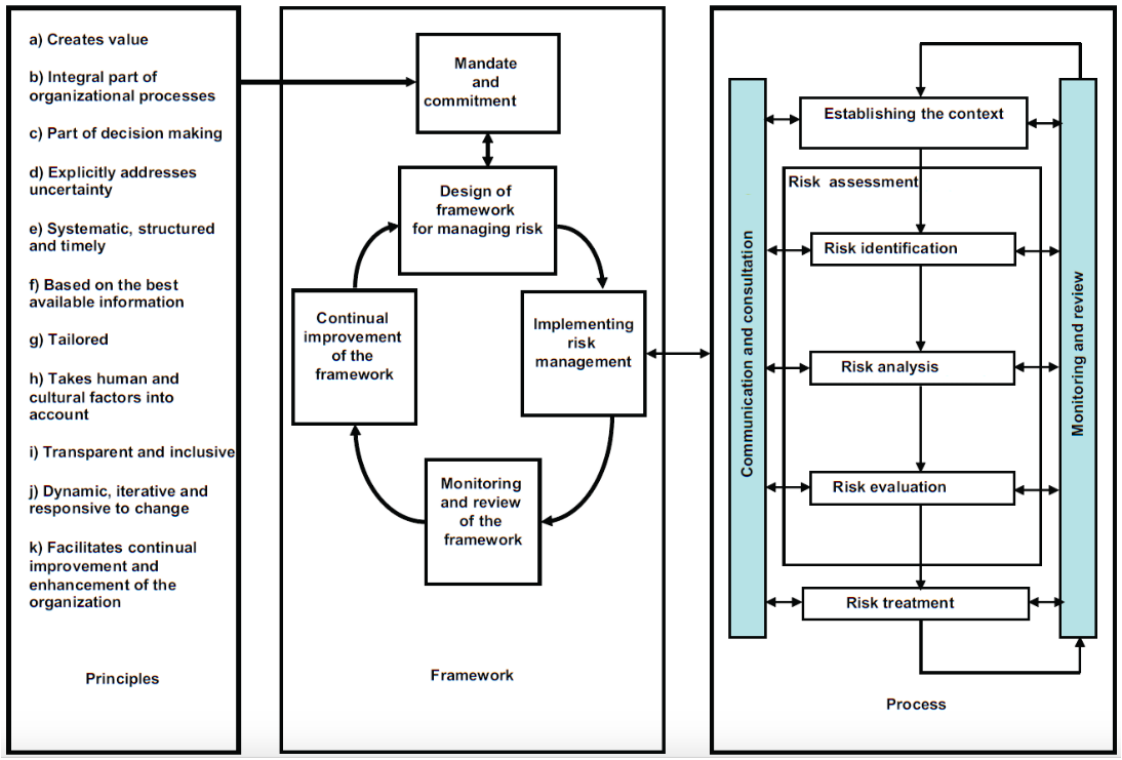



Figure 2.2: ISO 31000 Risk Management structure extracted from [9]

**ISO 27000 Series**

The ISO 27000 Series is a set of standards created by ISO and IEC, International Electrotechnical Commission [11]. These standards have different aims, covering the implementation, metrics, and the risk management of an Information Security Management System (ISMS). The standard related to the risk management is the ISO/IEC 27005 - Information Security Risk Management [4]. This standard does not provide any specific methods to apply the risk management process, it indicates what kind of actions have to be done in each phase of the process, emphasizing more the establishing context, the risk assessment and the risk treatment phases than ISO 31000 - Risk Management.

Figure 2.3 displays all the information security risk management process supported by this standard.

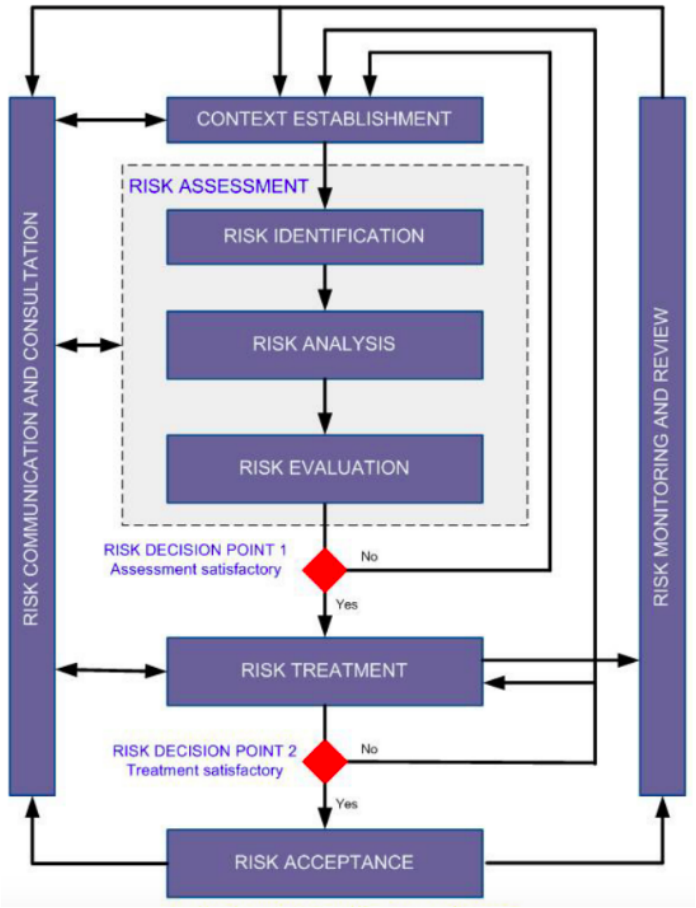



Figure 2.3: ISO/IEC 27005:2011 Risk Management Structure extracted from [4]

The main reason to follow a study on this standard is the process of risk assessment indicated in the Figure 2.3. This process has three main steps: risk identification, risk analysis, and risk evaluation.

The purpose of risk identification is to determine what could happen to cause a potential loss and to gain insight about how, where and why the loss or risk might happen or be, whether or not the source of the risk is under control of the organization.

The Risk Analysis is the process of scoring each risk based on methodologies, assessment of the consequences, and the incident likelihood as well, defining a level of risk for the asset assessed.

Finally, Risk evaluation is a process that relates the level of risks, estimated in the risk analysis process mentioned above, with the risk evaluation criteria and the risk acceptance criteria from the establishing context phase. Risk Acceptance criteria consists in the criteria to accept a risk on an asset, meaning that a organization is willing to have the risk on that asset, being necessary to monitor and review it periodically instead of reducing or mitigating the risk. This acceptance of risk in organizations is is a standard choice mostly due to the financial implications required to reduce or mitigate the risk.

These criteria are important to define the limits of each organization and to use them

in a relative scale for the model that will be presented in this work.

## 2.2 SIEM

Security Information and Event Management, or SIEM, is a tool that collects and correlates events occurred in the IT structure with rules and alerts the security experts of the organization in order to be possible to detect deviations from the normal behavior of the IT structure. This type of tool allows the collection, normalization, filtration, aggregation, correlation, and management or visualization of the data received (designated as log event or notification), in a physical centralized way, from technological components (server, firewall, IDS, router) in near real-time.

For a better comprehension about SIEM, it is possible to separate in phases what the tool can do. Figure 2.4 displays the structure of a general SIEM.

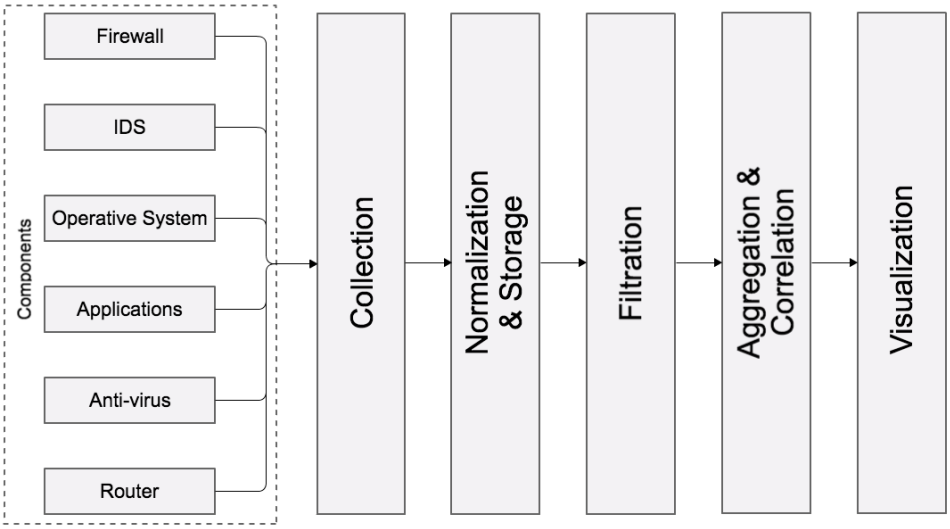



Figure 2.4: General SIEM Structure

The collection phase allows to obtain data from the technological components through connectors. A connector can be software or hardware based and is a component that interacts with the SIEM providing the data from the technological component where the connector is installed.

Since there are different types of components, such as firewalls, routers, operating systems, IDS, it is implicit that there are different structures to represent data. In order to establish a common pattern between all possible representative structures of data, there must be a phase to normalize all the data gathered and preserve it in an appropriate and secure database.

After normalizing all the data, it is required to remove all the unnecessary data with the purpose of managing all the information created. To be capable to remove the unnecessary data is necessary to pass through a filtration phase based on specific parameters and rules.

After filtering the data, it is necessary to aggregate and correlate it with the highest possible number of events and components in order to detect any kind of anomaly in the system.

At last, the visualization component materializes an understandable view (e.g., graphics) of all processes done to the information.

With all the data collected, normalized, filtered, aggregated and correlated it is possible to monitor, alert on, respond to, report, analyze, audit and manage security events of an organization including a risk scoring process.

The process of risk scoring in SIEMs can vary between scoring events, as AlienVault [12] and ArcSight [13] do, or scoring assets based on their vulnerabilities, as the IBM QRadar does [14]. However, in order to accomplish a score in all solutions that will be discussed further, it is necessary to provide crucial information about the criticality of the assets and other attributes.

### 2.2.1 Alien Vault

AlienVault is a SIEM solution, created by the AlienVault enterprise, and is divided in three main components. These three main components are the Sensor, the Logger, and the SIEM. There are two types of SIEM provided by AlienVault: Open Source Security Information and Event Management (OSSIM) [15], which is free, and the Unified Security Management (USM) [16], which is the most complete one and, for that reason, we decided to review it.

The AlienVault USM is available in software or hardware appliance and has a deployment flexibility that allows to have an all-in-one or divided implementation according to the needs of the organization.

The following sections will present the structure of the AlienVault SIEM solution and, right after, a description of the structure divided by the existing components, and some limitations of this solution.

#### 2.2.1.1 AlienVault Architecture and Components

Figure 2.5 exhibits the architecture of AlienVault SIEM [12], which includes three main components: the sensor, the logger, and the SIEM (component).

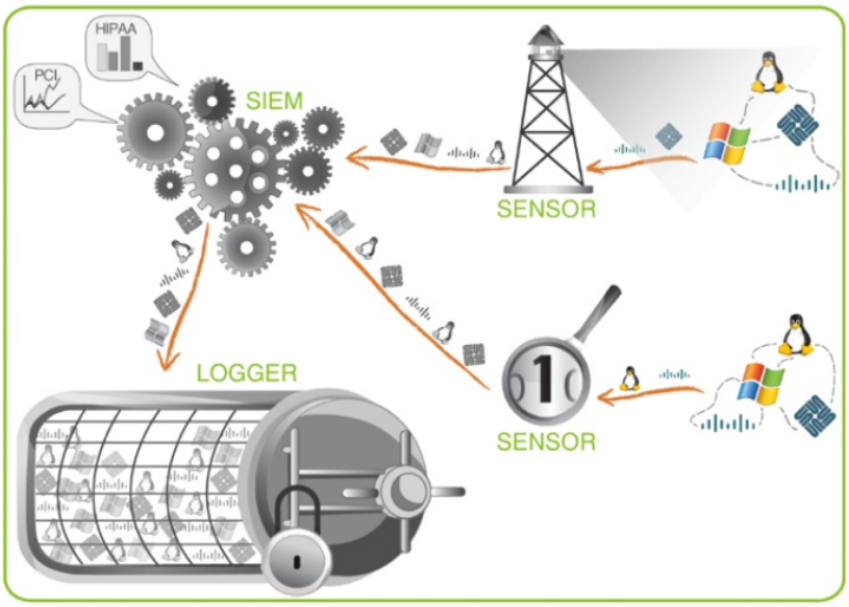



Figure 2.5: AlienVault SIEM's Architecture extracted from [12]

The Sensor component consists on a stipulated number of technologies aggregated in one single device. The Sensor is installed on network fragments to be able to inspect all possible traffic occurred in the organization, providing services, in near real-time, such as:

- Intrusion Detection
- Anomaly Detection
- Vulnerability Detection
- Discovery of inventory and network profiling systems
- Aggregate and normalize events with the purpose to send to SIEM and/or Logger

In order to obtain those events or data, it is necessary to have a connector as mentioned previously. The AlienVault SIEM solution gives an extended list of connectors (approximately 2000), and also flexible methods of gathering content. Some examples of those methods are Syslog, SNMPv2, FTP, Samba, and NFS.

The Logger component has the aim to preserve, in a secure way, a huge percentage of events occurred and detected by the Sensors in their not normalized state or raw state. It is required to be in a raw state due to being admissible as evidence for judicial purposes and to be possible to reassess hereafter [12].

The SIEM component provides the Security Intelligence activities related with correlation, risk evaluation, policy management, visualization, and reporting. The risk evaluation will be discussed separately due to being the spotlight of all the study realized of this SIEM [12].

The correlation process in AlienVault SIEM Solution can be done by three methods: Logical Correlation, Inventory Correlation, and Cross Correlation. The Logical Correlation main purpose is to assess if there is a security event that is a threat or if it is just a false positive to the organization. The Inventory Correlation has the purpose to correlate the goal of an attack and the asset attacked to realize if the attack occurred is or is not a threat to consider. Finally, the Cross Correlation allows prioritizing the events occurred based on the Sensors and vulnerability scans results.

The policy manager allows to adapt the system's behavior to determine specific situations. For instance, risk equation remodeling to a concrete event, redesigning the correlation process and other possibilities.

About the visualization and reporting, AlienVault USM SIEM uses a Web Interface where all information is available, including reports created to provide all the necessary information to manage and audit the system.

#### 2.2.1.2 Risk Score Evaluation

Risk score evaluation in AlienVault USM SIEM is done for each event and the parameters, given by the security expert, consist on the asset value, the priority, as well as the reliability of the data used to identify the attack. The risk score is established on an integer scale of 0 to 10 based on Equation 2.1 [17]:

$$Risk = (ASSET\_Value * PRIORITY * RELIABILITY)/25 \tag{2.1}$$

Where $ASSET\_Value \leq 5$, $PRIORITY \leq 5$, $RELIABILITY \leq 10$.

The $ASSET\_Value$ parameter is specified using an integer scale between 0 and 5. Unfortunately, it seems that AlienVault does not have a particular method or suggestion of how to classify the variables aforementioned. Due to this fact, there are some aspects to be understood about AlienVault USM SIEM on evaluating an asset.

The first aspect, when the AlienVault USM SIEM is evaluating the risk score of an event, it seeks the manually inserted value of the asset in question. If the value is not inserted, AlienVault will use the value assigned to the network where the asset is. The values of the networks and respective components are inserted in the AlienVault by a collaborator of the organization. Henceforth, the SIEM will assume that the value of the asset is the network value until the security expert changes it. It could happen that the asset does not belong to a network or it is just not possible to determine another value and, in that case, it will be used a default value, 2.

The second aspect to consider is that for an event having multiple assets involved, the SIEM will use the asset value from the most valuable asset, even if the most valuable asset has a default calculated value.

The $PRIORITY$ parameter focuses on the nature and impact that a threat can make to the organization. In terms of scale, it has integer values between 0 and 5 as it can be

seen in Table 2.2.

| Level Number | Qualitative Description |
|---|---|
| 0 | No Important |
| 1 | Very Low |
| 2 | Low |
| 3 | Average |
| 4 | Important |
| 5 | Very Important |

Table 2.2: AlienVault Priority scale

Lastly, the $RELIABILITY$ parameter relies on identifying the attack and its likelihood to happen. The scale for this parameter differentiates from the other ones, this one goes from 0 to 10 with steps of 1 as shown in Table 2.3.

| Level Number | Qualitative Description |
|---|---|
| 0 | False Positive |
| 1 | 10% chances of attack |
| 2 | 20% chances of attack |
| 3 | 30% chances of attack |
| 4 | 40% chances of attack |
| 5 | 50% chances of attack |
| 6 | 60% chances of attack |
| 7 | 70% chances of attack |
| 8 | 80% chances of attack |
| 9 | 90% chances of attack |
| 10 | Real Attack |

Table 2.3: AlienVault Reliability scale

#### 2.2.1.3 Limitations

According to [18], AlienVault solutions provide too basic statistics and problems with the generation of alerts on the NetFlow component. NetFlow is a protocol created by Cisco that can be integrated with the connectors in SIEM [19].

### 2.2.2 ArcSight Solution

The ArcSight [13] is a SIEM solution created by Hewlett-Packard (HP) [20]. The ArcSight SIEM Platform is an integrated set of products for collecting, analyzing, and managing enterprise event's information.

The architecture could vary, depending on how many supplemental modules are added.

The following sections will present a general architecture of the ArcSight SIEM solution, a description of the existing components and some limitations of this solution.

#### 2.2.2.1 ArcSight Architecture and Components

Figure 2.6 displays an ArcSight SIEM architecture divided by each main component.

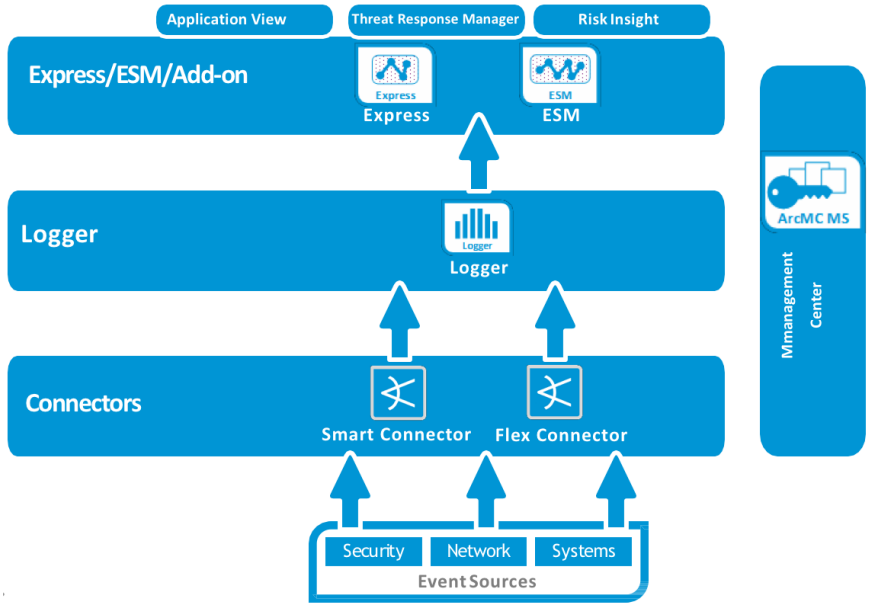



Figure 2.6: General ArcSight SIEM Architecture

Each component is assigned to a phase. Fundamentally, it is exactly the same to talk about the phase or the component.

The connector component is a software that collects events from end-point devices, normalizes the events, then sends processed data to the Logger component or to the ESM/Express.

There are two main types of connectors, Smart Connectors, and Flex Connectors. The Smart Connectors are more standard components than the Flex ones. Both operate in the same way but the flex connectors are adaptable to specific requirements needed for each organization or situation [21] [22].

The Logger component has an aim to consolidate and store, in a secure way, all the events caught by the connectors. Then, it forwards the log data allowing to normalize and analyze that data [23].

The ArcSight Express (Express) or the Enterprise Security Manager (ESM) are SIEM systems that correlate data (from the Logger component or the Connectors), monitor users, flows and applications, and provide visualization of threats. The main difference between these two types of SIEM is the capability or scalability of the system. The Express is more adaptable for small-medium size organizations, while the ESM is for larger systems.

The Add-on component is a set of applications that may help the correlation process, such as, the ArcSight Management Center [24], the Threat the Response Manager [25], the Application View, and the User Behaviour Analytics [26].

The ArcSight Management Center, or ArcMC, is an application to simplify the management of all events by implementing a centered and automated management and can reduce the resource requirements for the SIEM.

The Threat Response Manager allows managing threats as incidents occurred in the system, which is very similar to the Application View.

There are several add-ons that can be applied to the system. Each time one is added, the system might be more complex, however, it could improve the result and the implementation of the SIEM.

#### 2.2.2.2 Risk Evaluation

Risk evaluation on ArcSight is based on a priority formula computed for each event generated [27] [28].

The priority indicates if an event has a higher or a lower priority to be carefully studied to determine if it is a threat trying to exploit a vulnerability or not.

The priority formula has four distinct parameters that have to be given by the security expert: Model Confidence, Relevance, Severity and Asset Criticality.

The Model Confidence variable concerns the level of knowledge available about the target asset (asset under evaluation), measuring the level to which the target asset was already modeled and/or scanned before. All possibilities of the Model Confidence score are described in Table 2.4.

| Level Number | Description |
|---|---|
| 0 | Target is not modeled at all, target asset id is not populated |
| 4 | Target asset id is present, but it hasn't been scanned for open ports or vulnerabilities |
| 8 | Target asset is either scanned for open ports or vulnerabilities, but not for both |
| 10 | Target asset is scanned for both open ports and vulnerabilities |

Table 2.4: Model Confidence score possibilities

The Relevance variable is affected by the fact that a target asset has an exploitable vulnerability, where the event represents an action that might exploit it and the fact that the port, that is being attacked, is opened or not.

The default value starts at the highest score, 10, and depending on the facts mentioned above, it might decrease or increase (in the case that it gets a score over the limit score, will not overcome the highest score). If the action on the event is a port scan the score is decreased by 5, and the same as for a vulnerability scan. If the port is open, the score will be increased by 5, and the same happens if there actually is a vulnerability that can be exploited.

For a better insight on this variable, if the action on the event is a scan, on a port or vulnerability, the importance or relevance for the system is decreased, but if the port is open and there is a vulnerability, it will cause an increase in the score. Figure 2.7 illustrates the equation to obtain the possible score of this variable.

For example, if the action on the event is a scan, the relevance is set down to 5, being classified as "Partially Relevant". However, if in addition the port is open and/or there is a vulnerability, the relevance value is set to 10, being classified as "Highly Relevant". Figure 2.7 illustrates the computation to obtain the possible scores of this variable.

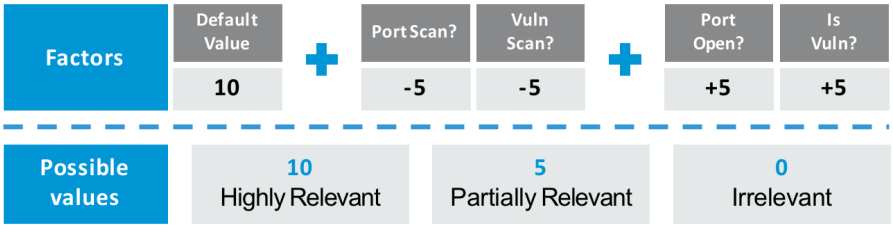



Figure 2.7: Relevance computation and its possible values extracted from [27]

The Relevance (R) and Model Confidence (MC) variables are related to each other in the $Score_{RMC}$, where

$$Score_{RMC} = \frac{R}{(R + MC) - \left(\frac{R*MC}{10}\right)},$$

$R \in \{0, 5, 10\}$, $MC \in \{0, 4, 8, 10\}$

The Severity variable (S) takes into account, not just if the target has already been compromised, but also if prior activity from this source has been detected. The score associated to this variable is:

$$Score_s = 1 + \left(\frac{SeverityLevel * 3}{100}\right)$$

The variable $SeverityLevel$ can be replaced by one of the values shown in Figure 2.8.

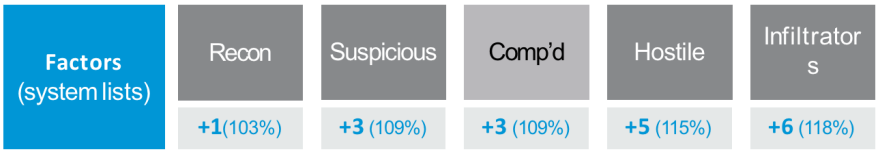



Figure 2.8: Severity Level possible values extracted from [27]

The Recognition value (1) is attained when it can be said, with certainty, that the asset is not indeed compromised. The Suspicious value (3) should be added if there is a possibility of the asset being compromised. The Compromised value (3) is a score that allows to indicate if the asset is, indeed, compromised but the attacker cannot do anything

yet or if it is even possible for the attacker to do anything. It is important to emphasize that the Suspicious and Compromised scores are the same because if the asset might be compromised, it is advisable to assume that it is indeed, to evaluate the severity. But it is necessary to divide the cases to avoid false positives. The other two scores, Hostile value (5) and Infiltrators value (6), correspond to situations where the attacker can jeopardize the system with or without more damage.

The last variable, Asset Criticality, is responsible for measuring the impact of an asset based on its importance in the context of the organization.

There are six levels of importance of an asset on ArcSight, as shown in Table 2.5.

| **Level Number** | **Description** |
|---|---|
| 0 | Unknown |
| 2 | Very Low |
| 4 | Low |
| 6 | Medium |
| 8 | High |
| 10 | Very High |

Table 2.5: Levels of importance of an asset

Taking into account the importance of an asset, it is possible to compute the criticality that the same asset has, as shown in Equation 2.2:

$$Score_{AC} = 1 + \left( \left( \frac{AssetImportance - 8}{10} \right) * 0.2 \right) \tag{2.2}$$

From all formulas and tables, the final equation is shown by Equation 2.3:

$$Priority = (Score_{RMC}) * (Score_s) * (Score_{AC}) \tag{2.3}$$

The final priority score is displayed in the ArcSight console similarly to Figure 2.9.

| Name | Agent Severit | Relevanc | Model Co | Severity | Asset Cri | Priority | Target Po |
|---|---|---|---|---|---|---|---|
| Very-High Asset Criticality | Medium | 5 | 8 | 9 | 10 | 4 | 25 |
| Very-Low Asset Criticality | Medium | 5 | 8 | 9 | 2 | 4 | 25 |
| Recon + Suspicious + Hostile | Medium | 5 | 8 | 9 | 0 | 4 | 25 |
| Recon + Suspicious | Medium | 5 | 8 | 4 | 0 | 3 | 25 |
| Scanned; non-open port | Medium | 5 | 8 | 0 | 0 | 3 | 25 |
| Asset in Database | Medium | 10 | 4 | 0 | 0 | 5 | 25 |

| Name | Agent Severit | Relevanc | Model Co | Severity | Asset Cri | Priority | Target Po |
|---|---|---|---|---|---|---|---|
| Very-High Asset Criticality | Medium | 10 | 8 | 9 | 10 | 8 | 80 |
| Very-Low Asset Criticality | Medium | 10 | 8 | 9 | 2 | 7 | 80 |
| Recon + Suspicious + Hostile | Medium | 10 | 8 | 9 | 0 | 6 | 80 |
| Recon + Suspicious | Medium | 10 | 8 | 4 | 0 | 6 | 80 |
| Recon | Medium | 10 | 8 | 1 | 0 | 5 | 80 |
| Scanned; non-open port | Medium | 10 | 8 | 0 | 0 | 5 | 80 |
| Asset in Database | Medium | 10 | 4 | 0 | 0 | 5 | 80 |

Figure 2.9: Priority scores extracted from [27]

#### 2.2.2.3 Limitations

Unfortunately, ArcSight is considered more complex and extensive to deploy, configure and operate than other leading solutions.

Another problematic situation is that ArcSight is undertaking a development effort to redo the core ArcSight technology platform, making pressure on the costumers to develop plans to ensure the availability of the features and the functions needed to support existing or planned deployments [18].

### 2.2.3 IBM QRadar

The IBM solution for SIEM system was brought to Q1 Labs company and it is called IBM QRadar Secure Intelligent Platform [29], from now on it will be referred as IBM QRadar.

IBM QRadar has deployments for hardware or software based and can have an all-in-one implementation or it can be divided through all the organization as well.

There is no specific architecture for IBM QRadar because it is composed by several modules which can work separately, giving many different possible architectures.

The following sections will demonstrate a general structure of the IBM QRadar SIEM solution, a description of the existing components, the way risk evaluation is performed, as well as some limitations of this solution.

#### 2.2.3.1 QRadar Architecture and Components

Figure 2.10 displays a succinct portrait of a general IBM QRadar architecture with all modules implemented.

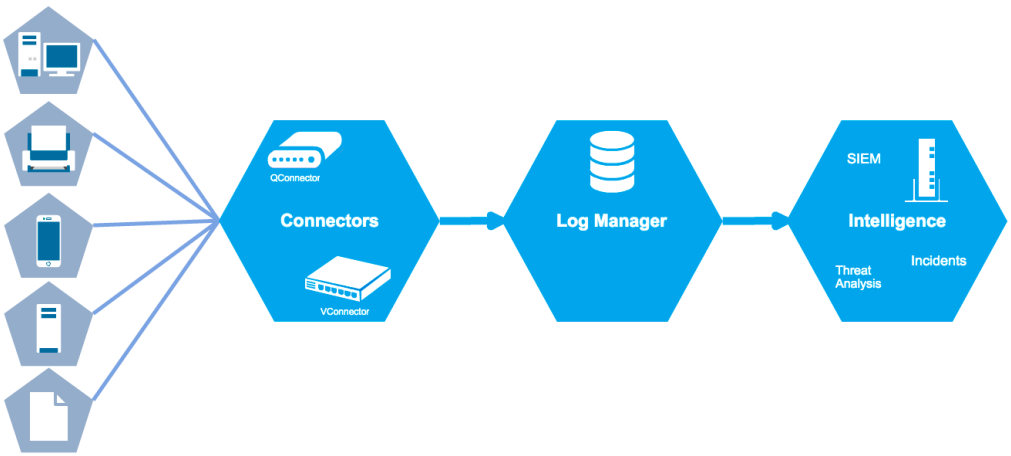



Figure 2.10: General IBM QRadar Architecture

The Connector component provides a greater visibility of the network activities and the assets of the organization, as well as it helps the organization to detect and treat malicious activities.

There are two types of connectors in IBM QRadar, the QConnector [21] and the VConnector [22]. Both have the same fundamental ideas but focusing on distinct manners. The QConnector is at the hardware level and the VConnector focuses more on virtual networks and applications. It is important to mention that VConnector is software based opposed to QConnector which is hardware based. Both, integrated or not, can provide threat detection, policy and regulatory compliance management, social media monitoring, advanced incident analysis and continuous asset profiling.

The Log Manager is the simplest component in IBM QRadar and is the component responsible for collecting the data acquired from the connectors, analyzing and storing it, in a secure way, to guarantee that it is helpful or useful to monitor and report the state of the security of the organization [23].

The Intelligence component is where all the collected data is competently analyzed and treated to proceed to threat analysis and incidents management. To be able to manage all the processes aforementioned, it is required to have the respective modules aforesaid in the beginning of this section.

In terms of SIEM as a whole, it is necessary to have the SIEM itself (that is in the constitution of the Intelligence component in the figure above) and, additionally, the IBM Security zSecure Adapters.

The SIEM allows to consolidate all logs events created and network data from thousands of devices, independently from the type. It can normalize and correlate raw data, to identify any kind of security offenses, and uses an advanced Sense Analytics mechanism to determine normal behavior, detect anomalies, advanced persistent threats and remove false positives as well [14].

The IBM Security zSecure Adapters are responsible for normalizing raw data to be analyzed with more detail for the SIEM, the log manager, the forensics analysis, and so on. The most valuable advantage of it is the extent of protection against Advanced Persistence Threat [30].

To analyze threats there are modules, such as the X-Force Threat Intelligence, the User Behaviour Analytics, the Vulnerability and the Risk Manager.

The X-Force Threat Intelligence is a module that adds Internet threat data, in a dynamic and automated way, to the capabilities of analysis of the IBM QRadar, providing deeper insight and greater protection [25].

The User Behaviour Analytics is an application, but in the same way it can be a module due to the compatibility with IBM QRadar, that allows to monitor internal threats through specific usage patterns in the system [26].

The Vulnerability Manager is a module that focuses on device discovery, detecting and managing vulnerabilities in devices, applications, configurations, and data flow spread to the system [31].

The Risk Manager is a module that has functions, namely, for monitoring equipment

of the network, creating simulations of attacks to assess the impact that those attacks may cause and can, as well, correlate the data that outcomes from the Vulnerability Manager, allowing to prioritize vulnerabilities and create plans to reduce them [24].

Finally, incident management is done in the SIEM as well, but IBM QRadar has a Incident Forensics module which allows to overhaul, step-by-step, actions made by a potential attacker [32]. With the overhaul completed, it is possible to create an investigation to understand what happened indeed and what changes have to be made so an incident of that type never happens again.

#### 2.2.3.2 Risk Evaluation

Risk evaluation on IBM QRadar is done by the Vulnerability Manager module. The Risk evaluation relies on finding vulnerabilities in the network and is associated with the asset that has the vulnerability.

To assess the score of each vulnerability found, IBM QRadar uses the Common Vulnerability Scoring System (CVSS) base score (will be described in further topics). Then, to evaluate the final risk score of each asset, all vulnerability scores are added [33].

It is important to mention that multi-level assets can exist and therefore a multi-level vulnerability score can be created. This means that there are super assets and those super assets are just a set of other assets, as an example, a network. A network can be considered a super asset where the other constituent assets are the lower level assets, such as personal computers, workstations, firewalls, and routers.

Figure 2.11 shows an example of an interface in IBM QRadar Vulnerability Manager module, where a set of super assets and low-level assets with the respective scores are indicated; in order to emphasize that the Vulnerabilities field is the number of all different kind of vulnerabilities, and the field Vulnerabilities Instances is the total number of vulnerabilities without differentiation [34].

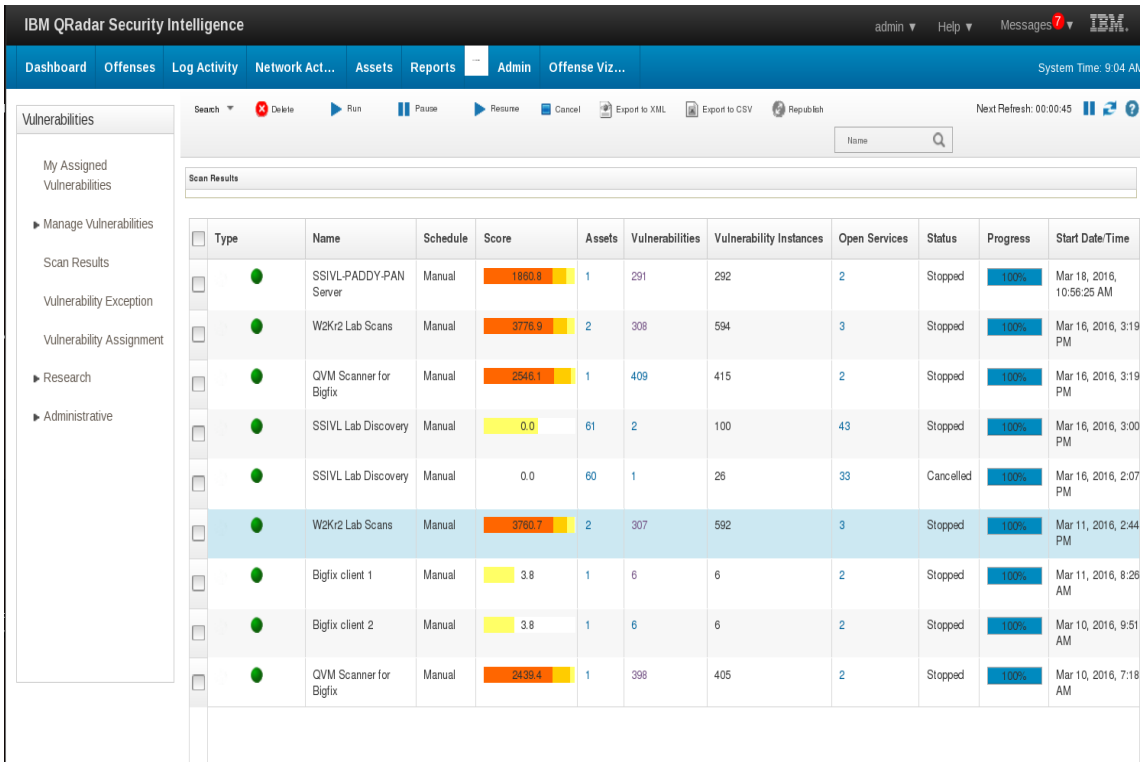



Figure 2.11: IBM QRadar Vulnerabilities Scan results extracted from [35]

#### 2.2.3.3 Limitations

IBM QRadar has a considerable dependency of third-party technologies, especially of Endpoint monitoring for threat detection and response, or basic file integrity. It also has problems to integrate the module Vulnerability Manager with the rest of the system according to [18].

## 2.3 Scientific Literature Review

RiskM is a multi-perspective modeling method for fostering and facilitating the communication and collaboration among stakeholders during the IT risk assessment process [36]. This method is sustained by a modeling language which represents all key concepts, objects and relationships between them, such as Risk, Impact Measure, and Uncertainty, for the method to be comprehensible. The multi-perspective view is divided into three different perspectives: IT Operations, Business Process, and Strategic level. Each perspective represents a different stakeholder and a different level of abstraction of the IT Risk.

The method also has a process model covering the three main phases of risk assessment, the risk identification, risk analysis, and the risk evaluation, indicating how each phase should be proceeded to have the better view, not just of each phase, but also for the IT structure of an entity.

RiskM recommends a two phased process to evaluate risk, which has a bottom-up approach initially and then a top-down to complete the process. We have adapted this concept of process to identify the assets on the organization, instead of evaluating the risk.

A management methodology that addresses risk dependencies and their impact on IT projects during an IT management process is presented in [37]. The authors have concluded that the current methodologies address the risk management in IT too poorly due to considering risk as independent events, leading to an inadequate identification and management of the same risks. In order to solve this problem, a new management methodology was proposed.

This methodology redefines the risk management process and defines the processes to evaluate, react to, monitor, and control the risk dependencies by introducing a novel set of practices and types of dependencies that exist.

A dependency is a relationship between two different risks, which are composed by the Impact and the Probability of happening, and it can have three different types.

After we analyzed the types of relationships that can exist between assets presented in this paper, we created new types of dependencies between assets divided by layers in order to develop the multi-level model proposed in this dissertation.

Also in this paper, it is presented a set of three methods to calculate the effects of the risk dependencies, namely the Conservative method, Optimistic method, and the Weighted method as well as new metrics to monitor and control the risks.

Further in the dissertation, the three types of dependencies and the methods to calculate the risk as well will be explained in more detail.

## 2.4 Common Vulnerability Scoring System (CVSS)

The Common Vulnerability Scoring System (CVSS) [38] is an open framework for communicating the characteristics and severity of software vulnerabilities, held and updated by FIRST Organization [39]. It provides quantification of the principal characteristics to a numeric score, which can be translated into a qualitative representation. The score, whether qualitative or quantitative, represents the severity of the vulnerability to the system.

Currently, CVSS has three versions of the framework, but nowadays the second version is the most used. Eventually, the third version will subside and, due to that, this section will only focus on the structure of this version.

The CVSS Version 3 [40] has three main metric groups: Base metrics, Temporal metrics, and Environment metrics.

The Base metrics group consists on characteristics that are intrinsic to the vulnerability that never changes over time and across environments. This particular group is composed

of two sets of metrics: Exploitability metrics and Impact metrics.

Exploitability metrics represent the ease and the technical means to successfully exploit a vulnerability and the characteristics of the vulnerable component. The set of Exploitability metrics is: Attack Vector (AK), Attack Complexity (AC), Privileges Required (PR), Score and User Interaction (UI).

The Impact metrics reflect the direct consequence of a vulnerability exploited successfully and represent the consequences of the impacted component. The metrics that measure the impact on impacted components are Confidentiality Impact (C), Integrity Impact (I) and Availability Impact (A).

The Temporal metrics group measures factors that may influence the process of exploiting the vulnerability successfully. Those factors can be patch releases, exploit code releases or confirmation of the vulnerability. To be able to measure the largest number of factors of this nature, there is Exploit Code Maturity (E), Remediation Level (RL) and Report Confidence (RC). The Temporal metrics group is the only group which is optional to the final score of the vulnerability.

Finally, the Environment metrics group allows the analyst to change the score of the assets in order to establish the most accurate score possible for the asset. For instance, if there is an asset where it is crucial for the organization to maintain the confidentiality at all costs. The importance of the confidentiality is higher than integrity and availability, with these metrics it is possible to adjust the degree of importance.

The Environment metrics group has two types of metrics: Security Requirement metrics, which is related to the aforementioned example, and the Modified Base metrics, which is the same concept of the Security Requirement metrics but aims the metrics from Base group, instead of the Integrity, the Confidentiality, and the Availability characteristics.

The final score is based on the three main equations (one for each main group), it is measured between 0.0 and 10.0 (as the other three equations) and can be qualitative as well, as shown in Table 2.6.

| **Rating** | **CVSS Score** |
|---|---|
| None | 0.0 |
| Low | 0.1 - 3.9 |
| Medium | 4.0 - 6.9 |
| High | 7.0 - 8.9 |
| Critical | 9.0 - 10.0 |

Table 2.6: Qualitative Scores

Figure 2.12 illustrates the concept of the CVSS V3.

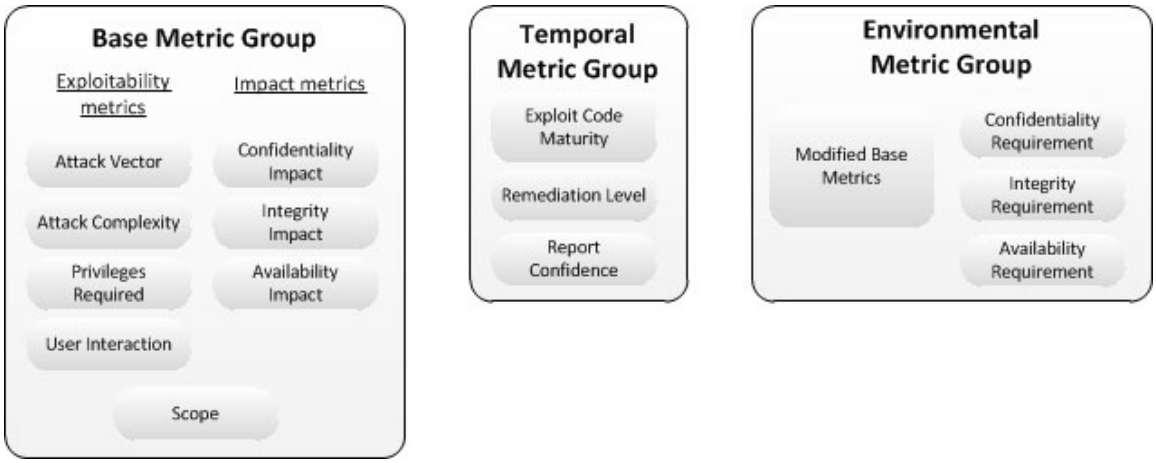



Figure 2.12: Concept of the CVSS V3 extracted from [40]

At last, the Common Vulnerability Scoring System framework allowed to understand what types of conditions that vulnerabilities have, being one of them the Scope. The Scope concept was the major idea taken from this framework and allowed the model to be more realistic. Further, in this dissertation, all these points will be explained in more detail.

**Summary**

In this chapter, we described how the process of risk assessment is done, what standards exist, which organizations are responsible to improve these standards, and how these standards are structure. This is important to set a ground on the methodology to adopt. We have also introduced a structure of a general SIEM and a brief description of the components, the risk evaluation process, and some limitations of the AlienVault, HP ArcSight, and IBM QRadar solutions. Finally, we presented reviews from the scientific literature that supported the work done in this dissertation. We reviewed several SIEM solutions to understand how the process of risk scoring is made. During the review, we realized that the majority of the SIEM solutions assess risk based on specific formulas applied to each event occurred on an asset and we also realized that incidents can be created and classified on SIEMs. Table 2.7 summarizes the existing solutions and their respective inputs.

| Solution | Inputs |
|---|---|
| **IBM QRadar** | Vulnerability ID<br>Vulnerability's Score<br>Asset ID |
| **AlienVault** | Asset Value<br>Priority<br>Reliability |
| **ArcSight** | Model Confidence<br>Relevance<br>Severity<br>Asset Criticality |

Table 2.7: All inputs required to the solutions presented previously

After the analysis of the scientific literature review, we realized that exists a gap between the risk assessment and the multi-level risk assessment that allows having a more precise reality of the risk.

We also reviewed scientific literature and the Common Vulnerability Scoring System (CVSS).

# Chapter 3

# A Multi-Level Model for Risk Assessment in SIEM

This chapter presents the general and non-technical concepts of the model.

We begin by introducing the structure of the model followed by the characteristics of each layer of the model, where we indicate the type of assets each layer contains. Then, we describe the possible types of dependencies that might exist between the elements of the model. Finally, we propose three distinct proposals for assessing risk in the assets.

## 3.1 Structure of the model

The structure of this model is divided hierarchically based on three levels of decision making and it has three main objectives: calculate assets' risk, supply additional information of each asset, and support the decision making process.

In order to calculate the assets' risk, this model divides the assets into three layers: hosts, applications, and services. The approach to assess the risk is a bottom-up approach, meaning that to be able to assess a service, it is required to assess all hosts and applications that are supporting it first.

The layers of the model were designed to be mapped to the different levels of decision making, allowing to enhance by supporting the process in each level. These levels of decision making have to be separated due to the nature and complexity of each every one of them.

The lowest and operational level of decision making is coincident with the hosts' layer, where we are concerned with more technical details about the hosts, their management, and the IT infrastructure itself.

The level of decision making that is coincident with the applications' layer is very similar with the previous one. Although, the abstraction of the IT technical details and infrastructure is more evident, which allows to start to focus on the business side.

However, most of the C-Level managers are more concerned with business and less focused on operational technical issues.

Since there is a necessity of having assets sufficiently abstract and at a strategic level to improve the communication between the IT managers and C-Level managers, this model has a strategic layer of decision making for business functions, which is represented by the services' layer.

The model can facilitate the communication between IT managers and C-level managers, but can also facilitate the process of decision making for each layer since this model provides a risk for each asset. By providing a risk for each asset in each layer, managers can determine which assets have to be treated with more or less urgency, creating a more efficient management process of that layer.

The risk assessment for each asset has three strands: vulnerabilities, dependencies, and incidents. The vulnerabilities strand assesses the security anomalies intrinsic to the asset itself, while the dependencies strand assesses the impact of other related assets to the asset currently under evaluation. Finally, the incidents strand assesses the impact of events with an abnormal pattern.

The assessment of the risk was not based on a probabilistic model due to the difficulty to determine the likelihood of an incident or a vulnerability to be exploited. Instead, we used a model based on scoring the severity of vulnerabilities and incidents to assess the risk scores of each asset.

## 3.2 Characteristics of the Layers

The proposed model for risk assessment aims at assessing risk in three points of view inside an organization, creating a global and detailed vision of the security of the information systems and the respective assets. The model has a hierarchical structure being composed of three layers of assets: Host, Application, and Service, where this last layer has a holistic view of the other ones.

The Host layer, the lowest level layer, consists of the set of all physical assets. These physical assets can be servers or virtualized servers, personal computers, routers, switches, firewalls, and others. At the Application layer, the set of assets includes all kind of software, e.g, middleware, web services, or websites, which supports the organization operation and business, as well as its non-profit services. Lastly, the Service layer represents the abstract assets that characterize a set of actions or functions that are supported by applications and hosts, in order to maintain the objectives of the organization.

## 3.3 Types of Dependencies

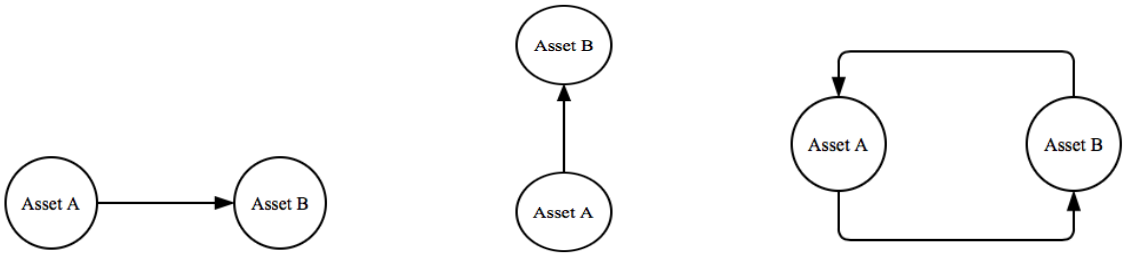
A dependency in this dissertation is a relationship between two assets that can be either unidirectional or bidirectional. Since the model is hierarchically divided in layers, a dependency can be intra or inter layer as well, where an intra layered dependency is on the

same layer and the inter layered dependency is between two assets from different ones. Regarding the possibilities of direction and if its inter or intra layered, we have considered three types of dependencies in this model.

The first type of dependency is an unidirectional and intra layered dependency. The second type is also unidirectional, although it is a inter layered instead of intra. The third and final type is a bidirectional and intra layered dependency. These dependencies were the only ones defined due to the model being designed with a bottom-up approach for the assessment of the risk scores.

Figure 3.1 represents the three types of dependencies considered for this model. It is important to mention that the horizontal connections between assets in the figure indicates an intra layered dependency, and consequently, the vertical connections represent an inter layered dependency.

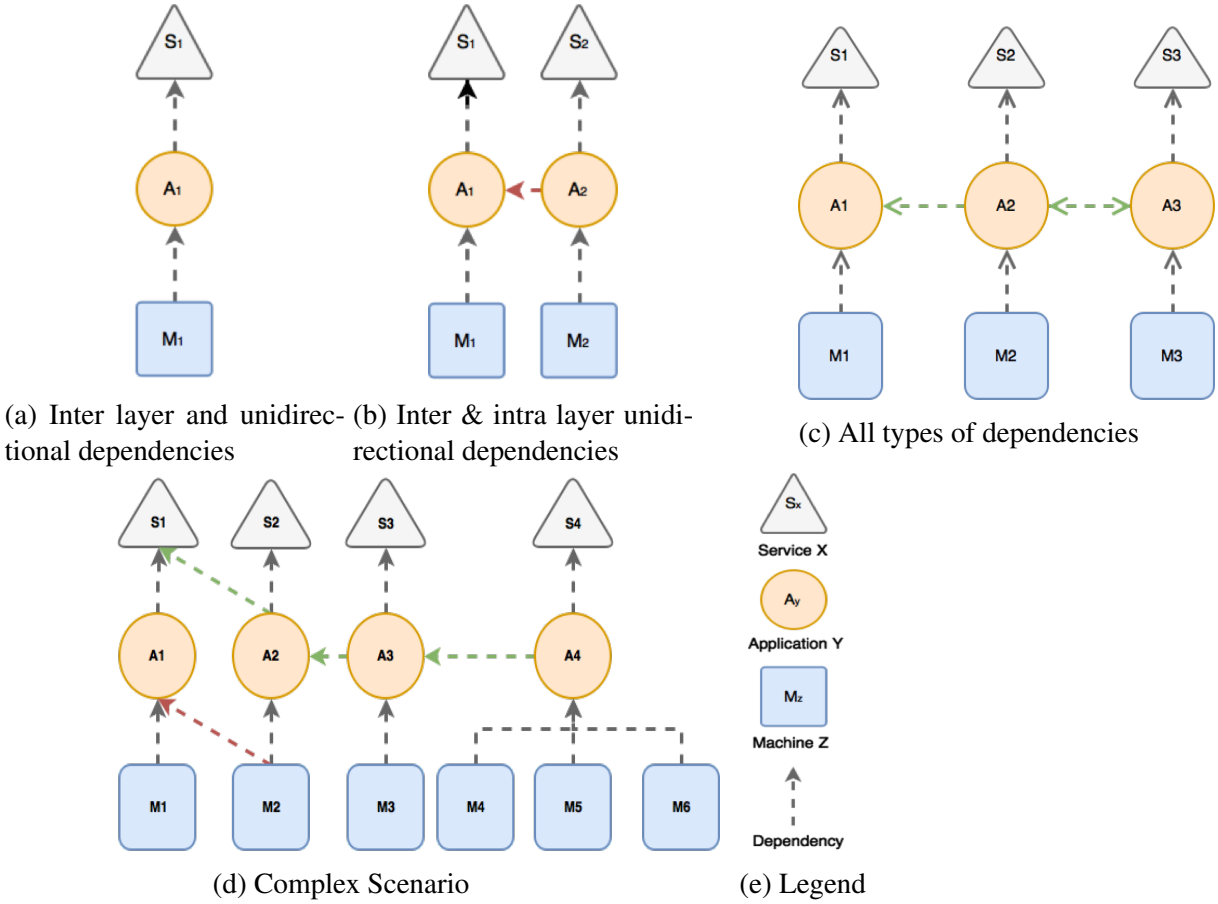
It is important to understand that a dependency can be seen in a different perspective, meaning if asset A depends on asset B, asset B supports asset A. This definition of support is important to understand due to several explanations that will be given further.




(a) Intra-layer dependency (b) Inter-layer dependency (c) Bidirectional dependency

Figure 3.1: Types dependencies between assets

Figure 3.2 shows examples of usage of all types of dependencies created.




(a) Inter layer and unidirectional dependencies

(b) Inter & intra layer unidirectional dependencies

(c) All types of dependencies

(d) Complex Scenario

(e) Legend

Figure 3.2: Example of scenarios with all types of dependencies

## 3.4 Identification of Assets and Dependencies

The process to identify the assets and dependencies between them is divided in two phases: bottom-up, and top-down phase.

The bottom-up phase has the purpose to identify assets supported by applications that have vulnerabilities or assets supported by hosts that have vulnerabilities as well. In order to accomplish its purpose, this phase has three steps.

The first step is to identify hosts that have vulnerabilities based on a list of vulnerabilities. The second step consists in finding all applications that are supported by hosts that have vulnerabilities. Finally, the third step is to identify all services that are either supported by applications that have vulnerabilities or supported by applications that have dependencies on hosts with vulnerabilities. In the end of the third step, all services that are supported by vulnerable assets should have been already identified.

The top-down phase aims at identifying the remaining applications and hosts that support the services that were identified in the previous phase. This phase has three step

as well and the first one is to identify all applications that are supporting each service. The second one is to find all hosts that are supporting each application. Finally, the third step is to display all assets and their dependencies on a dashboard. Further, in this dissertation, the dashboard will be described thoroughly.

This process can be extended to consider incidents along with the vulnerabilities, even it can be extended to identify all assets, with or without vulnerabilities, from the bottom-up phase. To implement the model, this process is not indispensable but applying it guarantees all assets that have vulnerabilities or relationships with other vulnerable assets are identified creating a more realistic notion of the risk of the IT system.

Figure 3.3 describes in detail the phases in six steps.

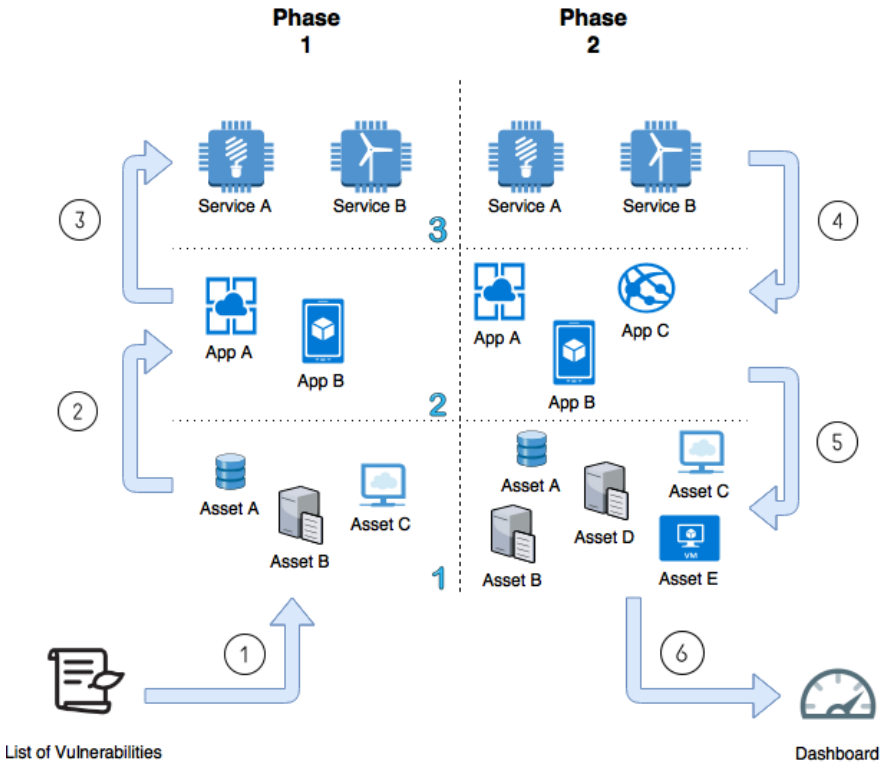



Figure 3.3: Assets and Dependencies Identification Process

## 3.5 Risk Assessment Formula and Proposals

As mentioned previously, for each asset, regardless of the layer, a risk score is assessed. To assess a risk, it is important to know that the risk can be divided into two types: intrinsic, and imported risk. The intrinsic risk is a risk focused on the existing issues on the asset itself, while the imported risk is a risk inherited from other assets due to the dependency on them. In order to weigh both types of risk and to grant a total risk of the asset, the assessment is based on three variables: vulnerabilities, dependencies, and incidents.

The vulnerabilities variable represents the risk of an asset regarding the vulnerabilities presented on it and can be classified as an intrinsic risk variable.

The dependencies variable assesses the risk of other assets that support the asset that is being evaluated, which leads to a classification as an imported risk.

Concluding, the incidents variable assesses the impact of events occurred on the asset that can jeopardize its security and can be classified as an intrinsic risk as well.

For each of these variables, it is needed to compute a risk score.

Equation 3.1 shows how to compute the risk score of a generic asset J.

$$Risk\ Score_J = WeightedSum(VV_J, DV_J, IV_J) \tag{3.1}$$

$Where,$

$VV_J$=*Vulnerability Variable score*

$DV_J$=*Dependency Variable score*

$IV_J$=*Incident Variable score*

It is important to refer several aspects about the previous equation. First, the Risk Score value is comprehended in an intended interval, considering zero as the minimum score value and as maximum a value predefined, e.g.( 10, 100, or 200), defined by the organizations. Second, this risk score uses a linear scale. Third, the function $WeightedSum$ in the equation indicates that each variable has a specific weight attributed where the sum of all variables' weights are equal to 1. In Chapter 4 is described in more detail how the weights from the previous equation are set. Fourth and final, the Services' layer only considers the dependency variable for the risk score, once each service does not have vulnerabilities or incidents. For the Hosts and Applications' layer, all variables are considered.

Equation 3.1 is an abstract formula of how to assess the risk and we developed three different proposals to instantiate it with different factors for each variable, allowing the model to be easier to adapt to different organizations' structures. Those proposals are designated as: Generic Additive (GA), Modified Additive (MA), and Maximum Score version (MS).

**Generic Additive (GA)**

The GA version focuses on adding all scores of vulnerabilities, dependencies, or incidents, depending on which variable is being evaluated, and it is compared with the risk appetite that the organization considers for each variable. The concept of risk appetite represents the amount of risk that an organization is willing to have, or to accept, on an asset, meaning that no further actions will be taken or are not needed on that asset.

As shown in Equation 3.2, as an example for the vulnerabilities variable, the first step is to sum the scores of all vulnerabilities present on the asset. Then, a conversion of scale

is made considering the risk appetite of an asset in order to determine the final score of the vulnerabilities variable.

$$VV_J = \frac{\left(\sum_{i \in Vulns(Asset_J)} VulnScore_i\right) * Scale}{MaxScoreV} \tag{3.2}$$

$Where,$

*$VulnScore_i$= Risk score of the vulnerability i*

*$Vulns(Asset_J)$=Set of Vulnerabilities on $Asset_J$*

*MaxScoreV= Maximum risk score accepted for vulnerabilities on an asset*

*Scale=Upper limit of the scale interval*

The $MaxScoreV$ variable in the previous equation represents the risk appetite of the organization in terms of vulnerabilities on an asset. In order to obtain the score of the $MaxScoreV$ variable is necessary to determine: the highest score value for a vulnerability that the organization is willing to have on an asset, e.g,. a vulnerability with a score of 8 with a range between 0 and 10; a number of vulnerabilities with the highest score value accepted by the organization present on an asset simultaneously; and the asset's business value. Equation 3.3 shows how to obtain the $MaxScoreV$.

$$MaxScoreV = HSV * 2 * NumbVulns * ABV \tag{3.3}$$

$Where,$

*HSV=Highest score Value of a vulnerability accepted by the organization on an asset*

*NumbVulns=Number of vulns with the HSV accepted by the organization on an asset*

*ABV=Asset's Business Value given by the organization*

The vulnerability scoring formula in this model has two elements: vulnerability rate, and vulnerability persistence. The vulnerability rate element has the function to assess the severity of the vulnerability. This model does not aim to evaluate the vulnerabilities themselves, which implies that the scores must be given. As mentioned before, the score of a vulnerability can be qualitative or quantitative, but for this model, a quantitative method to rate the vulnerability is the most appropriate one because we want to assess the risk in the most precise way possible (see Table 2.1).

The vulnerability persistence has the purpose to quantify how long the vulnerability has not been treated in days. As the time passes by and the vulnerability remains on the asset, the attackers have the opportunity to explore the vulnerability in more diversified ways leading to a more probable exploitation. By giving weight to this factor it becomes possible to alarm the top managers that something has to be done to prevent future unwanted situations. The Vulnerability Persistence is between 0 and 1, where a conversion of scale is made between the number of days that the vulnerability has not been treated, the

maximum of days admitted (risk appetite), and the scale with a maximum scale of 1. This element is also considered as an extra emphasis to the vulnerability's score, meaning that the vulnerability's score does not have its impact reduced, only increased or maintained with the vulnerability persistence. Equation 3.4 shows how the vulnerability persistence is calculated.

$$Vulnerability\ Persistence = \frac{NOD}{MAD} \tag{3.4}$$

$Where,$

*NOD = Number of days that the vulnerability has not been treated*

*MAD= Number of days admitted for a vulnerability to be open*

It is important to refer that the $HSV$ is multiplied by 2 due to the vulnerability persistence factor that contributes with a value of 1 for the vulnerability scoring formula when assessing the $HSV$ on Equation 3.3.

Equation 3.5 presents the structure of the final vulnerability score relating the vulnerability score and vulnerability persistence.

$$Vulnerability\ Score = VulnerabilityRate * (1 + VulnerabilityPersistence) \tag{3.5}$$

The dependencies variable $DV_J$ is the sum of all dependencies' scores, which is the total risk of the assets that are supporting the asset J. Then, a conversion of scale is made. The Maximum Score Possible is obtained by multiplying the scale with the number of dependencies that the asset that is under evaluation has. It is important to mention that a dependency in a formula is a representation of an asset that supports the asset that is currently being evaluated, meaning every time the dependency score is mentioned or something similar, it represents the score of the asset that is shown by the dependency in question.

Equation 3.6 expresses the formula for the dependencies variable Score.

$$DV_J = \frac{\left(\sum_{i \in Deps(Asset_J)} DepScore_i\right) * Scale}{NumberDeps} \tag{3.6}$$

$Where,$

*$DepScore_i$=Risk Score of $asset_i$ for which there is a dependency from $asset_J$*

*$Deps(Asset_J)$=Set of assets supporting $Asset_J$*

*NumberDeps= # $Deps(Asset_J)$*

Lastly, the incident score is more complex to calculate. It not only has the scores of the incidents but also the concept of history of the asset.

The history of the asset's incidents is a concept that influences the current score with the score of incidents from the last three months, to emphasize the fact that the asset had problems and those problems were exploited.

As represented by Equation 3.7, the incident score is divided into two parts: the score of the incidents occurred in the current month and the history of the past three months, weighed accordingly to suit the reality of the organization.

$$IV_J = WeightedSum\left(CurrentMonthScore_J, PreviousMonthsScore_J\right) \tag{3.7}$$

The current month risk assessment process is the sum of the scores of the incidents that occurred in the current month in asset J. A conversion of scale is applied to the sum using the intended scale, and the risk appetite for incidents on an asset. This process is similar to the computation of the vulnerabilities variable and is represented by Equation 3.7, specifically on the $CurrentMonthScore_J$ factor. Equation 3.8 describes thoroughly the process of assessing the current month score.

$$CurrentMonthScore_J = \frac{\left(\sum_{i \in Incs(Asset_J)} IncScore_i\right) * Scale}{MaxScoreI} \tag{3.8}$$

$Where,$

*$IncScore_i$=Score of incident i*

*$Incs(Asset_J)$=Set of Incidents on $Asset_J$ in the current month*

*MaxScoreI= Maximum score accepted for incidents on an Asset*

Similarly with the $MaxScoreV$ variable in Equation 3.2, the $MaxScoreI$ represents a risk of appetite for the organization for the incidents, instead of the vulnerabilities. In order to obtain the score of the $MaxScoreI$ variable is essential to determine: the highest score value of an incident that an organization is willing to have; and the number of incidents with the highest score value accepted by the organization. Equation 3.9 shows how to compute the $MaxScoreI$.

$$MaxScoreI = HSV * NumbIncs \tag{3.9}$$

$Where,$

*HSV=Highest score Value of a incident accepted by the organization on an asset*

*NumbIncs=Number of incidents with the HSV accepted by the organization on an asset*

The historical concept assesses the impact of previous incidents, where the contribution of the incidents in the latest months as a higher importance. The score variables used

by Equation 3.10, which has the aim to measure the historical concept of the incident variable formula, is the total risk of the incidents variable on the month specified.

$$PreviousMonthsScore_J = WeightedSum(FMScore, SMScore, TMScore) \tag{3.10}$$

$Where,$

*FMScore=Total Incident Variable Score of one month preceding*

*SMScore=Total Incident Variable Score of two months preceding*

*TMScore=Total Incident Variable Score of three months preceding*

**Modified Additive (MA)**

The MA version has several different factors to be considered on each variable.

The variable $VV_J$, that on the GA version only considers a sum of the scores of the vulnerabilities, in this version gives more relevance to the vulnerability with the highest severity level and the quantity of the vulnerabilities.

On the vulnerabilities variable, the following factors are considered: the Highest Vulnerability's Score, the Sum of All Vulnerabilities But the Highest One (SAVBHO), and the Number of Vulnerabilities.

The highest vulnerability's score factor represents the most severe vulnerability not treated on asset J (Equation 3.11).

$$HV_J = MAX\ (\ VulnScore_i \mid i \in Vulns(Asset_J)\ ) \tag{3.11}$$

$Where,$

*VulnScore$_i$= Risk score of the vulnerability i*

*Vulns(Asset$_J$)= Set of vulnerabilities on asset$_J$.*

The number of vulnerabilities factor is a proportion of the maximum number of vulnerabilities admitted on an asset converted to the defined scale. Equation 3.12 shows the structure of the number of vulnerabilities factor.

$$NoV_J = \frac{(NumbOfVulns_j) * Scale}{MaxVulnsAcc} \tag{3.12}$$

$Where,$

*NumbOfVulns$_J$=Exact number of vulnerabilities present on an asset$_J$*

*MaxVulnsAcc=Maximum number of vulnerabilities accepted on an asset*

*Scale=Upper limit of the scale interval*

It is important to refer that the $MaxVulnsAcc$ variable in the previous equation represents the risk appetite that an organization is willing to have on an asset, which is the maximum number of vulnerabilities regardless of the score of them.

The sum of all vulnerabilities but the highest one (SAVBHO) is the total score of all vulnerabilities excluding the highest score in the asset $J$ and is computed by Equation 3.13. It is important to know that the score of the vulnerability itself must be given to the model and there are no restrictions on the methodology used.

$$SAVBHO_J = \frac{\left(\sum_{i \in Vulns(Asset_J)} VulnScore_i - HV_J\right) * Scale}{MaxScoreV} \tag{3.13}$$

$Where,$

*$HV_J$=Highest Vulnerability in asset$_J$*

*MaxScoreV= Maximum risk score accepted for vulnerabilities on an asset*

*Scale=Upper limit of the scale interval*

The computation of the final vulnerabilities variable for the MA version is given by Equation 3.14.

$$VV_J = WeightedSum(HV_j, SAVBHO_J, NoV_J) \tag{3.14}$$

$Where,$

*$HV_J$=Highest Vulnerability Score on asset J*

*SAVBHO$_J$= Sum of All Vulnerabilities But Highest Score on an asset*

*NoV$_J$=Number of Vulnerabilities on an asset*

The use of SAVBHO allows to differentiate the scores between assets when both have the same highest scored vulnerability, although one of them has other vulnerabilities that can create a considerable impact on the system as well. In terms of the usage of the number of vulnerabilities, this can be a differential factor when two assets have the same highest vulnerability, but one of them has other vulnerabilities, regardless if they can make a considerable impact or not.

The value of the dependencies variable is the sum of the scores of assets supporting asset J weighted by the relative business value of these assets has, considering the total business values of the assets supporting asset J. This allows to give more importance to assets having higher business values in comparison with assets with lower business values.

$$DV_J = \sum_{i \in Deps(Asset_J)} \frac{DependencyValue_i}{TotalSummedDependencyValues} DepScore_i \tag{3.15}$$

$Where,$

*Deps(Asset$_J$)=Set of assets supporting Asset$_J$*

*DepScore$_i$= Risk score of asset i*

*Dependency$_i$Value= Business value of asset i*

*TotalSummedDependencyValues= Total sum of business values of assets supporting Asset$_J$*

In terms of scoring incidents on the MA version, we used the formula of the GA version (see Equation 3.7).

**Maximum Score (MS)**

The MS version is the simplest one presented aiming to assess the risk by considering only the highest scores in terms of vulnerabilities, dependencies, and incidents, as can be seen in the following equations.

Equation 3.16 represents the vulnerabilities variable score, which is the highest vulnerability score present on asset J.

$$VV_J = MAX\ (\ VulnScore_i \mid i \in Vulns(Asset_J)\ ) \tag{3.16}$$

$Where,$

*VulnScore$_i$= Risk score of the vulnerability i*

*Vulns(Asset$_J$)= Set of vulnerabilities on asset$_J$.*

Equation 3.17 structures the formula to obtain the dependencies variable score, which is the highest dependency score on asset J.

$$DV_J = MAX\ (\ DepScore_i \mid i \in Deps(Asset_J)\ ) \tag{3.17}$$

$Where,$

*DepScore$_i$= Risk score of asset i*

*Deps(Asset$_J$)=Set of assets supporting Asset$_J$*

Lastly, Equation 3.18 allows to obtain the incidents variable score, which is the highest incident score on the asset J.

$$IV_J = MAX\ (\ IncScore_i \mid i \in Incs(Asset_J)\ ) \tag{3.18}$$

$Where,$

*IncScore$_i$= Risk score of incident i*

*Incs(Asset$_J$)= Set of incidents occurred on asset J*

**Summary**

This chapter presented the model developed for risk assessment dividing the assets into several layers. The model has a structure based on three layers, the hosts, applications, and services layers respectively. Also in this chapter, three versions of the model were proposed, the Generic Additive, Modified Additive, and Maximum Score.

The main reason for us to propose three different versions is to investigate aspects such as complexity of the version, and the results of these versions regarding their complexity. The Maximum Score version is the most simple version where each variable only considers the highest value among the sets of vulnerabilities, dependencies, and incidents as well. Nevertheless, the Generic Additive has a structure where all scores of vulnerabilities, dependencies, and incidents contribute for the score. When comparing these two versions, it is possible to understand if the extra complexity of the Generic Additive version has a higher impact on the risk score or not, being possible to conclude if the same extra complexity is worth it to have. The ideology of the Modified Additive is to understand the impact that might exist when giving different levels of importance for different factors, such as the additional number of factors in the vulnerability variable, and the differentiation of the importance of related assets when computing the dependency variable. Table 3.1 shows the factors that are considered in each variable in all versions.

| | **Vulnerability variable** | **Dependency Variable** | **Incident Variable** |
|---|---|---|---|
| **GA** | All scores summed<br>Maximum score Accepted | All scores summed<br>Number of deps | All scores summed<br>Maximum score Accepted<br>Current month<br>Previous months |
| **MA** | Highest Scored Vuln<br>Number of Vulns<br>SAVBHO | All scores summed<br>Asset's business value<br>All assets' business value summed | Same as **GA** |
| **MS** | Highest Scored Vuln | Highest Scored Dep | Highest Scored Inc<br>Current month<br>Previous months |

Table 3.1: All factors for each version in each variable

# Chapter 4

# Model Implementation

This chapter describes the implementation of a tool for risk assessment integrated into a real industrial organization, in specific, in the EDP's SOC system.

The chapter starts by giving a global vision of the tool architecture, describing how the sources of information, the database, the process of risk assessment and the dashboards are interconnected.

A description of the possible scenarios that may exist based on the definition of dependencies and their types is given.

The process of risk assessment is presented and focusing the technical details associated with.

The chapter ends with a description of the dashboard created to allow visualizing the risk of each asset as well as the process of navigating between the assets hierarchically and the configuration of the model's parameters.

## 4.1 Tool Architecture & Data Flow

Figure 4.1 displays a representation of the tool architecture, the data flow and what interactions are made between all tool's components. The Dashboard is fed by the SIEM archives and the database created specifically for the risk assessment.

The structure of the SIEM was already mentioned previously, where the sources, such as firewalls, IPSs, and hosts, send logs with data to the SIEM's connectors and consequently, these connectors send them to the Loggers and to the SIEM itself.

Based on events that are caught on the SIEM as a potential threat, the EDP SOC's team analyzes them and categorizes them as an incident or not. In case of being categorized as an incident, they are separated to be put in a database strategically designed for the risk assessment process.

The database for the risk assessment process has three different sources of information: incidents created by the SIEM, EDP's internal applications, and a list of vulnerabilities and the assets where they were found. With all these sources it is possible to gather all

information needed. Thereinafter, an application collects the data required and proceeds with the risk assessment process. Once the application has finished the process, it stores back to the database with updated information about the corresponding assets risk scores.

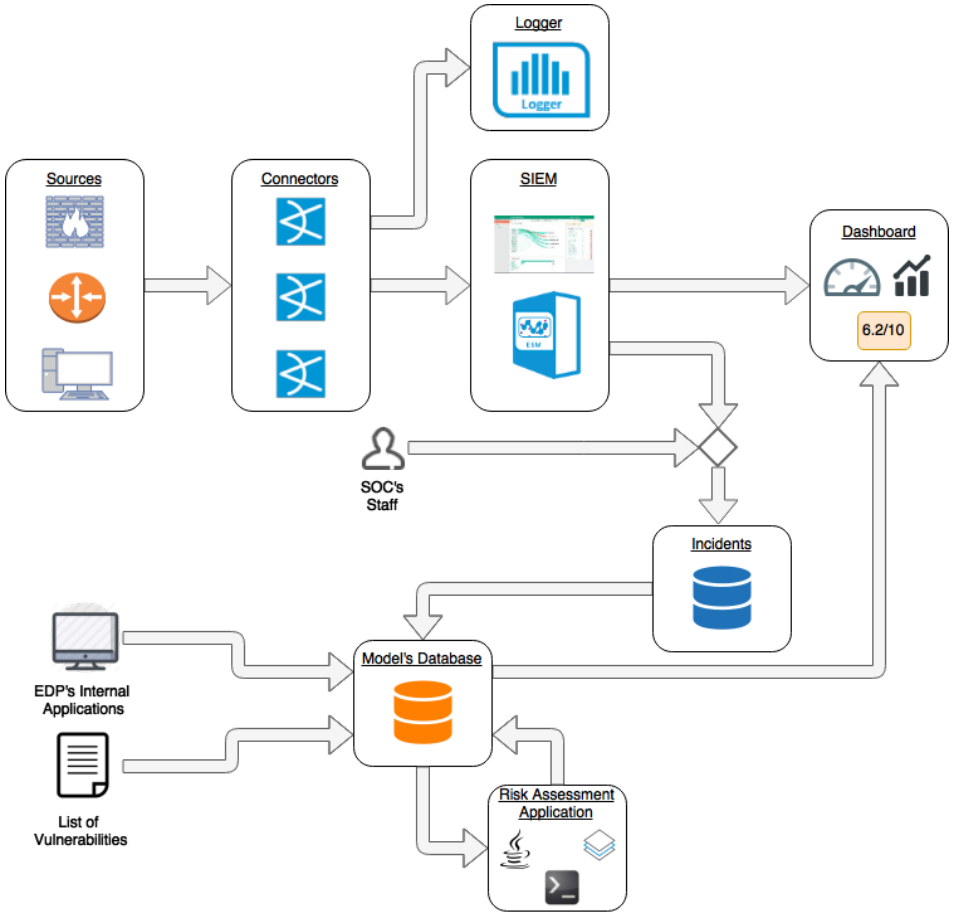



Figure 4.1: Tool architecture and Data Flow between components

The dashboard shows the assets hierarchically divided between hosts, applications, and services with the respective risk scores and dependencies, and also shows the settings of the model's parameters and a global risk of the services.

## 4.2 Model Database

To implement the model, it is necessary to dispose, keep and make available all the information that is needed to support it and the best solution that we came across was to use a modeling language.

To represent the database model, we decided to use the Unified Modelling Language (UML) [41], because it is general purpose, allows to provide a standard visualization of the designed model, and is object oriented.

Figure 4.2 describes the data model. As the main table, the $Asset$ Table represents

the organization's assets and the attributes to assess the risk of each asset. An asset must have a unique Identification ($ID$), $Value$, and $Typeof Asset$.

The $ID$ is used to differentiate all assets, preventing possible mismatches during the assessment. The $Value$ attribute represents the business value of the asset to the organization.

The possible values should be defined coherently with the values that the organization has to classify the assets, with the purpose to normalize the model with the rest of the organization's environment. Once the development of the model was made at EDP's facilities and their environment, the qualitative values chosen for the implementation were: $Bronze$, $Silver$, $Gold$, and $Diamond$, where $Diamond$ is the highest value.

Finally, the $Typeof Asset$ attribute is used to separate the assets into the layers of the model, being the only options: Host, Application, and Service.

The attributes mentioned above are the crucial attributes to assess the risk. However, it is desirable to add other attributes according to the organization's needs, such as, $Name$, Internet Protocol Address ($IPAddress$), Media Access Control Address ($MACAddress$), $AssetOwner$, or a $Description$ of the asset to help the security analysts with their functions or simply to display more information about an asset on a dashboard.

To illustrate the dependencies between assets, the $Dependency$ Table allows to save both of the $IDs$ in attributes $Successor$ and $Predecessor$. The $Successor$ attribute holds the $ID$ of the asset that depends on the other asset, which consequently, the $Predecessor$ attribute holds the $ID$ of that other asset. In other words, the $Successor$ attribute will have the $ID$ of Asset A and the $Predecessor$ attribute will have the $ID$ of Asset B, if Asset A depends on Asset B.

Since the main focus is to assess the risk of the assets, a table to represent the risk itself is needed. The $Risk$ Table has several scores as attributes due to the multiple variables in the equations used. The $TotalScore$ attribute aims to indicate the total risk based on Incident, Dependency and Vulnerability Scores, which are also indicated as attributes on this table. Furthermore, the $Risk$ Table also has a $Date$ attribute to store the date when the risk was calculated and to help to track the evolution of the asset's risk, an $ID$ to be easily identified, and the $ID$ of the asset with which is associated.

To track the evolution of an asset's risk, the $AssetRiskScoreHistoric$ Table is used. This table has four attributes: $IDAssetRiskScore$, $AssetID$, $RiskScore$, and $Date$.

Both $IDs$ are to identify uniquely the score of an asset and the asset respectively, and intuitively, the $RiskScore$ and $Date$ attributes to keep the score and the date when the score was calculated.

The main difference between these two tables is the fact that the $AssetRiskScore-Historic$ Table is only used once a month, storing the last risk score calculated for a particular month and holding the risk's evolution of an asset from the past twelve months.

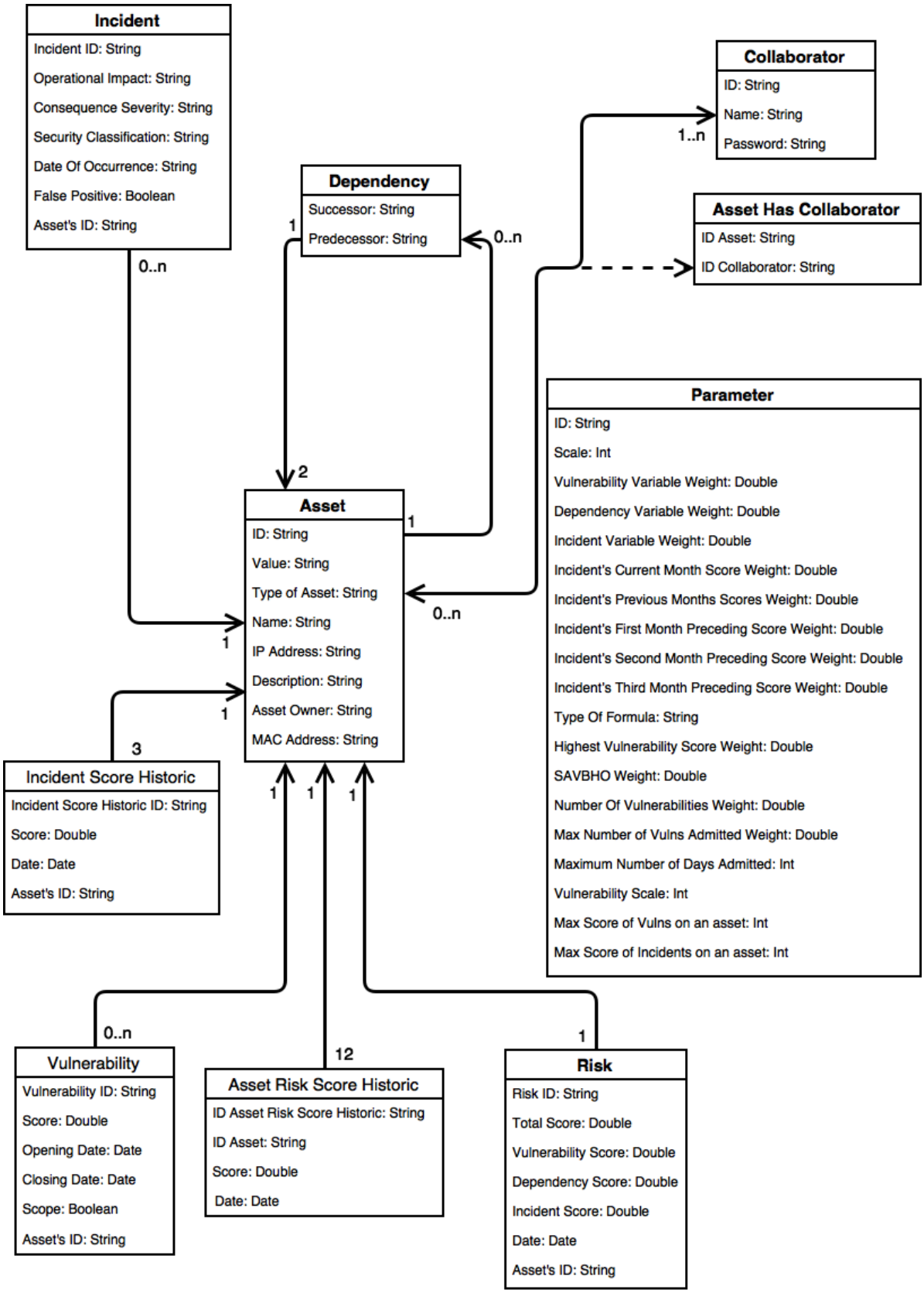



Figure 4.2: Structure of the model and database

To provide the scores of each variable to the $Risk$ Table, it is required to portray the incidents and the vulnerabilities on the model as well, although the dependencies are not needed. As mentioned previously, the score of the dependencies is related to the score of the asset that those dependencies represent, meaning when a dependency's score is needed, we should obtain the $TotalScore$ attribute of the $Risk$ Table associated with those assets.

The $Incident$ Table has the aim to indicate all mandatory attributes of an incident to be assessed. As EDP uses the ArcSight HP Technology as SIEM solution, is used the $OperationalImpact$, $ConsequenceSeverity$ and $SecurityClassification$ attributes.

## 4.3 Implementation

When creating an incident detected by the ArcSight, it is required to fulfill certain properties, including the operational impact, consequence severity and the security classification of the incident. These properties quantify the impact that an incident can cause, how severe the consequences are for the organization after the incident and what type of classified material is involved respectively. The values for these properties are provided by the ArcSight itself and are immutable. Table 4.1 shows the values for each propriety.

As important as the incident's attributes described previously, there is the $DateOf-Occurrence$, the $FalsePositive$, the $Incident'sID$ and $Asset'sID$ attribute.
The $DateofOccurrence$ attribute is essential to determine when the incident happened and which month will have its score affected by it. The $FalsePositive$ attribute consists on an indication if the incident really occurred or not.

Occasionally, there are incidents that are triggered by the SIEM and its rules, which are indeed not an incident. In a way to improve those rules and the SIEM itself, instead of just deleting the incident, it is considered as false positive to be an evidence of the misbehavior of those rules or SIEM and will not be considered to the risk's assessment.

| Propriety | Possible Values |
|---|---|
| Operational Impact | 0 - No Impact<br>1- No Immediate Impact<br>2- Low Priority Impact<br>3- High Priority Impact<br>4- Immediate Impact |
| Consequence Severity | 0 - None<br>1- Insignificant<br>2- Marginal<br>3- Critical<br>4- Catastrophic |
| Security Classification | 1 - None<br>2- Insignificant<br>3- Marginal<br>4 - Critical |

Table 4.1: Values provided by ArcSight

The incidents variable formulas discussed in the previous chapter have a historic factor. In order to keep the track of what scores were calculated for each of the three preceding months, the $IncidentScoreHistoric$ Table was considered.

This table is a basic table which has it's $ID$, $asset'sID$, the $RiskScore$ of the months and the $Date$ attribute to determine the actual month is on the calendar. For each asset, three instances of this table will be created to portray each month and to save the respective total score.

Once we have used the ArcSight technology to implement the model, the score of an incident is done by multiplying the Operational Impact, Consequence Severity, and Security Classification, as can be seen in Equation 4.1.

$$Incident\ Score = OI\ * CS\ *\ SC \tag{4.1}$$

$Where,$

*OI=Operational Impact*

*CS=Consequence Severity*

*SC=Security Classification*

It should be mentioned that the process of scoring the incident itself changes from SIEM solution to SIEM solution, meaning that it is required to correctly configure the $Incident$ table along with its attributes to truly adapt to all SIEM solutions every time that it is implemented.

All vulnerabilities are represented by the $Vulnerability$ Table. This table has the

following attributes: $ID$, $Score$, $Type$, $Asset'sID$, $OpeningDate$, $ClosingDate$, and $Scope$.

The $ID$ and $Asset'sID$ are the attributes that allow to uniquely identify each vulnerability and the asset which has it respectively. The Score attribute serves to store the vulnerability rating. EDP has a service which allows to investigate the assets in order to find vulnerabilities. Once a vulnerability is found, it can be classified as: $Info$, $Low$, $Medium$, $High$, and $Critical$. After the qualitative assessment of the vulnerability, it is converted to a quantitative score based on the CVSS method. Table 4.2 shows for each qualitative value, the respective quantitative value.

| **Qualitative Value** | **Quantitative Value** |
|---|---|
| Info | 0 |
| Low | 4 |
| Medium | 6,5 |
| High | 8 |
| Critical | 9 |

Table 4.2: Vulnerability's qualitative scores and the respective quantitative ones

The $OpeningDate$ and $ClosingDate$ are attributes to keep the date when the vulnerability was found and the date when the same vulnerability was treated respectively. Once the $ClosingDate$ attribute contains a date, the vulnerability is not used for the asset's risk assessment, otherwise the vulnerability is taken into account.

Lastly, the $Scope$ attribute is a Boolean value which shows if an exploitation of a vulnerability does not grant access to other assets related to the asset that has the vulnerability. The problem of having a vulnerability capable of spreading to other assets is too grievous not to be considered because it can jeopardize the organization even more. In case of a vulnerability affects related asset's, its Scope is considered $Changed$, otherwise, the Scope remains $Unchanged$. When assessing the risk for each asset, it is considered the vulnerabilities of other connected assets that have vulnerabilities with the Scope $Changed$ by comparing the highest scored vulnerability of the related assets with the Total Score of the asset that is being evaluated. In case the Total Score of the asset under evaluation is lower than the highest vulnerability with Scope $Changed$, the $TotalScore$ of the asset will be automatically changed to the score of that vulnerability.

Every asset should have its owner or responsible identified to be possible to communicate with them in the case of some unwanted situation. In order to fulfill that idea, it was created the $Collaborator$ Table containing the $Collaborator'sID$, $Name$, and $Password$. Once a collaborator can have multiple assets and an asset can have multiple owners or managers, the $AssetHasCollaborator$ Table is used to identify the asset and collaborator's IDs to identify the connection between them.

Finally, it is necessary to have a table to store all values for all weights and parame-

ters the model requires. Table 4.3 and Table 4.4 display the $Parameter$ Table having a detailed description for each attribute, which corresponds to a parameter to be taken into consideration on the model.

| **Attribute** | **Description** |
|---|---|
| Scale | The intended scale for the total score of an asset as for the Vulnerability, Incident, and Dependency variables |
| Weight Vulnerabilities Variable | The pretended weight of the Vulnerability Variable to assess the total risk, ranged between 0 and 1 (See Equation 3.1) |
| Weight Dependencies Variable | The pretended weight of the Dependency Variable to assess the total risk, ranged between 0 and 1 (See Equation 3.1) |
| Weight Incidents Variable | The pretended weight of the Incident Variable to assess the total risk, ranged between 0 and 1 (See Equation 3.1) |
| Weight Incident's Current Month Score | Weight of the Incident's Current Month factor to assess the Incident Variable score (See Equation 3.7) |
| Weight Incident's Previous Months Scores | Weight of the Incident's Previous Months factor to assess the Incident Variable score (See Equation 3.7) |
| Weight Incident's First Month Preceding Score | Weight of the Incident's first preceding month factor on PrevioustMonthScores (See Equation 3.10) |
| Weight Incident's Second Month Preceding Score | Weight of the Incident's second preceding month factor on PrevioustMonthScores (see Equation 3.10) |
| Weight Incident's Third Month Preceding Score | Weight of the Incident's third preceding month factor on PrevioustMonthScores (See Equation 3.10) |
| Type Of Formula | Type of formula being used, Generic Additive, Modified Additive, or Maximum Score, for the application to proceed correctly the risk assessment |
| Weight Highest Vulnerability Score | This attribute holds the weight of the highest vulnerability score in the Vulnerability Variable when using the Modified Additive formula (See Equation 3.11 and Equation 3.14) |

Table 4.3: Weight Table attributes and their descriptions Part I/II

| Attribute | Description |
|---|---|
| Weight SAVBHO | Weight of the Sum All Vulnerabilities But the Highest One score in the Vulnerability Variable when using the Modified Additive formula (See Equation 3.13 and Equation 3.14) |
| Weight Number Of Vulnerabilities | Weight of the number of vulnerabilities score in the Vulnerability Variable when using the Modified Additive formula (See Equation 3.12 and Equation 3.14) |
| Maximum Number of Vulnerabilities Admitted | Maximum number of vulnerabilities on an asset stipulated by the organization (See Equation 3.12) |
| Maximum Number of Days Admitted | Maximum time, in days, that an asset's vulnerability has not been treated stipulated by the organization (See Equation 3.4) |
| Vulnerability Scale | Maximum limit of the method to assess the vulnerability rating (See Equation 3.5) |
| Maximum Total Score Accepted of Vulnerabilities on an asset | Maximum value admitted for the vulnerabilities' score in a single asset (See Equation 3.2) |
| Maximum Total Score Accepted of Incidents on an asset | Maximum value admitted for the incidents' score in a single asset (See Equation 3.8) |

Table 4.4: Weight Table attributes and their descriptions Part II/II

## 4.4 Handling Dependencies

As mentioned in the previous chapter, there are several types of dependencies, related both to the layers or the assets themselves. Based on them, it is possible to identify eight distinct interconnections as described in the following list:

- Interconnection Type 1 - A Service depending on an Application
- Interconnection Type 2 - An Application depending on a Host
- Interconnection Type 3 - A Host depending on another Host
- Interconnection Type 4 - An Application depending on another Application
- Interconnection Type 5 - Two Applications depending on each other simultaneously
- Interconnection Type 6 - Two Hosts depending on each other simultaneously

- Interconnection Type 7 - One Host supporting more than one Application simultaneously
- Interconnection Type 8 - One Application supporting more than one Service simultaneously

Interconnection Type 1 to Interconnection Type 6 are the standard and intuitive interconnections for this model, and they have already been mentioned previously when we described the possible dependencies, leaving only Interconnection Type 7 and Interconnection Type 8 to be explained.

Interconnection Type 7 aims to show the possible situations where two or more Applications have a dependency with the same Host. If any of those Applications do not have any vulnerability with its scope unchanged, the Total Score of the Host remains unaffected. Nonetheless, if there is a vulnerability, or vulnerabilities, with its Scope changed on those applications, the Total Score of the Host might change. The Total Score of the host changes in case of the Total Score of the host being lower than the score of the vulnerability presented on the application. We developed the model in this way to emphasize the danger of having vulnerabilities with the Scope $Changed$ because this type of vulnerability can cause problems on related assets, which increases the global risk of the system. This can also happen when the host only supports one Application, however the impact on the system is observed more clearly when the hosts have two or more Applications to support.

It is important to change the Total Score of a Host that supports several Applications, in case any of these Applications have vulnerabilities with their Scope changed, because that Host is the common element, the single point of failure or infection propagation between those Applications. The change itself is only applied when the Total Score of the Host is lower than the highest scored vulnerability with its Scope changed.

Interconnection Type 8 is quite similar to the previous one, where instead of Applications and Hosts are Services and Applications. Once the Services do not have vulnerabilities, due to its type of asset, there is no score changes for the Applications whatsoever, meaning the real emphasis on this type of interconnection is only the fact that an Application can support more than one Service at the same time.

The last two types of interconnections are important to be referred because they are a common practice among organizations due to many reasons, some of them being the unnecessary extra servers to support new Applications, and budget limitations.

It might be difficult to find all interconnections types in a single Information System simultaneously, especially in a small organization, however, the model has to be prepared for the majority of possible scenarios that might exist.

## 4.5 Risk Assessment Application

The application that assesses the risk was implemented in Java. The application can be divided into three phases: Gathering Information, Assessing Risk, and Updating Database.

The Gathering Information phase consists on accessing the database to obtain the data needed. For each instance of each table, an object with the same attributes as the tables is created to represent the instance. For example, if there are forty assets in the database, forty Java objects will be created. In case of any of the forty assets have dependencies, vulnerabilities or incidents, the relationship between them will be created as well.

After all objects and their relationships have been created, the history of incidents and risk scores is updated. However, this step only occurs on the first day of each month.

Also during this phase, the Host assets need to have its Value (Business value) confirmed by updating this one with the value of the most valuable supported application. This means, if a Host supports two applications valued as Bronze and Diamond respectively, the Host has its value altered to Diamond automatically. This step is needed due to EDP's business strategy of evaluating only the applications in terms of criticality, which simplifies the business process once the number of assets to be assessed is smaller.

As a consequence of all previous steps, the Assessing Risk phase is initiated. This phase has as its purpose to assess the risk for each asset that is represented in the application.

The first step is to identify which formula will be applied by checking the attribute of the object created to represent the weights ($Parameter$ Table from the database).

Then, it is necessary to order correctly the assets taking into account the dependencies between them. This means the dependencies of a particular asset will appear first to be asses when compared with the particular asset.

After the assessment of all assets is completed, the Updating Database phase is initiated, which consists in updating the database with all new data.

## 4.6 Dashboard

The risk assessment application itself does not have an interface to show the assets and their respective risk scores once the security expert gives instructions to import all data needed directly into the database.

In order to integrate with EDP's organization of work and dashboards, we created a dashboard adapted to EDP's main dashboard to display all details that are crucial for the risk assessment, and also other details of each asset that are stored in the database, such as, Name, IP Address, and so on, to have a better comprehension of assets when the security specialist is analyzing them.

The dashboard is composed by six PHP pages: Global Risk, Services, Applications, Hosts, Parameters Configurations, and Login page.

The Global Risk page has two main panels: a graphic of the organization's global risk, and the metrics. The metrics panel displays metrics that represent the three highest scored services, applications, and hosts, granting a better perception of the most critical assets to be treated for each layer.

The graphic of the global risk represents the evolution of the global risk in the last twelve months. The global risk is an additional service asset that relates all available services of an organization by depending on them, creating a global view of the state of all services. This graphic is composed of the risk scores of the global risk in the last twelve months, scaled accordingly to the intended scale defined in the Parameters Table of the model.

Figure 4.3 exemplifies the structure of the Global Risk page.

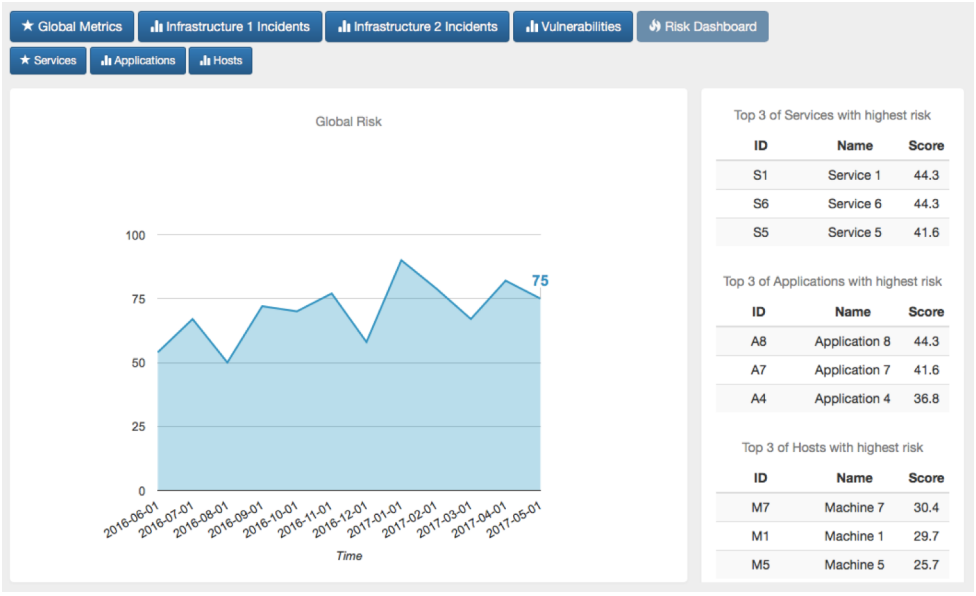



Figure 4.3: Global Risk Dashboard Page

The Services, Applications, and Hosts pages have the same structure, which is a table with the most appropriated attributes for each level, as can be seen in Figure 4.4, Figure 4.5, Figure 4.6, respectively. Each line refers to a different asset and their dependencies. The dependencies can only be seen when the button of the far left of each line is pressed as shown in Figure 4.4 and Figure 4.5. For every page, there are four buttons in the upper right corner of the page, designated as 'Risk Dashboard', 'Services', 'Applications', and 'Hosts' that allow the user to navigate through the different layers of the model and to access the global risk page as well.


Figure 4.4: Services Dashboard Page


Figure 4.5: Applications Dashboard Page

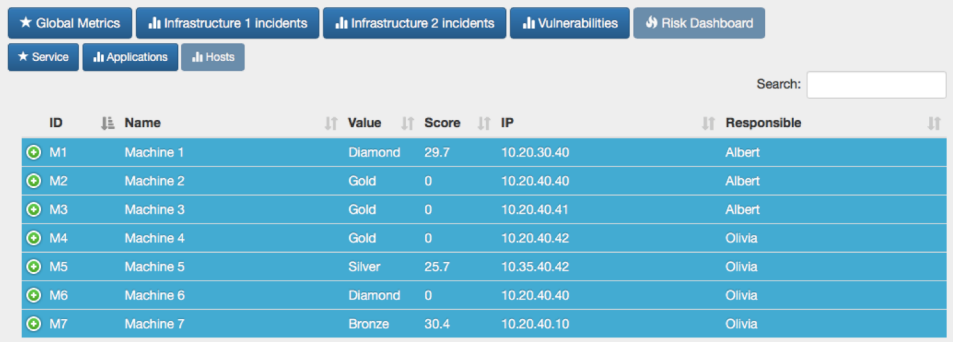



Figure 4.6: Hosts Dashboard Page

The Parameters Configurations page has the aim to configure all the weights and parameters from the database's $Parameter$ Table. This page divides the weights into General, Vulnerability, and Incident, as can be seen in Appendix .1. However, only authenticated users can access this page.

The Login page allows to authenticate the users to access the dashboard based on a form, illustrated in Figure 4 in Appendix .1, and the Collaborator's ID and Password stored in the $Collaborator$ Table in the database.

**Summary**

This chapter began by presenting the architecture and the data flow of the tool developed in EDP's facilities. Then, a detailed description of the data model was presented, followed by how the dependencies between assets were handled. Finally, a description of the application that assesses the risk and a description of the dashboard were presented as well.

After the tool is implemented, the security expert has to fill all parameters of the assets, vulnerabilities, previous incidents, collaborators, and so on. After the phase of filling the tool's database is completed, the effort reduces drastically once only information about vulnerabilities and incidents has to be updated. The main benefit of having a risk assessment process with different levels of decision making comparing with the simple risk evaluating per event on an asset that the SIEMs have.

# Chapter 5

# Results and Discussion

This chapter presents the results of the risk assessment process using the model presented in Chapter 3 and the implementation described in Chapter 4.

Firstly, the experiment is described, where it is defined the parameters of each component of each version of the model, what is a scenario, and how scenarios were generated. The results obtained for each scenario are presented by each level of the model based on charts for a better visualization of the differences.

Finally, a set of comparisons between scenarios is presented to assess the impact of having more or less dependencies in all types of assets and having more or less vulnerabilities on applications and hosts as well. These comparisons are supported by the analysis of the charts followed by a discussion of how the versions behaved.

## 5.1 Description of the experiment

In order to achieve the results, first, we established scenarios and configured the parameters of each variable and factor needed for the process.

A scenario is a structured representation of the reality considering assets, divided by their type, the relationships between them, and the corresponding vulnerabilities and incidents.

Four scenarios were generated in order to test the model and evaluate how the risk score varies specifically when there is a low or a high number of dependencies between assets, as well as a low or a high number of vulnerabilities in hosts and applications.

Each scenario has five services and each service has ten applications. To support all applications and services, there are fifty hosts. In order to design the scenarios, we used a random generator [42] to determine several characteristics, such as:

- How many Hosts does an Application depend on
- Which Hosts support each Application
- How many Hosts does a Host depend on

- Which Hosts support a Host
- How many Applications does an Application depend on
- Which Applications support each Application
- How many vulnerabilities and incidents an Application or a Host has
- The severity of the vulnerabilities, and the incidents scores as well

For all scenarios, the number of dependencies between applications is comprehended between zero and three, and each application can have at maximum five hosts supporting it, and a minimum of one host. The differentiation between scenarios is in the number of vulnerabilities and in the number of dependencies between hosts.

In terms of the number of vulnerabilities in applications and hosts, two groups were considered: low number of vulnerabilities, where the maximum number of vulnerabilities in an asset is two ($-V$), and the high number of vulnerabilities group ($+V$), which limits the number of vulnerabilities to six instead of only two.

The dependencies between hosts differentiation factor is also divided in two groups: no dependencies at all ($MDep-$), and a number of dependencies between hosts comprehended between zero and three ($MDep+$).

With these differentiation factors, we aimt at understanding what happens with the scores of each asset for each model and the respective impact on the system, and how risk varies when the number of dependencies increases or decreases, as well as for the number of vulnerabilities in the system.

Table 5.1 describes the four scenarios based on the two factors mentioned previously.

| | $MDep-$ | $MDep+$ |
|---|---|---|
| $-V$ | Scenario 1 (Sce1) | Scenario 2 (Sce2) |
| $+V$ | Scenario 3 (Sce3) | Scenario 4 (Sce4) |

Table 5.1: All scenarios tested and the respective factors

For all scenarios, each version of the model was tested and compared with the others to investigate the differences between them.

Table 5.2 indicates the value chosen for each weight of parameter of the model. The selection of these values was made after tests, where we compared several abstract assets with different number of vulnerabilities as well as with different scores, until we obtained results that we have considered adequate for each asset. However, these parameters are easily reconfigured for each organization to adapt the model to its environment.

| Attribute | Value |
|---|---|
| Scale | 100 (0-100) |
| Weight Vulnerabilities Variable | 0.7 |
| Weight Dependencies Variable | 0.15 |
| Weight Incidents Variable | 0.15 |
| Weight Incident's Current Month Score | 0.8 |
| Weight Incident's Previous Months Scores | 0.2 |
| Weight Incident's First Month Preceding Score | 1/2 |
| Weight Incident's Second Month Preceding Score | 1/3 |
| Weight Incident's Third Month Preceding Score | 1/6 |
| Weight Highest Vulnerability Score | 0.75 |
| Weight SAVBHO | 0.2 |
| Weight Number Of Vulnerabilities | 0.05 |
| Maximum Number of Vulnerabilities Admitted | 6 |
| Maximum Number of Days Admitted | 365 |
| Vulnerability Severity Scale | 10 (0-10) |
| Maximum Score Accepted for Vulnerabilities on an asset | 216 |
| Maximum Score Accepted for Incidents on an asset | 64 |

Table 5.2: Parameters

The justification for the values of the maximum total score of incidents and vulnerabilities relies on the risk appetite previously defined.

The maximum total score of vulnerabilities in an asset was obtained by applying the Vulnerability Score Formula (see Equation 3.5) of three Critical (9) vulnerabilities that have not been treated for more than 365 days multiplied by 4, which is the highest value possible that an asset can have for EDP (Diamond). Equation 3.3 structures the equation used to compute the maximum total score of vulnerabilities.

For the MS model version, instead of three critical vulnerabilities without treatment over a year, we consider only one vulnerability not treated over 365 days with the score of 10, and asset valued as Diamond (4), leading to a final score of 80. This new mechanism is done due to the fact that the MS version only considers one vulnerability. If the Total Score accepted remains with a value of 216 instead of 80, it would not be possible for the MS version to obtain results comparable with the other two because the result would always be too low.

In terms of maximum total score accepted of incidents in an asset, it was obtained by multiplying all the maximum values for all components of an incident (Operational Impact, Consequence Severity, and Security Classification). The maximum value is 4, thus leading to a value of 64. Equation 3.9 structures the equation used to compute the maximum total score accepted of incidents.

Once the versions are too complex to show all sub-components scores, only the total score of each asset will be presented.

## 5.2 Evaluation of Model Versions by Scenario

This section is devoted to compute each model version for each scenario. A brief description of each scenario is made, a comparison of the model versions follows highlighting the three top differences at hosts, and applications layers, while for services, all differences are presented and discussed.

### 5.2.1 Scenario 1

**Description**

Sce1 is the simplest scenario evaluated, where the number of vulnerabilities on applications and hosts is comprehended between zero and two ($-V$) and there are no dependencies between hosts ($MDep-$).

Appendix .2 describes the structure of the scenario. In the same Appendix, it is possible to see the list of vulnerabilities, dependencies, and incidents for each asset.

**Comparison**

The results were obtained for the three layers separately, and they can be seen in Figures 5.1, 5.2, and 5.3 for the hosts, applications, and services' layers respectively. The purpose of these figures is to compare results from all versions for to this specific scenario.

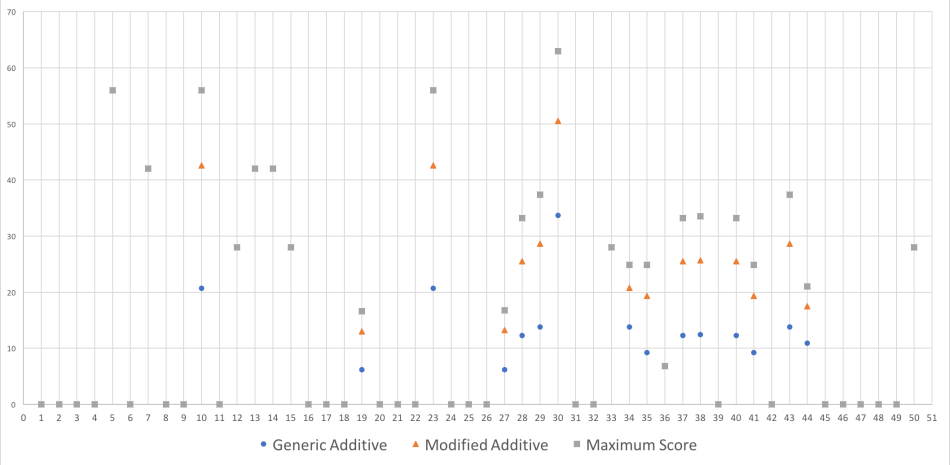



Figure 5.1: Comparison of versions of the model on the hosts' layer - Sce1

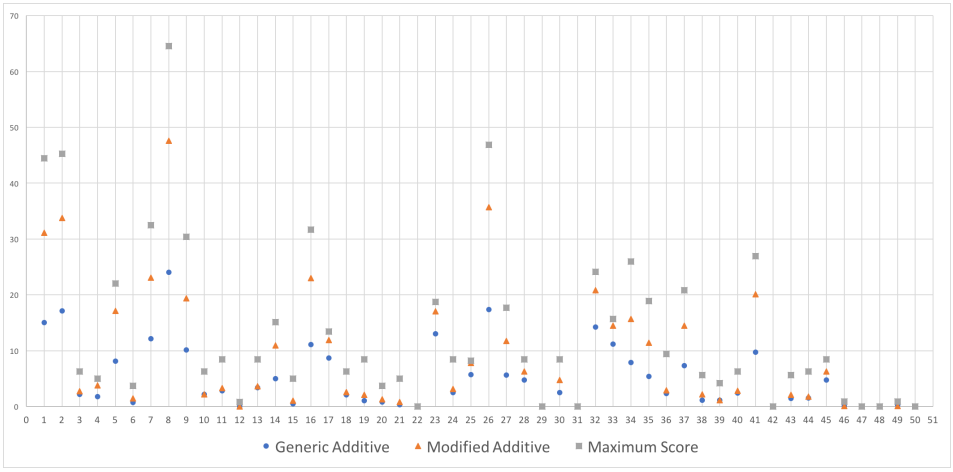



Figure 5.2: Comparison of versions of the model on the applications' layer - Sce1

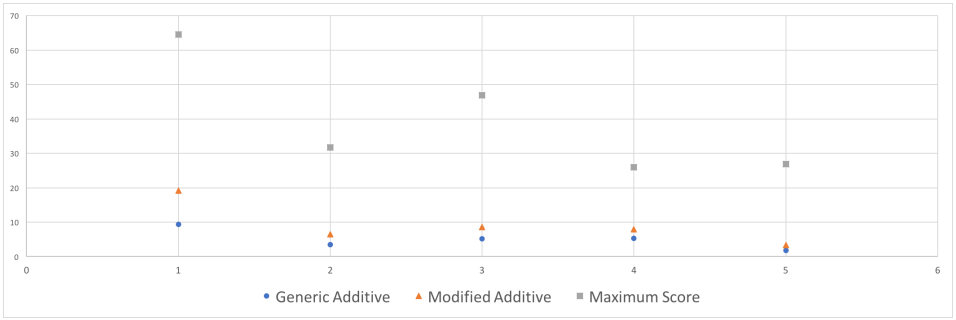



Figure 5.3: Comparison of versions of the model on the services' layer - Sce1

**Discussion**

Before we obtained any results, we intuitively expected to obtain higher scores with the MS version, followed by the MA, and sequentially, the GA version. The reason to expect such results is derived from the fact that the MS version only uses the highest scores on the assets, which can leads to higher scores.

In terms of comparison between the MA and the GA, we expected the MA to give us higher scores due to the fact that additional factors are taken into consideration since the GA is the most straightforward and simple version.

As observed in the previous figures, there are assets with different scores when applying the different versions, where the MS has the highest values followed by the MA, and finally, the GA as expected.

The relative differences between the results of the MA and the GA version, in percentage, is given by Equation 5.1.

$$Relative\ Difference = \frac{(MAValue\ -\ GAValue)}{GAValue} * 100 \tag{5.1}$$

*Where,*

*MAValue = Value of the score obtained by the Modified Additive version*

*GAValue = Value of the score obtained by the Generic Additive version*

For the MS version the differences were computed using a similar equation but considering the values obtained by the MS version, instead of the MA.

In regards to the host risk scores, the hosts that have the most considerable score differences are the hosts 'M29', 'M27', 'M44' with a range between 170 and 171 percent score increase comparing with the MS and the GA score. In terms of applications, the applications 'A1', 'A9', and 'A34' with a range between 196 and 226 percent score increase comparing the MS with the GA score. Finally, for the services, the range of differentiation of scores is comprehended between 390 and 1394 percent score increase comparing the MS with the GA score. Table 5.3 represents the percentile relative difference of results between the GA version and the other two versions results. GA was considered as basis for comparison because it provides the lower scores in this scenario for the assets.

| **Asset** | **MA** | **Maximum Score** |
|---|---|---|
| M27 | 112,90% | 170,96% |
| M29 | 107,25% | 171% |
| M34 | 50,72% | 80,43% |
| A1 | 105,80% | 170,53% |
| A9 | 92% | 200,9% |
| A34 | 98,7% | 229,1% |
| S1 | 106,4% | 594,6% |
| S2 | 85,7% | 805,7% |
| S3 | 65,3% | 801,9% |
| S4 | 49% | 390,5% |
| S5 | 83,3% | 1394,4% |

Table 5.3: Relative Differences between the GA and MA, as well as, the GA with the MS version

Hosts 'M27', and 'M29' have big score differences between versions due to the GA and MA versions consider the quantity and severity of all vulnerabilities, while MS considers only the highest scored vulnerability. Since, these two versions used other factors, it reduces the importance of the highest scored vulnerability, leading to a difference when compared with the MS version because the MS version does not reduce the importance of the highest scored vulnerability. This scenario has a low number of dependencies and vulnerabilities per asset, which aggravates the differences when compared to the MS version because the additional factors of GA and MA will not create a sufficient impact.

After analyzing the results, we came across with a pattern that makes the differentiation between scores larger as the level of abstraction of the model increases. In other words, the differences are smaller when we are comparing assets at the hosts' layer than when we are comparing at the applications' layer. The same happens between the applications' layer and the services' layer.

This pattern happens due to the fact that the GA and MA versions consider all dependencies of the evaluated asset, even if those dependencies have a low score. The problem of considering these low scored dependencies is the fact that they will always decrease the score of the asset, specially when the dependencies have very low scores. For instance, an application with three hosts supporting it, which has only one of the hosts with vulnerabilities. The score of the dependencies variable will vary dramatically if the calculus is based on only one host with vulnerabilities, instead of based on three hosts, where only one of them contributes with vulnerabilities. This situation happens with frequency once the majority of the applications have no vulnerabilities and are also supported by hosts with no vulnerabilities as well. Since these applications have a lower score due to that, the services they support will have even lower scores because the services' scores are based on the applications' scores.

Table 5.4 presents the standard deviation value for the risk score obtained for each model version for each layer. Higher values are derived as a high $stdev$ value represents a higher spread of values and consequently, a higher granularity.

| $\sigma$ | **Generic Additive** | **Modified Additive** | **Maximum Score** |
|---|---|---|---|
| Hosts | 14,1 | 16,7 | 19,3 |
| Applications | 5,6 | 10,9 | 14,2 |
| Services | 2,4 | 5,3 | 14,7 |

Table 5.4: Standard deviation value for the risk score obtained for each model version for each layer

### 5.2.2 Scenario 2

**Description**

Sce2 is quite similar to Sce1, being the only differences the number of dependencies between hosts. This number is comprehended between zero and three ($MDep+$) while in Sce1 there ware no dependencies between hosts. In Appendix .3 the structure of this scenario is described.

**Comparison**

Figures 5.4, 5.5, and 5.6 represent the risk scores obtained for the hosts, applications, and services' layers respectively for each model version.

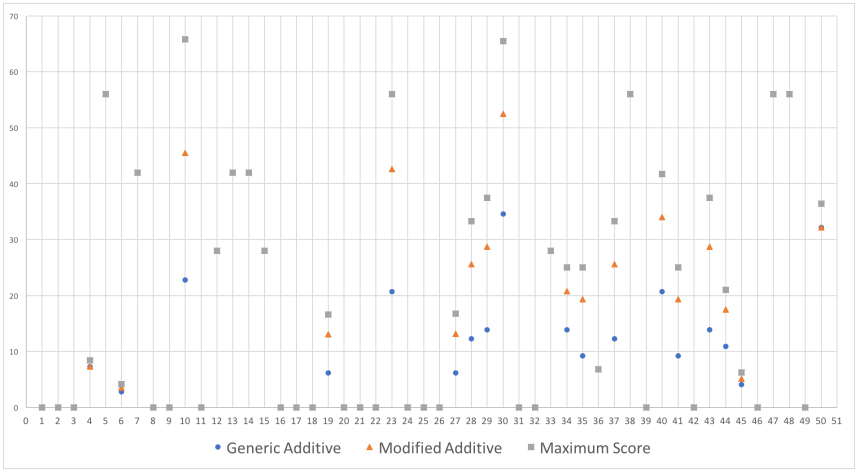



Figure 5.4: Comparison of versions of the model on the hosts' layer - Sce2

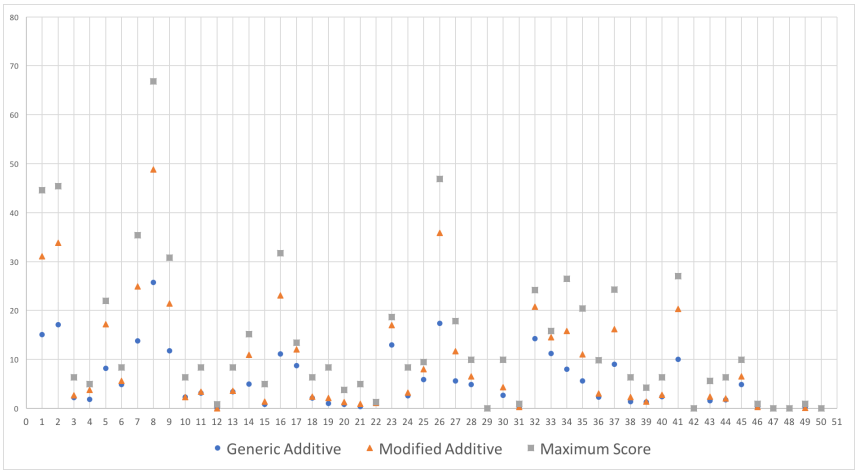



Figure 5.5: Comparison of versions of the model on the applications' layer - Sce2

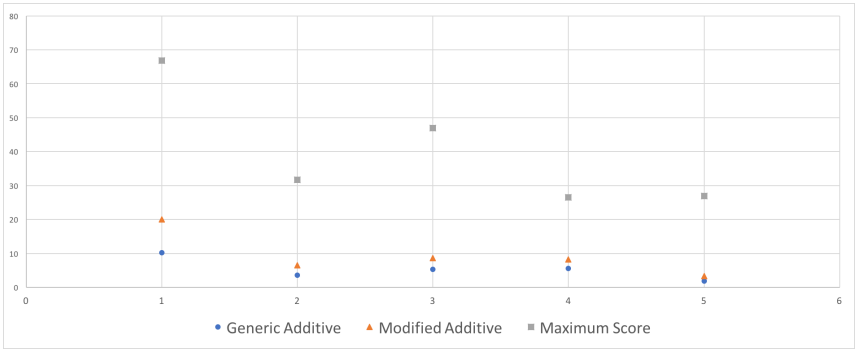



Figure 5.6: Comparison of versions of the model on the services' layer - Sce2

**Discussion**

The obtained results are in line with the same expectation we had for Sce1 and they are very similar with the results obtained for Sce1. The MS version is the one with the higher scores, followed by the MA, and sequentially, the GA version.

The hosts that have the most considerable difference of scores are 'M10', 'M35', 'M27' with a range between 170 and 188 percent score increase in the MS version relatively to the GA version. In terms of applications, the applications 'A1', 'A16', and 'A34' with a range between 195 and 231.5 percent score increase comparing the MS with the GA score. Finally, for the services, the range of differentiation of scores is comprehended between 373 and 1321 percent score increase comparing the MS with the GA score. Table 5.5 represents the differentiation of results in percentage between the GA version and the other two versions, being the GA version that provides the lower scores in this scenario.

| **Asset** | **Modified Additive** | **Maximum Score** |
|---|---|---|
| M10 | 99,5% | 188,5% |
| M27 | 112,9% | 170,9% |
| M35 | 109,7% | 171,7% |
| A1 | 105,9% | 195,3% |
| A16 | 108,1% | 185,5% |
| A34 | 97,5% | 231,2% |
| S1 | 94,1% | 548,5% |
| S2 | 80,5% | 780,5% |
| S3 | 61,1% | 768,5% |
| S4 | 46,4% | 373,2% |
| S5 | 78,9% | 1321% |

Table 5.5: Relative Differences between the GA and MA, as well as, the GA with the MS version

After an analysis of the results, we have concluded that the differentiation has decreased in general, even if its on a small amount. The reason for this decrease is the fact the assets have higher scores in the GA version due to more dependencies. Since several dependencies between hosts were added, an increase of scores of hosts were noticed once the scores of the dependencies variable of each host increase. Nevertheless, the same pattern discussed on the discussion of the Sce1 was detected in this scenario as well and the reason remained the same.

At the global view, the score of the services, applications, and hosts have increased as expected once there are new dependencies between hosts. Further in this chapter, a deeper explanation about the differences between scenarios is made.

Table 5.6 presents the standard deviation value for the risk score obtained for each model version for each layer. Higher values are derived as a high $stdev$ value represents a higher spread of values and consequently, a higher granularity.

| $\sigma$ | **Generic Additive** | **Modified Additive** | **Maximum Score** |
|---|---|---|---|
| Hosts | 17,6 | 19,3 | 21,5 |
| Applications | 5,7 | 11,0 | 14,4 |
| Services | 2,8 | 5,6 | 15,3 |

Table 5.6: Standard deviation value for the risk score obtained for each model version for each layer

### 5.2.3 Scenario 3

**Description**

Sce3 is a scenario where there are no dependencies between hosts ($MDep-$), but has a number of vulnerabilities in hosts ranged between zero and six ($+V$).

The set of vulnerabilities in this scenario, and in Sce4 that will be discussed further, is completely different from the set of vulnerabilities from Sce1 and Scenario 2. In Appendix .4, the structure of Sce3 is described.

**Comparison**

Figures 5.7, 5.8, and 5.9 represent graphically the scores obtained for the hosts, applications, and services respectively for each model version.

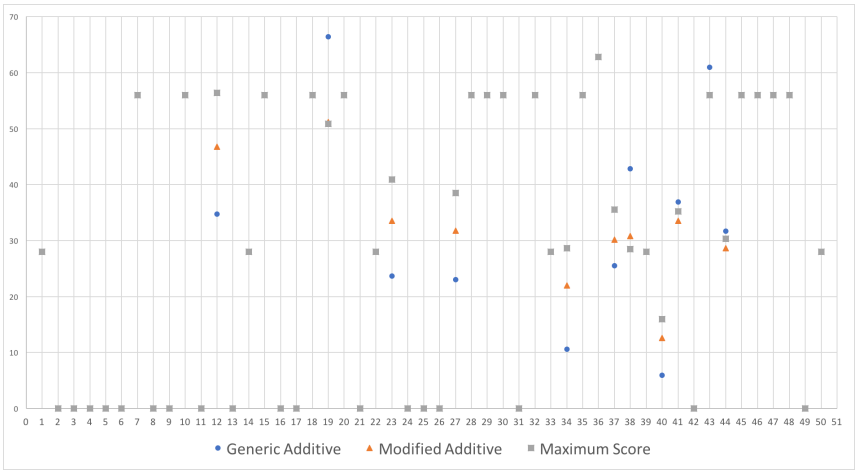



Figure 5.7: Comparison of versions of the model on the hosts' layer - Sce3

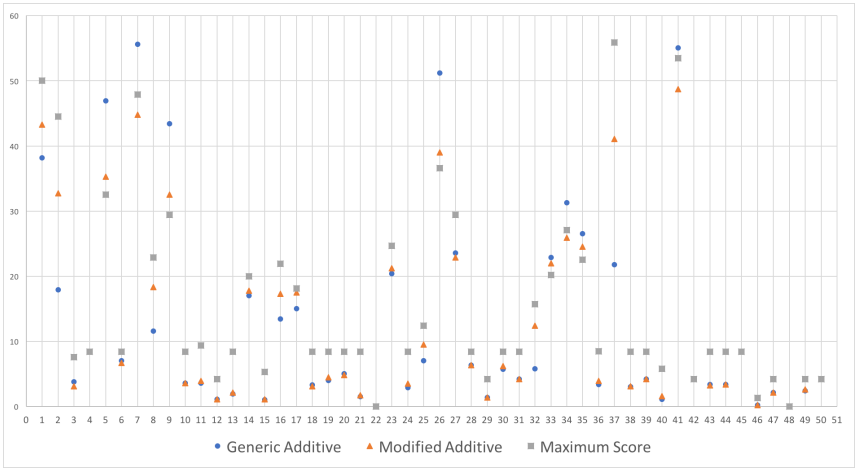



Figure 5.8: Comparison of versions of the model on the applications' layer - Sce3

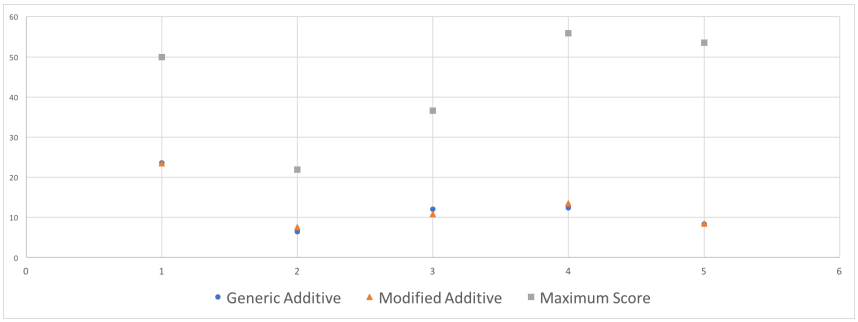



Figure 5.9: Comparison of versions of the model on the services' layer - Sce3

**Discussion**

Before obtaining the results, we expected to have results in the same line of behavior of the Sce1 and Sce2 but that did not happen. These results shown us a completely different variation in terms of which version of the model has the highest scores. As can be seen in the previous figures, the MS and the GA swap places constantly, in terms of which has the highest score.

The hosts that have the most considerable differences of results are the hosts 'M40', 'M34', 'M27' with a range between 67 and 171 percent score increase when comparing the MS with the GA score. In terms of applications, the applications 'A2', 'A32', and 'A37' with a ranged between 148 and 170 percent score increase comparing the MS with the GA score. Finally, for the services, the range of differentiation of scores is comprehended between 111 and 544 percent score increase comparing the MS with the GA score. Table 5.7 represents the differentiation of results in percentage between the GA version and the other two versions, as the GA is the version that provides the lower scores in this scenario for the assets indicated previously.

| Asset | Modified Additive | Maximum Score |
|---|---|---|
| M40 | 113,5% | 171,1% |
| M34 | 107,4% | 169,8% |
| M27 | 38,2% | 67,3% |
| A2 | 82,6% | 148,6% |
| A32 | 113,7% | 170,6% |
| A37 | 88,5% | 156,4% |
| S1 | -0,4% | 111,8% |
| S2 | 16,9% | 236,9% |
| S3 | -10% | 205% |
| S4 | 8,8% | 350,8% |
| S5 | 2,4% | 544,7% |

Table 5.7: Relative Differences between the GA and MA, as well as, the GA with the MS version

After the analysis of the results, we could identify why the MS and GA versions have swapped constantly the places, in terms of which has the highest score.

In this scenario, the number of vulnerabilities per asset has increased from two to six as the maximum limit, causing changes on the GA and MA versions.

The GA and MA versions consider all vulnerabilities present on an asset, which is the opposite of the MS. The MS takes into account only the highest severity vulnerability, regardless if there are several other vulnerabilities with a high score as well. This mechanism creates a tie when we have two different assets with equally highest scored vulnerabilities but in one of the assets has a number of other vulnerabilities that are highly severe as well. In this particular case, the MS will present the same result for the different assets, which might not give the best perception of risk. On the other hand, the GA and MA versions have more sensibility to the quantity of vulnerabilities, and for their severities.

As mentioned previously, the GA version equalizes the importance of each vulnerability, pondering other vulnerabilities that may create a unwanted impact as well, and the MA version has the factors of SAVBHO and the number of vulnerabilities. With these particular mechanisms, the GA and the MA versions can be more precise and higher scored when compared with the MS version for assets that have a considerable amount of vulnerabilities.

When we were checking the quantity of vulnerabilities that each asset has in Sce3, we realized that the GA and the MA versions produced higher results than the MS for any asset having more than two or three vulnerabilities. For all those assets which they did not have a two or less vulnerabilities, the highest score was given by the MS version.

Since there are some assets for which the GA version gives a score higher than that of the MS, there are assets that present a difference of scores negative.

Also important to refer is the fact that the differences between versions of the model

are smaller than in the previous scenarios. The main reason for this is the fact that the number of vulnerabilities have increased, allowing for the GA and MA versions to explore their potential by having higher values in factors such as SAVBHO, and Number of Vulnerabilities. Nevertheless, the Services' layer remains with a large difference due to the fact that the majority of the applications have a low scored risk causing a decrease on the total score of the services that depend on those applications.

Table 5.8 presents the standard deviation value for the risk score obtained for each model version for each layer. Higher values are derived as a high $stdev$ value represents a higher spread of values and sequentially, a higher granularity.

| $\sigma$ | **Generic Additive** | **Modified Additive** | **Maximum Score** |
|---|---|---|---|
| Hosts | 24,5 | 23,8 | 24,0 |
| Applications | 15,3 | 13,8 | 14,3 |
| Services | 5,9 | 5,7 | 12,7 |

Table 5.8: Standard deviation value for the risk score obtained for each model version for each layer

### 5.2.4 Scenario 4

**Description**

Sce4 is the most complex scenario evaluated, where the number of vulnerabilities is ranged between zero and six ($+V$) and the number of dependencies between hosts is comprehended between zero and three ($MDep+$). The sets of vulnerabilities in this scenario are the same in Sce3, being the only difference the number of dependencies between hosts. Appendix .5 describes the structure of this scenario.

**Comparison**

Figures 5.10, 5.11, and 5.12 represent the score obtained for the hosts, applications, and services' layers respectively for each version created.

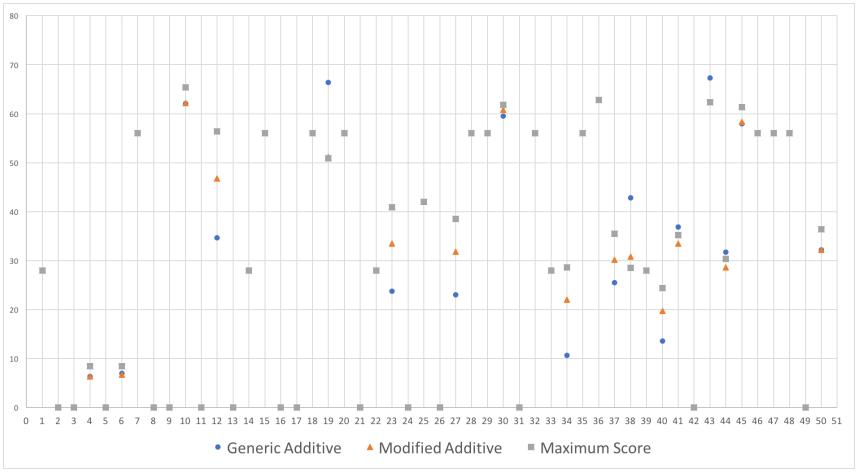



Figure 5.10: Comparison of versions of the model on the hosts' layer - Sce4

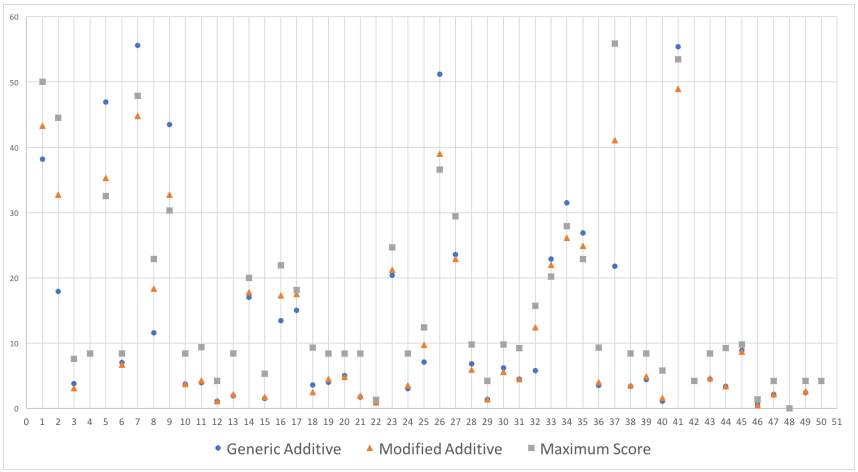



Figure 5.11: Comparison on the applications' layer - Sce4

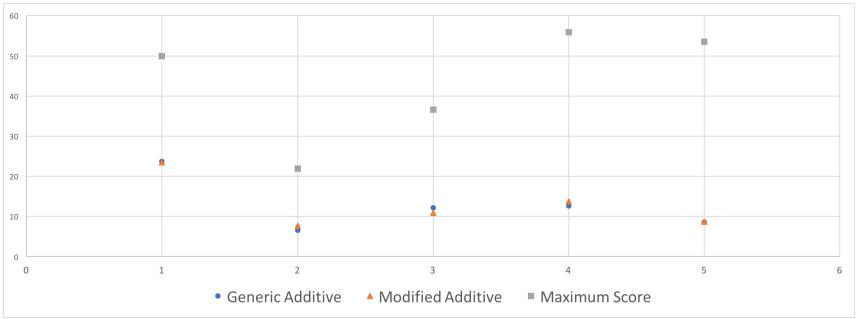



Figure 5.12: Comparison of versions of the model on the services' layer - Sce4

**Discussion**

In a very similar way with the Sce3, we expected that the MS version would provide higher scores, followed by the Modified and then by the GA version but the results did

not shown that. The results have shown us a similar behavior as the Sce3, with higher scores as the major differences due to the new set of dependencies.

The hosts that have the most considerable difference of results are the hosts 'M34', 'M40', 'M23' with a range between 72 and 169 percent score increase comparing the MS with the GA score. In terms of applications, the applications 'A2', 'A8', and 'A37' with a ranged between 97 and 156 percent score increase comparing the MS with the GA score. Finally, for the services, the range of differentiation of scores is comprehended between 110 and 522 percent score increase comparing the MS with the GA score. Table 5.9 represents the differentiation of results in percentage between the GA version and the other two versions, due to the GA being the version that does provide the lower scores in this scenario for the assets indicated previously.

| **Asset** | **Modified Additive** | **Maximum Score** |
|---|---|---|
| M40 | 44,8% | 79,4% |
| M34 | 107,5% | 169,8% |
| M23 | 41,3% | 72,5% |
| A2 | 82,6% | 148,6% |
| A8 | 57,7% | 97,4% |
| A37 | 88,5% | 156,4% |
| S1 | -0,8% | 110,9% |
| S2 | 16,6% | 231,8% |
| S3 | -10,6% | 200% |
| S4 | 8,8% | 350,8% |
| S5 | 1,1% | 522% |

Table 5.9: Relative Differences between the GA and MA, as well as, the GA with the MS version

After analyzing the results, we detected the same behavior detected from Sce3, being the only difference is on the higher scores from this scenario when compared with the previous one. The differences of risk scores between this scenario and Sce3 is the fact that this scenario has more dependencies than the Sce3. Once again, it was possible to see that the differences between versions of the model are smaller for the other scenarios due to the increase of the scores of assets in the lowest layers.

Table 5.10 presents the standard deviation value for the risk score obtained for each model version for each layer. Higher values are derived as a high $stdev$ value represents a higher spread of values and sequentially, a higher granularity.

| $\sigma$ | **Generic Additive** | **Modified Additive** | **Maximum Score** |
|---|---|---|---|
| Hosts | 24,3 | 23,6 | 24,0 |
| Applications | 15,2 | 13,7 | 14,3 |
| Services | 5,9 | 5,6 | 12,7 |

Table 5.10: Standard deviation value for the risk score obtained for each model version for each layer

### 5.2.5 Discussion of the Scenarios

After analyzing all scenarios and the behavior of the versions, we realized that the MS version is the most appropriated version to be used in the case of systems with characteristics similar to Sce1 and Sce2, where the number of vulnerabilities is small.

As mentioned previously, the MS version does not consider other vulnerabilities but the highest one, leading to the problem of misinterpreting the asset's risk by not taking into account other vulnerabilities that can create a considerable impact besides the highest one. Sce1 and Sce2 consider a low number of vulnerabilities per asset, which reduces the miscalculation of the asset's risk by the MS version. In case there is a considerable amount of vulnerabilities per asset, the MS version would not correctly assess the risk as shown in Sce3 and Sce4.

Sce3 and Sce4 represent systems that have a higher number of vulnerabilities per asset and a higher number of dependencies per asset as well. In these types of scenarios, the most advisable version is the GA version. This version has an advantage, compared with the MA in our experiments, because to the MA having its Sum All Vulnerabilities But the Highest One (SAVBHO) factor with a weight too small in comparison with the weight of the Highest Scored Vulnerability factor. Nevertheless, these weights can be reconfigured in order to obtain different results though the simplicity of the GA version makes the GA version adaptable to every system, without the necessity of trial and error procedures to find the right weights of all factors.

We have also realized that the GA and MA versions have a considerable decrease of value when assessing an asset that has a lot of dependencies with low risk, or has vulnerabilities with low scores. This behavior can be seen when assessing the Services' layers at any scenario due to several applications having a low scored risk, causing major differences between the versions as mentioned on the discussions of the scenarios.

During the presentation of the major differences that exist between versions, it is important to indicate that there were assets that had scores so low in the GA version that a relative difference with the MS version would lead us to 1322 percent, or higher, in terms of differentiation between versions. As an example in Sce1, the application 'A21' had a scored of 0,3 when using the GA version, and with the MS, the score was changed to 5. This difference of 4.7 units is led to a 1566,6 % difference. In order to properly analyze

the differentiation between scores that can cause impact on the system, we removed all scores below 10 in the GA version.

In terms of the MA version, this version of the model is the version of the middle-terms, meaning that this version is consistent and adaptable to every scenario. For Sce1 and Sce2, this version has a behavior more similar with the MS. However, for the Sce3 and Sce4, this version behaves similarly with the GA. Even if this version does not have a single score that is considered the highest score for an asset, this version is capable to provide reasonable results regardless of the scenario allowing us to conclude that this version is the best version for an organization to adopt.

Both Sce 1 and Sce2 have a low number of vulnerabilities in total and per asset, and the differences between versions is more accentuated, specifically between the MS and the other two versions. In Sce3 and Sce4, which have a higher number of vulnerabilities in comparison with Sce1 and Sce2, the differences are less accentuated. This phenomenon happens because to the GA and MA versions considering all vulnerabilities and dependencies for the assessment, meaning that when an asset has a considerable number of vulnerabilities, these versions have a more realistic and higher score comparing with assets that have a low number of vulnerabilities and dependencies. The justification for this to happen is the factors that both versions have to consider the situations when the asset have more vulnerabilities. The MA version has the factor SAVBHO and the number of vulnerabilities that, when an asset has a low number of vulnerabilities, both of this factors will not have a considerable contribute. Even further, since these factors have a contribution for the formula itself of the MA, the impact of the highest vulnerability will not be totally considered (see Table 5.2).

Table 5.11 resumes the most advisable version for each scenario.

| **Scenario** | **version** |
|---|---|
| Sce1 | Maximum Score |
| Sce2 | Maximum Score |
| Sce3 | Generic Additive |
| Sce4 | Generic Additive |

Table 5.11: The most advisable version for each scenario

In summary, for a scenario with a low number of dependencies and vulnerabilities, the MS version is the most suited version because it is not necessary to consider other vulnerabilities but the highest ones to have a realistic notion of the risk score of the system. However, for a scenario with a high number of dependencies and vulnerabilities, similar with Sce3 and Sce4, the GA version has a more realistic view of the risk scores because it considers all vulnerabilities as discussed previously. The MA version is more suitable for scenarios with similarities with Sce3 and Sce4 as well because it considers other factors than the highest scored vulnerability, as mentioned previously as well.

The information system of EDP is not exactly the same as one of these four scenarios studied because the number of vulnerabilities is not that high, when compared with Sce3 or Sce4, and it is not that low, when compared with Sce1 and Sce2. Nevertheless, we consider the most suitable version for EDP is the MA version, because this version is sufficiently realistic to provide a correct notion of the state and risk score of the system.

## 5.3 Scenarios Comparison

### 5.3.1 Impact of the Number of Vulnerabilities

In order to assess the impact of the number of vulnerabilities in the scores of the system, we decided to compare Sce1 versus Sce3 as well as Sce2 versus Sce4. The main reasons for these comparisons consist in the fact that the number of vulnerabilities increases from Sce1 to Sce3 and from Sce2 to Sce4, and being the only differences between the Scenarios as stated on Table 5.1.

**Scenario 1 and Scenario 3 Comparison**

Figures 5.13, 5.14, 5.15 represent both Sce1 and Sce3 score comparisons when using GA version for the hosts, applications, and services respectively. The reason for using the GA version consists in the fact that this version had the lowest scores in Sce1 and the highest ones in Sce3, allowing to show the biggest differences between results and to facilitate our conclusions about the impact of the number of vulnerabilities has in the systems.

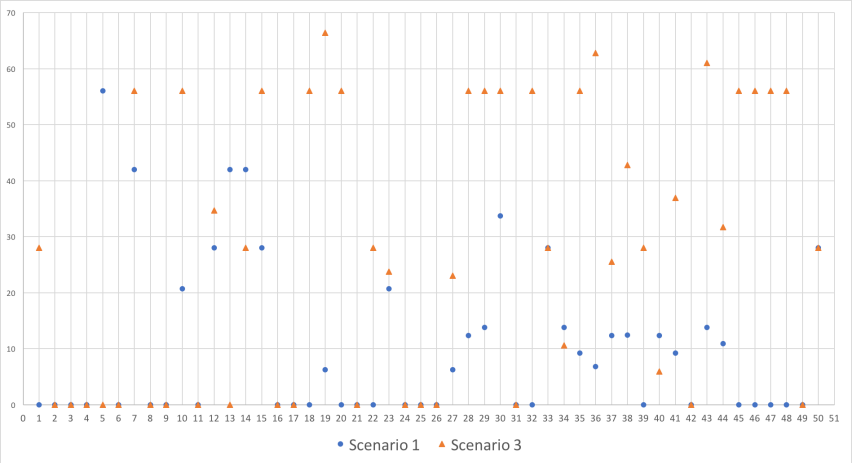



Figure 5.13: Comparison between Sce1 and Sce3 on the hosts' level

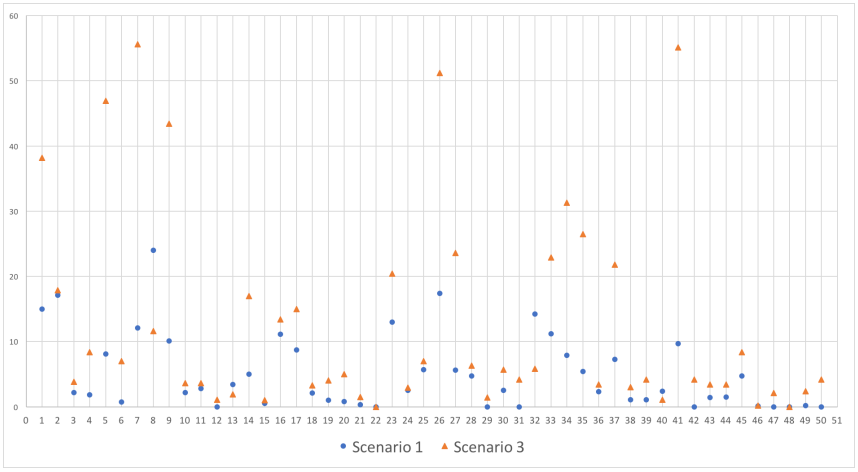



Figure 5.14: Comparison between Sce1 and Sce3 on the applications' level

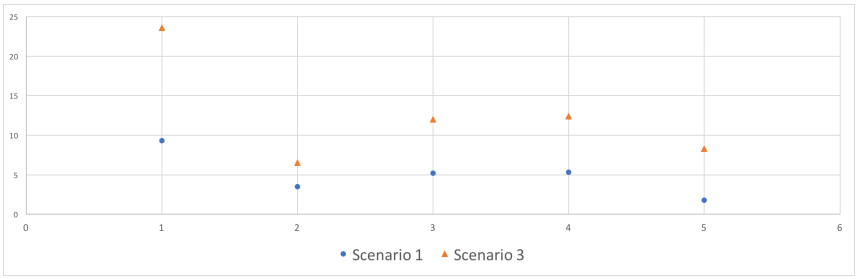



Figure 5.15: Comparison between Sce1 and Sce3 on the services' level

**Discussion**

Tables 5.12 and 5.13 justify some differences between both scenarios by indicating the Asset ID and the respective justification. It is important to refer that not all assets are indicated in the table. Those assets that are not in the table have similar reasons to have their score changed with assets that are already in the table. As an example, the asset M1 and M15 have the same reason of having their score changed, however only the M1 is presented on the table.

| Asset | Justification |
|---|---|
| M1 | The increase of the risk is based on the support to the application A26. In Sce1, A26 has only one vulnerability with the Scope Unchanged and in Sce3 has 6 vulnerabilities, one of them having its Scope Changed. Once the Scope its changed, automatically M1 will have its score increased due to be related directly with the application A26. |
| M5 | In Sce1, this asset has a higher score when comparing to Sce3 due to a vulnerability on application A32. This vulnerability has its Scope Changed in Sce1 but does not have its Scope Changed in Sce3, meaning this asset is affected only on Sce1. |
| M7 | In the same line of justification, this asset has higher score in Sce3 due to the application A9 having a vulnerability with its Scope Changed on Sce3. This asset itself already has a score, however, the impact of the vulnerability on the A9 is higher, updating the score on Sce3. |
| M12 | Similar with host M10, where the difference is the application A25 instead of the A35. |
| M19 | The number of vulnerabilities in Sce3 increased from 1 to 6. In the set of vulnerabilities in Sce3, there is a vulnerability with a score of 9, leading to an increase in the score. |
| M23 | The number of vulnerabilities in Sce3 increased from 1 to 3, leading to a higher score. |
| M27 | Both Sce1 and Sce3 have the application A37, supported by the host M27, with one vulnerability. Nevertheless, in Sce3, the score of the vulnerability is 9 against the score of 4 on the Sce1. |

Table 5.12: Differences of scores between Sce1 and Sce3 PartI

| Asset | Justification |
|---|---|
| M34 | The number of vulnerabilities in Sce3 decreased from 2 to 1, leading to a lower score. The value of the scores on Sce1 are 8 and 4. On Sce3, the value is 4. |
| M37 | The number of vulnerabilities in Sce3 increased from 1 to 2, leading to a higher score.The values of the vulnerabilities on Sce3 are 8 and 9. On Sce1, the vulnerability has a score of 8. |
| M38 | The number of vulnerabilities in Sce3 increased from 1 to 5, leading to a higher score.The values of the vulnerabilities on Sce3 are 6.5, 6.5, 9, 4, 4, and 6. On Sce1, the vulnerability has a score of 8. |
| M40 | Both Sce1 and Sce3 have the host M40 with one vulnerability. Nevertheless, in Sce3, the score of the vulnerability is 4 against the score of 8 on the Sce1. |
| M43 | The number of vulnerabilities in Sce3 increased from 1 to 5, leading to a higher score. |
| M44 | The number of vulnerabilities in Sce3 increased from 2 to 3, leading to a higher score. |
| A1 | In Sce1 this asset has only one vulnerability with a score of 8 and in Sce3 has 3 vulnerabilities with the scores of 9, 8, and 4, which leads to a higher score in Sce3. |
| A3 | This asset has its score aggravated in Sce3 due to the fact that assets M13 and M19 have their scores increased. |
| A5 | This asset has more vulnerabilities in Sce3 than the Sce1. Also, the assets that support this application had their scores increased as well. |
| A8 | In Sce1 this asset had one vulnerability with a score of 9 and in the Sce3 had only one vulnerability, as well, however the score was 4 instead of 9. |
| A26 | The number of vulnerabilities have increase from 1 to 6 from the Sce1 to Sce3 respectively. |
| A41 | The number of vulnerabilities have increase from 1 to 4 from the Sce1 to Sce3 respectively. Nevertheless, the set of vulnerabilities in Sce3 have higher scores in general. |
| S1 to S5 | In each service, the score of the majority of the applications have increased. Since that happened, the services will have their scores changed as well. |

Table 5.13: Differences of scores between Sce1 and Sce3 PartII

**Scenario 2 and Scenario 4 Comparison**

Figures 5.16, 5.14, 5.15 represent both Sce2 and Sce4 comparisons when using GA version for the Hosts, Applications, and Services respectively.

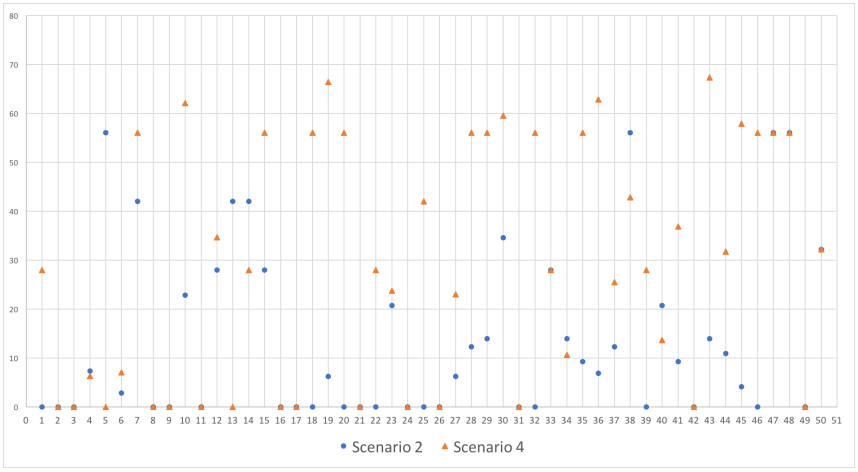



Figure 5.16: Comparison between Sce2 and Sce4 on the hosts' level

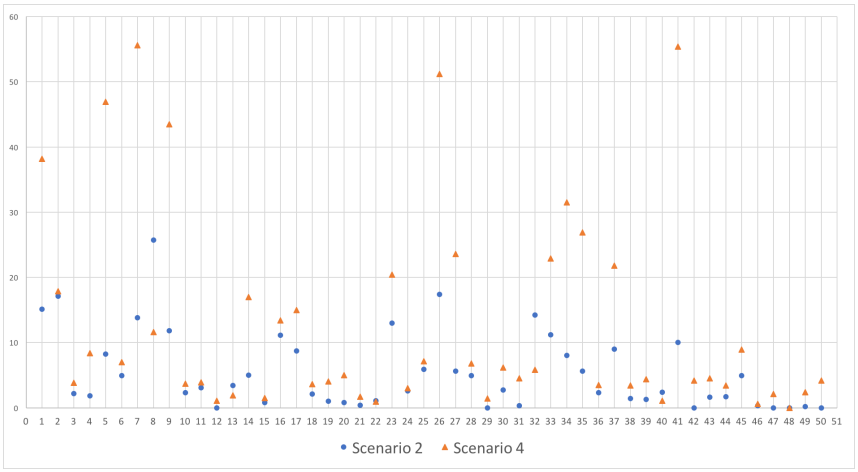



Figure 5.17: Comparison between Sce2 and Sce4 on the applications' level

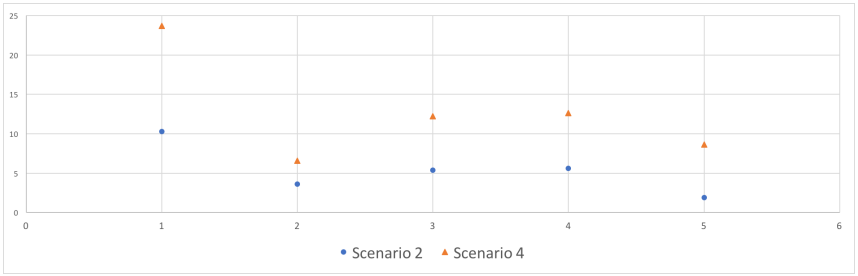



Figure 5.18: Comparison between Sce2 and Sce4 on the services' level

**Discussion**

Table 5.14 justifies some differences between both scenarios by indicating the Asset ID and the respective justification. It is important to refer that not all assets are indicated in the table. Assets in the table were selected because the reasons to have their score changed are representative.

| Asset | Justification |
|---|---|
| M1 | This asset supports the Application 'A26' which has a vulnerability with the Scope Changed on Sce4 and does not have it on Sce2. |
| M14 | This assets decreases its score from Sce2 to Sce4 due to the vulnerability in Sce3 having a score of 9, and on Sce4 the same vulnerability has a score of 4. |
| M30 | This asset supports the application 'A9', which has a vulnerability with the score of 8 and ts Scope Changed in Sce4. |
| M41 | This assets have a increase of number of vulnerabilities from 1 to 4 vulnerabilities in Sce4, causing an increase of the score of the asset. |
| A1 | This assets have a increase of number of vulnerabilities from 1 to 3 vulnerabilities in Sce4, causing an increase of the score of the asset. |
| A8 | This assets decreases its score from Sce2 to Sce4 due to the vulnerability in Sce3 having a score of 9, and on Sce4 the same vulnerability has a score of 4. |
| A41 | This assets have a increase of number of vulnerabilities from 1 to 4 vulnerabilities in Sce4, causing an increase of the score of the asset. |
| S1 to S5 | In each service, the score of the majority of the applications have increased. Since that happened, the services will have their scores changed as well. |

Table 5.14: Differences of scores between Sce1 and Sce3 PartII

**Discussion of the impact of the number of vulnerabilities**

After an analysis of the comparisons between Sce1-Sce3 and Sce2-Sce4, it was possible to understand the impact of the number of vulnerabilities on the risk score. As mentioned previously, the difference between the scenarios that were compared between each other was the number of vulnerabilities, where the Sce1 and Sce2 had a lower number of vulnerabilities, and the Sce3 and Sce4 had a higher number of them.

Table 5.15 shows the absolute differences, in percentage, between the scores of assets when comparing Sce3 in an absolute way to Sce1 (because Sce1 has lower scores), and also when comparing Sce4 in an absolute way to Sce2. The differences were made based on Sce1 and Sce2 scores compared with the Sce3 and Sce4 respectively. In cases of having negative percentage, it means that an asset on Sce1 or Sce2 had a higher score than on Sce3 or Sce4. The assets to consider in this analysis were the ones that were identified in previous Tables 5.12, 5.13, and 5.14.

| Asset | Differences - Sce1 and Sce3 | Differences - Sce2 and Sce4 |
|---|---|---|
| M1 | 28 | 28 |
| M5 | -56 | - |
| M7 | 14 | - |
| M12 | 6,7 | - |
| M14 | - | -14 |
| M19 | 60,2 | - |
| M23 | 3 | - |
| M27 | 16,8 | - |
| M30 | - | 24,9 |
| M34 | -23,1 | - |
| M37 | 13,2 | - |
| M38 | 30,4 | - |
| M40 | -6,4 | - |
| M41 | - | 27,7 |
| M43 | 47,2 | - |
| M44 | 20,8 | - |
| A1 | 23,2 | 23,1 |
| A3 | 1,6 | - |
| A5 | 38,8 | - |
| A8 | -12,4 | -14,1 |
| A26 | 33,8 | - |
| A41 | 45,4 | 45,4 |
| S1 | 14,3 | 13,4 |
| S2 | 3 | 3 |
| S3 | 6,8 | 6,8 |
| S4 | 7,1 | 7 |
| S5 | 6,5 | 6,7 |

Table 5.15: Differences of scores between Sce1 and Sce3, and also between Sce2 and Sce4

The differences of scores that can be seen in the Table 5.15 clearly indicates that adding new vulnerabilities into the system leads to a higher impact on the scores of the assets. This situation occurs because the weight considered for the vulnerabilities variable is 0,7, which is remarkably higher when compared to the dependencies and incidents variables, which are 0,15 each.

### 5.3.2 Impact of the Number of Dependencies

In order to assess the impact of the number of dependencies in the risk score, we decided to compare Sce1 versus Sce2 as well as Sce3 versus Sce4. The main reasons for these comparisons consist in the fact that the number of dependencies increases from Sce1 to Sce2 and Sce3 to Sce4, and being the only differences between the Scenarios as stated on Table 5.1.

**Scenario 1 and Scenario 2 Comparison**

Figures 5.21, 5.20, 5.21 represent between Sce1 and Sce2 comparisons when using GA version for the hosts, applications, and services respectively.

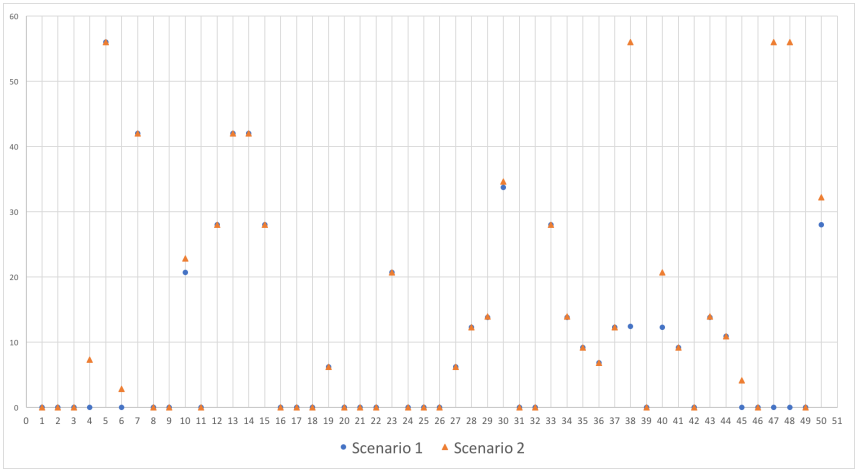



Figure 5.19: Comparison between Sce1 and Sce2 on the hosts' level

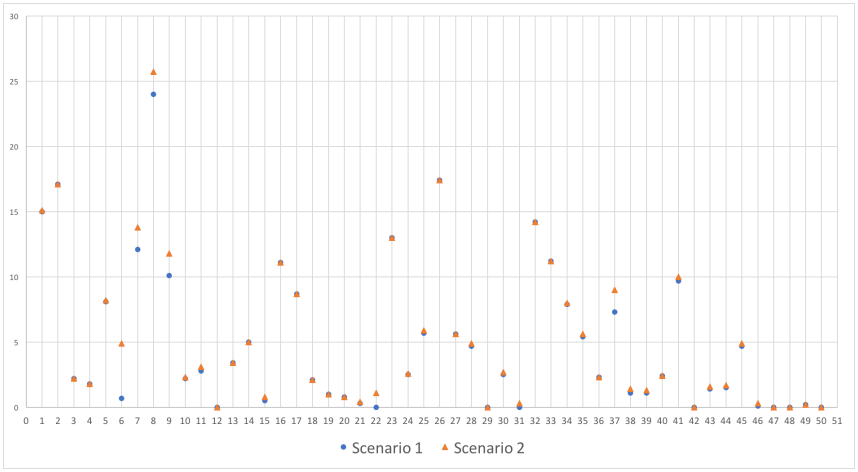



Figure 5.20: Comparison between Sce1 and Sce2 on the applications' level

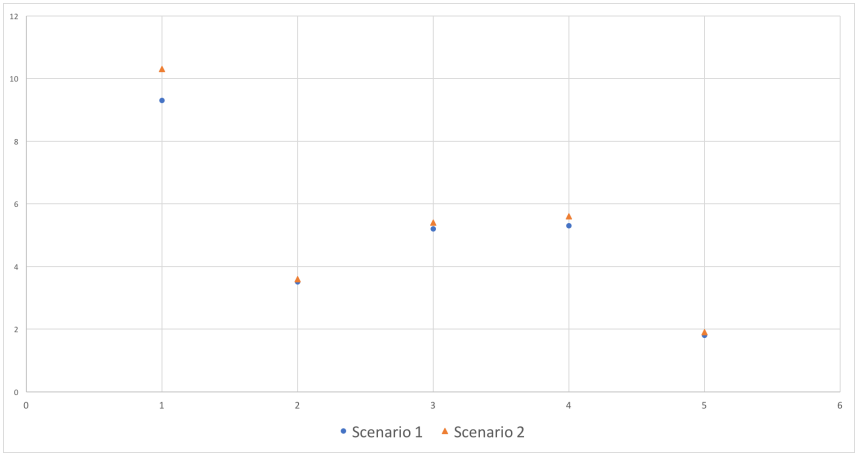



Figure 5.21: Comparison between Sce1 and Sce2 on the services' level

**Discussion**

Table 5.16 justifies some differences between both scenarios by indicating the Asset ID and the respective justification. It is important to refer that not all assets are indicated in the table. Those assets that are not in the table have similar reasons to have their score changed with assets that are already in the table.

| Asset | Justification |
|---|---|
| M38 | This asset supports the host 'M40' and this host has a vulnerability with a score of 8 and it has the Scope Changed. |
| M6 | This asset is supported by the other hosts 'M15' and 'M41'. These hosts have they own risk score, which are different from zero, and they contribute for the Dependencies variable of this asset. |
| M47 | This asset supports the host 'M40' and this host has a vulnerability with a score of 8 and it has the Scope Changed. |
| A6 | This asset has its score increase due to the host 'M48'. This host supports another host, 'M40', which has a vulnerability with the Scope Changed causing an increase on the score. Consequently, this asset has repercussions on this asset. |

Table 5.16: Differences of scores between Sce1 and Sce2

**Scenario 3 and Scenario 4 Comparison**

Figures 5.22, 5.23, 5.24 represent both Sce3 and Sce4 comparisons when using GA version for the hosts, applications, and services respectively.

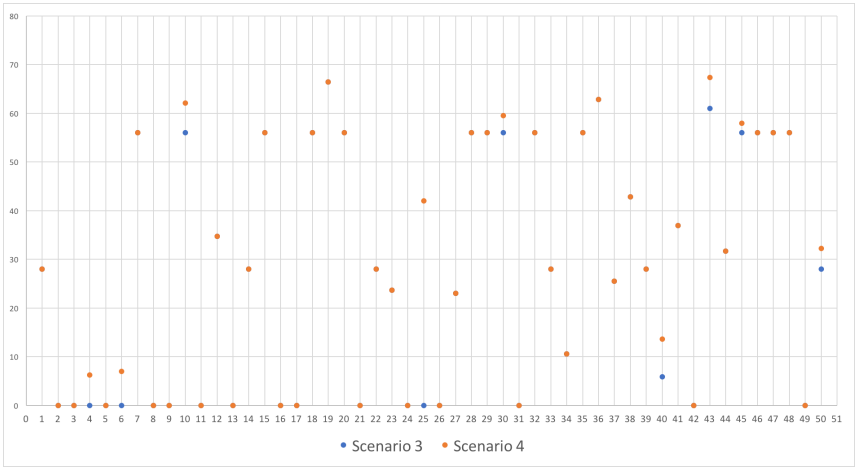



Figure 5.22: Comparison between Sce3 and Sce4 on the hosts' level

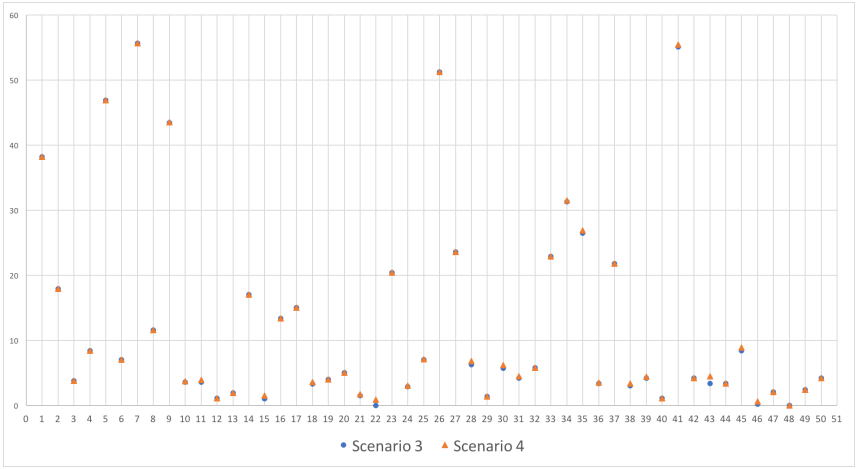



Figure 5.23: Comparison between Sce3 and Sce4 on the applications' level

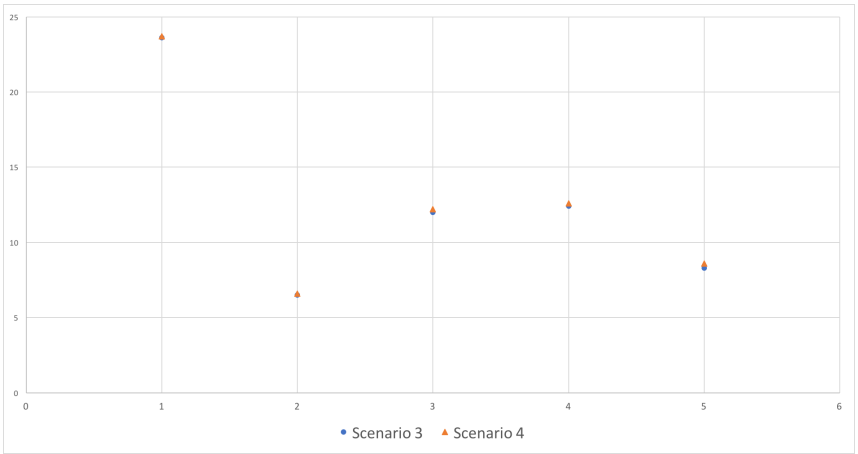



Figure 5.24: Comparison between Sce3 and Sce4 on the services' level

**Discussion**

Table 5.17 justifies some differences between both scenarios by indicating the Asset ID and the respective justification. It is important to refer that not all assets are indicated in the table. Those assets that are not in the table have similar reasons to have their score changed with assets that are already in the table. This table does not have several items due to the impact that the dependencies have. Further on the discussion of the impact of the dependencies, it will be explained why.

| Asset | Justification |
|---|---|
| M6 | This asset has its score increase due to the host 'M41' having vulnerabilities. |
| M25 | This asset supports the host 'M43', which has a vulnerability with its Scope Changed. |

Table 5.17: Differences of scores between Sce3 and Sce4

**Discussion of the impact of the number of dependencies**

After an analysis of the comparisons between Sce1-Sce2 and Sce3-Sce4, it was possible to understand the impact of the dependencies on the system. As mentioned previously, the differences between the scenarios that were compared between each other was the number of dependencies, where the Sce1 and Sce3 had a lower number of them, and the Sce2 and Sce4 had a higher number of them.

Intuitively, we expected to have higher scores on the scenarios that have more dependencies, when compared with the scenarios with a lower number of them, and that expectation was confirmed. Table 5.18 shows the differences, in percentage, between the scores of the assets when comparing Sce1 and Sce3, and also when comparing Sce2 and Sce4.

| Asset | Differences - Sce1 and Sce2 | Difference - Sce3 and Sce4 |
|---|---|---|
| M6 | 2,8 | 7 |
| M25 | - | 42 |
| M38 | 43,6 | - |
| M47 | 56 | - |
| A6 | 4,2 | - |

Table 5.18: Differences of scores between Sce1 and Sce2, and also between Sce3 and Sce4

Unfortunately, the impact of the dependencies on the scores of the assets is almost imperceptible due to two main reasons.

The first reason is the fact that the added dependencies were only added on the host level, which reduces the impact on the applications and services, specially in this last one. The second reason is due to the weight given to each variable on the different versions

of the model. As can be seen in Table 5.2, the weight of the dependencies variable is only 0.15 which is much lower than the weight of the vulnerabilities, which is 0.7. This difference leads to a much lower impact of the dependencies, even if those dependencies are considered as high scored ones.

# Chapter 6

# Conclusion and Future Work

In line with the DiSIEM project, this dissertation introduces a novel multi-layer risk assessment model for SIEMs to enhance, diversify, and advance this technology and its importance on an organization.

The model considers three layers: Services, Applications, and Hosts, where the Services' layer is an abstract representation of the actions or functions supported by Applications, and these Applications are supported by Hosts.

We also propose three versions of the model to calculate the risk score: Generic Additive (GA), Modified Additive (MA), and Maximum Score (MS). Each version has a different way to assess the risk taking into account different factors of three main variables: Vulnerabilities, Dependencies, and Incidents.

A tool was developed to assess the risk in an organization. This tool has three different components: Database, application, and a dashboard. The database allows to store all significant information for the application to assess the risk of each asset. After the assessment is completed, the dashboard gathers all assets of the organization, the respective risk score, and dependencies to simplify the work of the security expert. In particular, the tool was developed to be integrated in an industrial environment, in concrete, the EDP SOC. In order to visualize the assessment, we developed a dashboard.

The different model versions were analyzed and compared between each other based on four different scenarios. These scenarios had their differences based on the number of vulnerabilities and the number of dependencies between assets, creating a common ground to understand what would happen to the score when the number of vulnerabilities and the number of dependencies increases or decreases.

Nevertheless, the model has potential to become more sophisticated, precise and effective to assess risk. In order to do so, several changes have to be made as future work. An aspect to consider is the necessary to perform a deeper study on the influence of the Incident variable. As mentioned during the dissertation, this variable is divided in two factors: Current Month and the Previous Three Months. We realized in the beginning of each month after the update of the previous month's scores, the score of the current

month is zero due to the fact that there are no incidents. Currently, this particular period decreases the total score of the Incident Variable drastically and it might be too extreme. Another consideration in the Incident Variable is to have another factor to weight differently the closed and the opened incidents. We have concluded, intuitively, that an incident might have a higher impact on the organization when the respective incident is not treated and properly studied comparing to an incident that was already closed and concluded.

In order to extend our comprehension of the assets belonging to the Host layer, we will analyze the possibility of having a Patched factor. This factor would have the objective to indicate if the version of the operating system of a host is the lasted version, which might lead to a safer environment.

We are also thinking in a novel way to integrate the assessment of the risk of collaborators to this model. During this dissertation, we realized that the majority of the incidents occur with the collaborators and their computers, and unfortunately, that type of incident is not considered in this model yet. We consider that with the assessment of the collaborators' risk, the risk of the organization would be more precise and realistic. Another direction for future work is to evaluate the usability of the tool and its adequacy to other real contexts.

With this model and tool, a SIEM solution can have its risk evaluation process enhanced once the impact of an incident can now be measured at a higher levels of decision making (e.g., applications or services), instead of only be measured on the asset that occurred, in a too operational level of decision making.

# Bibliography


[1] DiSIEM. Disiem project. http://disiem-project.eu. Accessed 26/06/2017.

[2] Faculdade de Ciências da Universidade de Lisboa. Fcul website. https://ciencias.ulisboa.pt/en. Accessed 26/06/2017.

[3] Energias de Portugal. Edp website. http://www.edp.pt/en/Pages/homepage.aspx. Accessed 26/06/2017.

[4] ISO entity. *ISO 27005 - Information Security Risk Management*. ISO entity, 2011.

[5] NIST entity. *NIST Special Publication 800-30 - Guide for Conducting Risk Assessments*. NIST entity, 2012.

[6] Michael E. Whitman and Herbert J. Mattord. *Management of Information Security, 4th Edition*. Cengage Learning, 2014.

[7] Michael E Whitman, Herbert J Mattord, and Andrew Green. *Principles of incident response and disaster recovery*. Cengage Learning, 2013.

[8] International Organization for Standardization. International Organization for Standardization. http://www.iso.org/iso/home.html. Accessed 26/06/2017.

[9] ISO entity. *ISO 31000 - Risk management*. ISO entity, 2013.

[10] International Organization for Standardization. International Organization for Standardization. https://www.iso.org/isoiec-27001-information-security.html. Accessed 26/06/2017.

[11] International Electrotechnical Commission. International Electrotechnical Commission. http://www.iec.ch. Accessed 26/06/2017.

[12] AlienVault LC. *AlienVault Unified SIEM - System description*. AlienVault LC, 2010.

[13] Hewlett-Packard Development Company. ArcSight Website. https://software.microfocus.com/en-us/software/siem-security-information-event-management. Accessed 05/10/2017.

[14] IBM. IBM Security QRadar SIEM. http://www-03.ibm.com/software/products/en/qradar-siem. Accessed 24/07/2017.

[15] AlienVault LC. AlienVault Open Source Security Information and Event Management. https://www.alienvault.com/products/ossim. Accessed 24/07/2017.

[16] AlienVault LC. AlienVault Unified Security Management. https://www.alienvault.com/products. Accessed 24/07/2017.

[17] AlienVault LC. *AlienVault Users Manual*. AlienVault LC, 2010.

[18] Magic Quadrant. *Magic Quadrant for Security Information and Event Management*. Magic Quadrant, 2016.

[19] AlienVault LC. *Netflow Collection with AlienVault*. AlienVault LC, 2013.

[20] Hewlett-Packard Company. Hewlett-Packard Company. http://www.hp.com. Accessed 26/06/2017.

[21] IBM. IBM Security QRadar QFlow Collector. http://www-03.ibm.com/software/products/en/qradar-qflow-collector. Accessed 24/07/2017.

[22] IBM. IBM Security QRadar VFlow Collector. http://www-03.ibm.com/software/products/en/qradar-vflow-collector. Accessed 24/07/2017.

[23] IBM. IBM Security QRadar Log Manager. http://www-03.ibm.com/software/products/en/qradar-log-manager. Accessed 24/07/2017.

[24] IBM. IBM Security QRadar Risk Manager. http://www-03.ibm.com/software/products/en/qradar-risk-manager. Accessed 24/07/2017.

[25] IBM. IBM Security X-Force Threat Intelligence. http://www-03.ibm.com/software/products/en/x-force-threat-intelligence. Accessed 24/07/2017.

[26] IBM. IBM QRadar User Behavior Analytics. http://www-03.ibm.com/software/products/en/qradar-user-behavior-analytics. Accessed 24/07/2017.

[27] Hewlett-Packard Development Company. Priority Formula. http://h41382.www4.hpe.com/gfs-shared/downloads-340.pdf. Accessed 26/06/2017.

[28] Hewlett-Packard Development Company. Priority Formula. https://h41382.www4.hpe.com/gfs-shared/downloads-304.pdf. Accessed 26/06/2017.

[29] IBM. IBM - IBM QRadar Security Intelligence Platform. http://www-03.ibm.com/software/products/pt/qradar. Accessed 26/06/2017.

[30] IBM. IBM Security zSecure Adapters for QRadar SIEM. http://www-03.ibm.com/software/products/en/zsecure-adapters-qradar-siem. Accessed 24/07/2017.

[31] IBM. IBM Security QRadar Vulnerability Manager. http://www-03.ibm.com/software/products/en/qradar-vulnerability-manager. Accessed 24/07/2017.

[32] IBM. IBM Security QRadar Incident Forensics. http://www-03.ibm.com/software/products/en/qradar-incident-forensics. Accessed 24/07/2017.

[33] IBM. Asset risk levels and vulnerability categories. http://www.ibm.com/support/knowledgecenter/en/SS42VS_7.2.6/com.ibm.qradar.doc/c_qvm_view_scan_rslthosts_.html. Accessed 24/07/2017.

[34] IBM. Scan Investigation on Vulnerability Manager. http://www.ibm.com/support/knowledgecenter/SS42VS_7.2.6/com.ibm.qradar.doc/c_qvm_scan_invest.html. Accessed 24/07/2017.

[35] IBM. IBM Security Scan results. https://www.ibm.com/developerworks/community/wikis/home ?lang=zh#!/wiki/W746177d414b9_4c5f_9095_5b8657ff8e9d/page/Bigfix%20&%20Qradar%20 Vulnerability%20Manager%20Security%20Software%20Integration. Accessed 28/11/2016.

[36] Stefan Strecker, David Heise, and Ulrich Frank. Riskm: A multi-perspective modeling method for it risk assessment. *Information Systems Frontiers*, 13(4):595–611, 2011.

[37] Tak-wah Kwan. *A risk management methodology with risk dependencies*. The Hong Kong Polytechnic University, 2010. Accessed 26/06/2017.

[38] Forum of Incident Response and Security Teams (FIRST). *Common Vulnerability Scoring System v3.0: Specification Document*. Forum of Incident Response and Security Teams (FIRST), 2015.

[39] Forum of Incident Response and Security Teams (FIRST). Forum of Incident Response and Security Teams (FIRST). https://www.first.org/about. Accessed 26/06/2017.

[40] Forum of Incident Response and Security Teams (FIRST). Forum of Incident Response and Security Teams (FIRST) CVSS V3 Specification Document. https://www.first.org/cvss/specification-document. Accessed 24/07/2017.

[41] Unified Modeling Language. UML - Unified Modeling Language. http://www.uml.org. Accessed 26/06/2017.

[42] Randomness and Integrity Services Lda. Random - True Random Number Service. https://www.random.org. Accessed 26/06/2017.

# .1 Dashboards

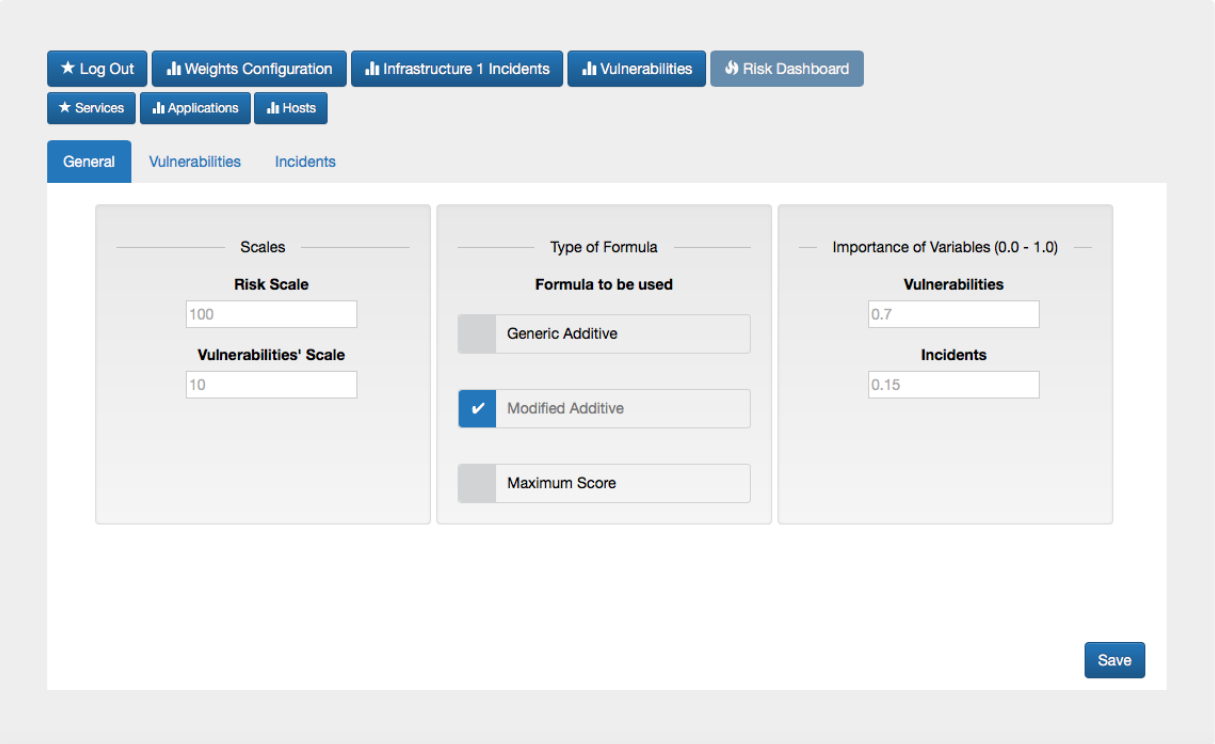



Figure 1: General Weights Configuration Dashboard Page

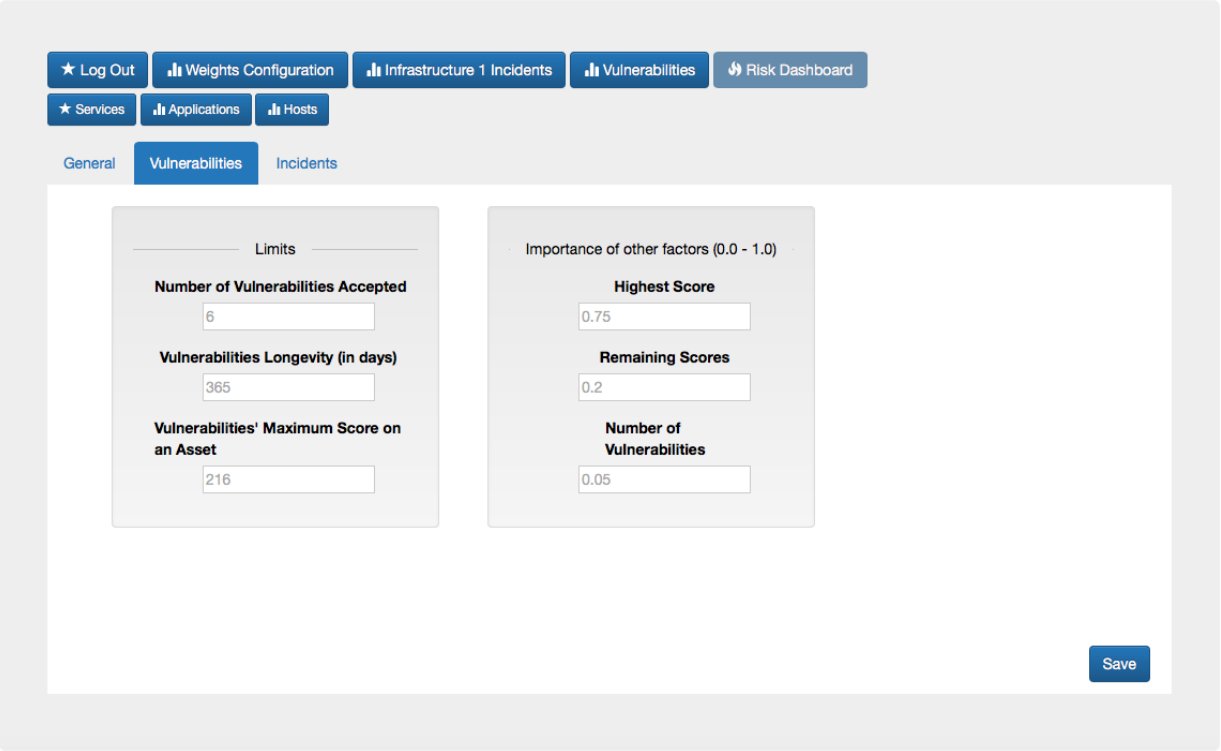



Figure 2: Vulnerabilities Weights Configuration Dashboard Page

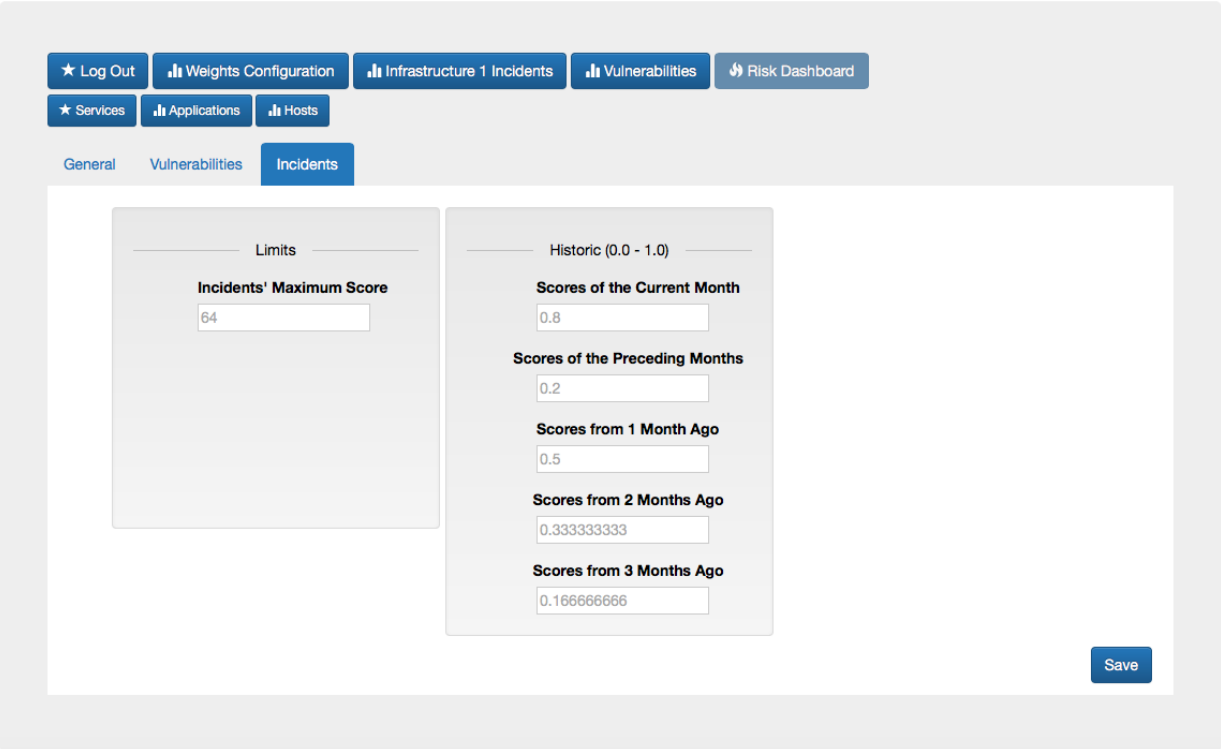



Figure 3: Incidents Weights Configuration Dashboard Page

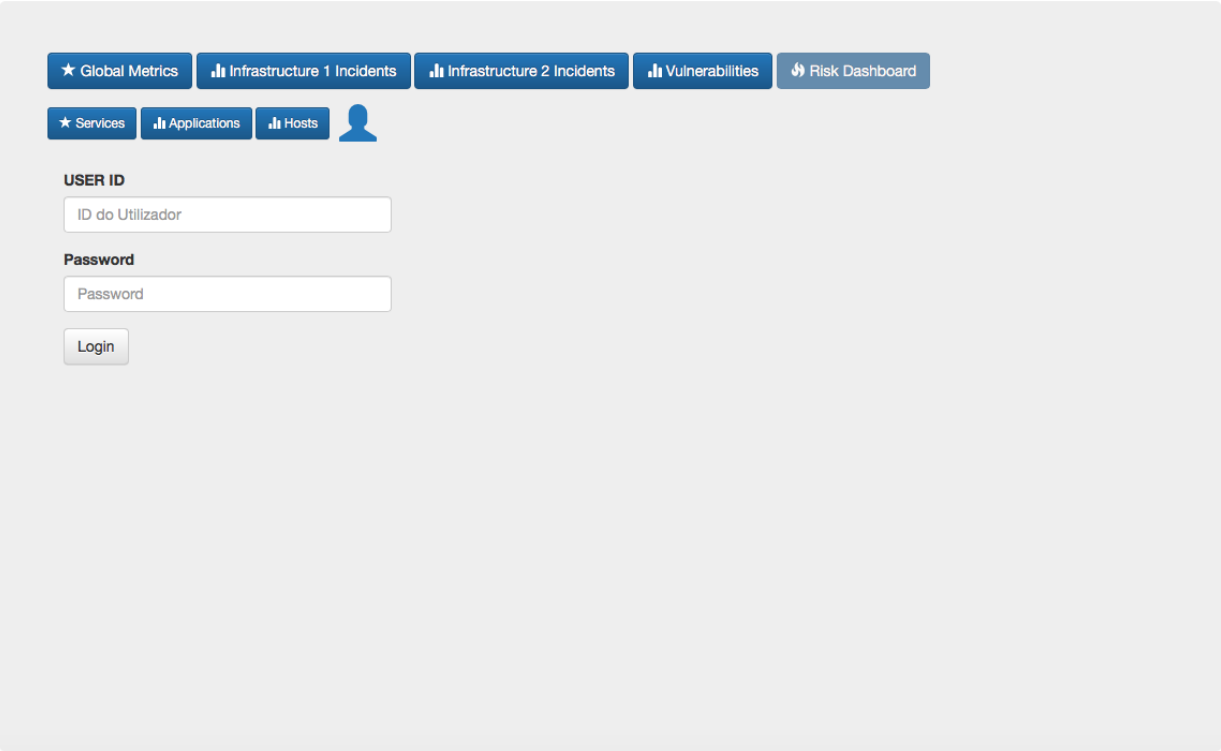



Figure 4: Login Dashboard Page

# .2 Structure of Scenario 1

**List of Dependencies:**

Services

S1 – A1, A2, A3, A4, A5, A6, A7, A8, A9, A10

S2 – A11, A12, A13, A14, A15, A16, A17, A18, A19, A20

S3 – A21, A22, A23, A24, A25, A26, A27, A28, A29, A30

S4 – A31, A32, A33, A34, A35, A36, A37, A38, A39, A40

S5 – A41, A42, A43, A44, A45, A46, A47, A48, A49, A50

Applications

A1 – M28, M32, M18, A20

A2 – M12, M35, M9

A3 – M33, M13, M19, M49, A7, A29

A4 – M28

A5 – M19

A6 – M41, M48

A7 – M29, M15, M47, A45, A40

A8 – M14, M39, M41, M38

A9 – M7, M46, M48, M30, A44

A10 – M31, M50, M14, M18, A4

A11 – M40, M5, M36, M39
A12 – M22, M42, A21, A21, A46
A13 – M31, M5, M49, M7, A32
A14 – M35, A20
A15 – M6, M37, M9, M4
A16 – M37, A13
A17 – M22
A18 – M43, M11, M14, M21
A19 – M20, M23, M26
A20 – M19, M34, M24, M18
A21 – M11, M6, M28, M2, A12, A39
A22 – M4
A23 – M7, M13
A24 – M50, M34, M5, M27, M32, A16, A30
A25 – M50, M15, M12, M33
A26 – M1
A27 – M3, M17, M18, M24, M34, A38
A28 – M10, M14
A29 – M1, M8, M21
A30 – M10, A23
A31 – M3, M6, M46, M45
A32 – M5
A33 – M14
A34 – M16, M30, M32, M34
A35 – M10, M35, M36, M43, M45
A36 – M9, M12, M24, M30
A37 – M12, M24, M37, M47, M27
A38 – M8, M29, M41, M40, M6
A39 – M1, M20, M4, M33
A40 – M8, M13, M27
A41 – M4, M11, A13, A11
A42 – M1
A43 – M4, M25, M29, M33, M44, A13, A9
A44 – M3, M13, M31, M35, M45
A45 – M7, M10
A46 – M6, A18, A22
A47 – M1, M3
A48 – M2
A49 – M22, A10

A50 – M1

Machines:

There are no dependencies between machines in this scenario.

**List of Vulnerabilities:**

M10 – 1 [8]

M30 – 2 [4, 9]

M38 – 1 [8*]

M23 – 1 [8]

M27 – 1 [4]

M44 – 2 [6.5, 4]

M7 – 1 [4]

M37 – 1 [8*]

M43 – 1 [9]

M12 – 1 [6.5]

M28 – 1 [8]

M40 – 1 [8*]

M19 – 1 [4]

M29 – 1 [9]

M41 – 1 [6.5]

M35 – 1 [6.5]

M34 – 2 [8, 4*]

A9 – 1 [6.5]

A35 – 1 [6.5]

A17 – 2 [4, 4]

A16 – 1 [8]

A1 – 1 [8]

A8 – 1 [9]

A41 – 1 [4]

A23 – 1 [6.5*]

A26 – 1 [8]

A14 – 1 [8]

A34 – 1 [6.5]

A2 – 1 [8]

A33 – 1 [6.5*]

A27 – 1 [6.5]

A5 – 1 [9]

A7 – 1 [8]

A25 – 1 [4*]

A37 – 1 [4]

A32 – 1 [8*]

In the List of Vulnerabilities, first it is presented the number of vulnerabilities that an asset has followed by the scores of each vulnerability contained in an array. Every time the symbol '*' is presented, it means that vulnerability has its Scope $Changed$ meaning that can jeopardize other assets.

**List of Incidents:**

A23 - 2 [(3, 1, 3); (3, 3, 1)]

A33 - 1 [(3, 2, 2)]

A7 - 1 [(2, 3, 1)]

M36 - 1 [(4, 3, 3)]

In the List of Incidents, first is present the number of incidents that an asset has followed by the scores of each incident contained in an array. Each incident has three factors, the Operational Impact, Consequence Severity, and Security Classification, which are represented as follow: (Operational Impact Value, Consequence Severity Value, Security Classification Value).

# .3 Structure of Scenario 2

**List of Dependencies:**

Services

S1 – A1, A2, A3, A4, A5, A6, A7, A8, A9, A10

S2 – A11, A12, A13, A14, A15, A16, A17, A18, A19, A20

S3 – A21, A22, A23, A24, A25, A26, A27, A28, A29, A30

S4 – A31, A32, A33, A34, A35, A36, A37, A38, A39, A40

S5 – A41, A42, A43, A44, A45, A46, A47, A48, A49, A50

Applications

A1 – M28, M32, M18, A20

A2 – M12, M35, M9

A3 – M33, M13, M19, M49, A7, A29

A4 – M28

A5 – M19

A6 – M41, M48

A7 – M29, M15, M47, A45, A40

A8 – M14, M39, M41, M38

A9 – M7, M46, M48, M30, A44

A10 – M31, M50, M14, M18, A4

A11 – M40, M5, M36, M39

A12 – M22, M42, A21, A21, A46
A13 – M31, M5, M49, M7, A32
A14 – M35, A20
A15 – M6, M37, M9, M4
A16 – M37, A13
A17 – M22
A18 – M43, M11, M14, M21
A19 – M20, M23, M26
A20 – M19, M34, M24, M18
A21 – M11, M6, M28, M2, A12, A39
A22 – M4
A23 – M7, M13
A24 – M50, M34, M5, M27, M32, A16, A30
A25 – M50, M15, M12, M33
A26 – M1
A27 – M3, M17, M18, M24, M34, A38
A28 – M10, M14
A29 – M1, M8, M21
A30 – M10, A23
A31 – M3, M6, M46, M45
A32 – M5
A33 – M14
A34 – M16, M30, M32, M34
A35 – M10, M35, M36, M43, M45
A36 – M9, M12, M24, M30
A37 – M12, M24, M37, M47, M27
A38 – M8, M29, M41, M40, M6
A39 – M1, M20, M4, M33
A40 – M8, M13, M27
A41 – M4, M11, A13, A11
A42 – M1
A43 – M4, M25, M29, M33, M44, A13, A9
A44 – M3, M13, M31, M35, M45
A45 – M7, M10
A46 – M6, A18, A22
A47 – M1, M3
A48 – M2
A49 – M22, A10
A50 – M1

Machines:
M10 – M3, M30, M36
M17 – M3
M30 – M27
M36 – M24
M4 – M14, M48
M40 – M38, M47, M48
M43 – M25
M45 – M13
M46 – M11
M50 – M3, M47
M6 – M15, M41

**List of Vulnerabilities:**

M10 – 1 [8]
M30 – 2 [4, 9]
M38 – 1 [8*]
M23 – 1 [8]
M27 – 1 [4]
M44 – 2 [6.5, 4]
M7 – 1 [4]
M37 – 1 [8*]
M43 – 1 [9]
M12 – 1 [6.5]
M28 – 1 [8]
M40 – 1 [8*]
M19 – 1 [4]
M29 – 1 [9]
M41 – 1 [6.5]
M35 – 1 [6.5]
M34 – 2 [8, 4*]
A9 – 1 [6.5]
A35 – 1 [6.5]
A17 – 2 [4, 4]
A16 – 1 [8]
A1 – 1 [8]
A8 – 1 [9]
A41 – 1 [4]
A23 – 1 [6.5*]

A26 – 1 [8]
A14 – 1 [8]
A34 – 1 [6.5]
A2 – 1 [8]
A33 – 1 [6.5*]
A27 – 1 [6.5]
A5 – 1 [9]
A7 – 1 [8]
A25 – 1 [4*]
A37 – 1 [4]
A32 – 1 [8*]

In the List of Vulnerabilities, first it is presented the number of vulnerabilities that an asset has followed by the scores of each vulnerability contained in an array. Every time the symbol '*' is presented, it means that vulnerability has its Scope $Changed$ meaning that can jeopardize other assets.

**List of Incidents:**

A23 - 2 [(3, 1, 3); (3, 3, 1)]
A33 - 1 [(3, 2, 2)]
A7 - 1 [(2, 3, 1)]
M36 - 1 [(4, 3, 3)]

In the List of Incidents, first is present the number of incidents that an asset has followed by the scores of each incident contained in an array. Each incident has three factors, the Operational Impact, Consequence Severity, and Security Classification, which are represented as follow: (Operational Impact Value, Consequence Severity Value, Security Classification Value).

# .4 Structure of the Scenario 3

**List of Dependencies:**

Services

S1 – A1, A2, A3, A4, A5, A6, A7, A8, A9, A10

S2 – A11, A12, A13, A14, A15, A16, A17, A18, A19, A20

S3 – A21, A22, A23, A24, A25, A26, A27, A28, A29, A30

S4 – A31, A32, A33, A34, A35, A36, A37, A38, A39, A40

S5 – A41, A42, A43, A44, A45, A46, A47, A48, A49, A50

Applications

A1 – M28, M32, M18, A20

A2 – M12, M35, M9

A3 – M33, M13, M19, M49, A7, A29

A4 – M28

A5 – M19

A6 – M41, M48

A7 – M29, M15, M47, A45, A40

A8 – M14, M39, M41, M38

A9 – M7, M46, M48, M30, A44

A10 – M31, M50, M14, M18, A4

A11 – M40, M5, M36, M39

A12 – M22, M42, A21, A21, A46

A13 – M31, M5, M49, M7, A32

A14 – M35, A20

A15 – M6, M37, M9, M4

A16 – M37, A13

A17 – M22

A18 – M43, M11, M14, M21

A19 – M20, M23, M26

A20 – M19, M34, M24, M18

A21 – M11, M6, M28, M2, A12, A39

A22 – M4

A23 – M7, M13

A24 – M50, M34, M5, M27, M32, A16, A30

A25 – M50, M15, M12, M33

A26 – M1

A27 – M3, M17, M18, M24, M34, A38

A28 – M10, M14

A29 – M1, M8, M21

A30 – M10, A23

A31 – M3, M6, M46, M45
A32 – M5
A33 – M14
A34 – M16, M30, M32, M34
A35 – M10, M35, M36, M43, M45
A36 – M9, M12, M24, M30
A37 – M12, M24, M37, M47, M27
A38 – M8, M29, M41, M40, M6
A39 – M1, M20, M4, M33
A40 – M8, M13, M27
A41 – M4, M11, A13, A11
A42 – M1
A43 – M4, M25, M29, M33, M44, A13, A9
A44 – M3, M13, M31, M35, M45
A45 – M7, M10
A46 – M6, A18, A22
A47 – M1, M3
A48 – M2
A49 – M22, A10
A50 – M1
Machines:
There are no dependencies between machines in this scenario.

**List of Vulnerabilities:**

A1 - 3 [9, 8*, 4]
A14 - 4 [6.5, 6.5, 4, 8]
A16 - 2 [8, 6.5]
A17 - 3 [4, 4*, 4];
A2 - 1 [6.5*]
A23 - 3 [6.5, 9, 6.5*]
A25 - 1 [4*]
A26 - 6 [4, 4, 4*, 4, 6.5, 8]
A27 - 4 [4, 4, 6.5, 6.5]
A32 - 1 [8]
A33 - 5 [4, 8, 8, 6.5, 4]
A34 - 5 [4, 6.5, 4, 8, 4]
A35 - 6 [6.5,6.5,4,8*,6.5,4]
A37 - 1 [9]
A41 - 4 [9, 8, 4, 4]

A5 - 6 [4, 8, 6.5*, 8, 8, 6.5*]
A7 - 4 [8, 8, 8*, 8]
A8 - 1 [4*]
A9 - 6 [4, 6.5, 8, 6.5, 8*, 4]
M10 - 1 [8]
M12 - 3 [9, 4, 4]
M19 - 6 [4, 6.5, 9, 4, 4*, 6.5*]
M23 - 2 [4, 8]
M27 - 2 [9, 4]
M28 - 2 [6.5, 9*]
M29 - 3 [4*, 4, 4]
M30 - 1 [6.5*]
M34 - 1 [4]
M37 - 2 [8, 9]
M38 - 5 [6.5*, 6.5, 4, 4*, 8]
M40 - 1 [4]
M41 - 4 [4, 4*, 9, 6.5*]
M43 - 5 [6.5*, 6.5*, 8, 4, 4]
M44 - 3 [9, 9, 9]
M7 - 1 [4*]

In the List of Vulnerabilities, first it is presented the number of vulnerabilities that an asset has followed by the scores of each vulnerability contained in an array. Every time the symbol '*' is presented, it means that vulnerability has its Scope $Changed$ meaning that can jeopardize other assets.

**List of Incidents:**

A23 - 2 [(3, 1, 3); (3, 3, 1)]
A33 - 1 [(3, 2, 2)]
A7 - 1 [(2, 3, 1)]
M36 - 1 [(4, 3, 3)]

In the List of Incidents, first is present the number of incidents that an asset has followed by the scores of each incident contained in an array. Each incident has three factors, the Operational Impact, Consequence Severity, and Security Classification, which are represented as follow: (Operational Impact Value, Consequence Severity Value, Security Classification Value).

## .5 Structure of the Scenario 4

**List of Dependencies:**

Services

S1 – A1, A2, A3, A4, A5, A6, A7, A8, A9, A10

S2 – A11, A12, A13, A14, A15, A16, A17, A18, A19, A20

S3 – A21, A22, A23, A24, A25, A26, A27, A28, A29, A30

S4 – A31, A32, A33, A34, A35, A36, A37, A38, A39, A40

S5 – A41, A42, A43, A44, A45, A46, A47, A48, A49, A50

Applications

A1 – M28, M32, M18, A20

A2 – M12, M35, M9

A3 – M33, M13, M19, M49, A7, A29

A4 – M28

A5 – M19

A6 – M41, M48

A7 – M29, M15, M47, A45, A40

A8 – M14, M39, M41, M38

A9 – M7, M46, M48, M30, A44

A10 – M31, M50, M14, M18, A4

A11 – M40, M5, M36, M39

A12 – M22, M42, A21, A21, A46

A13 – M31, M5, M49, M7, A32

A14 – M35, A20

A15 – M6, M37, M9, M4

A16 – M37, A13

A17 – M22

A18 – M43, M11, M14, M21

A19 – M20, M23, M26

A20 – M19, M34, M24, M18

A21 – M11, M6, M28, M2, A12, A39

A22 – M4

A23 – M7, M13

A24 – M50, M34, M5, M27, M32, A16, A30

A25 – M50, M15, M12, M33

A26 – M1

A27 – M3, M17, M18, M24, M34, A38

A28 – M10, M14

A29 – M1, M8, M21

A30 – M10, A23

A31 – M3, M6, M46, M45
A32 – M5
A33 – M14
A34 – M16, M30, M32, M34
A35 – M10, M35, M36, M43, M45
A36 – M9, M12, M24, M30
A37 – M12, M24, M37, M47, M27
A38 – M8, M29, M41, M40, M6
A39 – M1, M20, M4, M33
A40 – M8, M13, M27
A41 – M4, M11, A13, A11
A42 – M1
A43 – M4, M25, M29, M33, M44, A13, A9
A44 – M3, M13, M31, M35, M45
A45 – M7, M10
A46 – M6, A18, A22
A47 – M1, M3
A48 – M2
A49 – M22, A10
A50 – M1
Machines:
M10 – M3, M30, M36
M17 – M3
M30 – M27
M36 – M24
M4 – M14, M48
M40 – M38, M47, M48
M43 – M25
M45 – M13
M46 – M11
M50 – M3, M47
M6 – M15, M41

**List of Vulnerabilities:**

A1 - 3 [9, 8*, 4]
A14 - 4 [6.5, 6.5, 4, 8]
A16 - 2 [8, 6.5*]
A17 - 3 [4, 4*, 4];
A2 - 1 [6.5*]

A23 - 3 [6.5, 9, 6.5*]
A25 - 1 [4*]
A26 - 6 [4, 4, 4*, 4, 6.5, 8]
A27 - 4 [4, 4, 6.5, 6.5]
A32 - 1 [8]
A33 - 5 [4, 8, 8, 6.5, 4]
A34 - 5 [4, 6.5, 4, 8, 4]
A35 - 6 [6.5,6.5,4,8*,6.5,4]
A37 - 1 [9]
A41 - 4 [9, 8, 4, 4]
A5 - 6 [4, 8, 6.5*, 8, 8, 6.5*]
A7 - 4 [8, 8, 8*, 8]
A8 - 1 [4*]
A9 - 6 [4, 6.5, 8, 6.5, 8*, 4]
M10 - 1 [8]
M12 - 3 [9, 4, 4]
M19 - 6 [4, 6.5, 9, 4, 4*, 6.5*]
M23 - 2 [4, 8]
M27 - 2 [9, 4]
M28 - 2 [6.5, 9*]
M29 - 3 [4*, 4, 4]
M30 - 1 [6.5*]
M34 - 1 [4]
M37 - 2 [8, 9]
M38 - 5 [6.5*, 6.5, 4, 4*, 8]
M40 - 1 [4]
M41 - 4 [4, 4*, 9, 6.5]
M43 - 5 [6.5*, 6.5, 8, 4, 4]
M44 - 3 [9, 9, 9]
M7 - 1 [4*]

In the List of Vulnerabilities, first it is presented the number of vulnerabilities that an asset has followed by the scores of each vulnerability contained in an array. Every time the symbol '*' is presented, it means that vulnerability has its Scope $Changed$ meaning that can jeopardize other assets.

**List of Incidents:**

A23 - 2 [(3, 1, 3); (3, 3, 1)]
A33 - 1 [(3, 2, 2)]
A7 - 1 [(2, 3, 1)]

M36 - 1 [(4, 3, 3)]

In the List of Incidents, first is present the number of incidents that an asset has followed by the scores of each incident contained in an array. Each incident has three factors, the Operational Impact, Consequence Severity, and Security Classification, which are represented as follow: (Operational Impact Value, Consequence Severity Value, Security Classification Value).

# .6 Application 26 Assessment Example

**Assessment of the application A26 using the Generic Additive Formula**

The assessment of the Asset: A26 has began.

This asset has also a value of: Diamond

Initiating the Vulnerability Variable scoring process...

This asset has 1 vulnerabilities

Vulnerability Nº1 has a score of 8.0 and it was opened 256 days ago leading to a score of:54.443835616438356

The assessment of the vulnerabilities has finished.

Currently, the Vulnerability Variable's Score is: 54.443835616438356

After comparing and/or truncation process, the final Vulnerability Variable's Score is: 25.205479452054792

Initiating the Dependency Variable scoring process...

List of Dependencies:

Asset: M1 with a score of: 0.0. Currently summed: 0.0

There are 1 dependencies, which leads to a score of: 0.0

This Asset has 0 incidents...

Total Score of the asset is: 17.643835616438352, where Vuln Var: 25.205479452054792, Dep Var: 0.0, and Inc Var: 0.0

**Assessment of the application A26 using the Modified Additive Formula**

The assessment of the Asset: A26 has began.

This asset has also a value of: Diamond

Initiating the Vulnerability Variable scoring process...

This asset has 1 vulnerabilities

Days converted is: 0.7041095890410959

Vulnerability Nº1 has a score of 8.0 and it was opened 257 days ago leading to a score of:54.531506849315065

SAVBHO is now: 54.531506849315065

Final SAVBHO is: 54.531506849315065 and the highest Score is: 54.531506849315065

SAVBHO converted based on maximum Score accepted on an asset: 0.0

Number of Vulns converted based on maximum limit accepted: 16.666666666666668

Highest Vulnerability Score after converting based on the maximum Score accepted: 68.16438356164383

After comparing and/or truncation process, the final Vulnerability Variable's Score is: 51.95662100456621

Initiating the Dependency Variable scoring process...

Total Value of the dependencies is: 4

List of Dependencies:

Asset: M1 with a score of: 0.0 leading to a score of (currently): 0.0

Final Dependency Variable Score is: 0.0

Asset: A26 Incidents

This Asset has 0 incidents...

Total Score of the asset is: 36.36963470319635, where Vuln Var: 51.95662100456621, Dep Var: 0.0, and Inc Var: 0.0

**Assessment of the application A26 using the Modified Additive Formula**

The assessment of the Asset: A26 has began.

This asset has also a value of: Diamond

Initiating the Vulnerability Variable scoring process...

This asset has 1 vulnerabilities

Days converted is: 0.7041095890410959

Before criticality: 0.7041095890410959

After criticality: 0.7041095890410959

Vulnerability N$^{o}$0 has a score of 8.0 and it was opened 257 days ago leading to a score of:54.531506849315065

The assessment of the vulnerabilities has finished. Currently, the Vulnerability Variable's Score is: 54.531506849315065

After comparing and/or truncation process, the final Vulnerability Variable's Score is: 68.16438356164383

Initiating the Dependency Variable scoring process...

This Asset has 0 incidents...

Total Score of the asset is: 47.71506849315068, where Vuln Var: 68.16438356164383, Dep Var: 0.0, and Inc Var: 0.0