Enterprise Management and Software Risk Prediction Based On Security Metrics

Aristeidis Chatzipoulidis

A dissertation submitted for the degree of
Doctor of Philosophy

University of Macedonia, Department of Applied Informatics

June, 2015

This Dissertation Advisory Committee (DAC) consists of the following members (listed alphabetically):

Professor Eugenia Alexandropoulou
(Dept. of Applied Informatics, University of Macedonia, Greece.)

Associate Professor Ioannis Mavridis (PhD Supervisor)
(Dept. of Applied Informatics, University of Macedonia, Greece.)

Professor Theodoros Kargidis
(Dept. of Marketing, Alexander Technological Educational Institute of Thessaloniki, Greece.)

I would like to dedicate this dissertation to my family

Acknowledgements

Towards the completion of this research journey, there are a number of persons that I would like to express my gratitude. First and foremost, I would like to thank my supervisor, Associate Prof. Ioannis Mavridis, for his support and mentorship throughout the completion of the dissertation. His exceptional insight and generous patience is sincerely appreciated.

I would like to thank Prof. Eugenia Alexandropoulou at the Department of Applied Informatics of University of Macedonia and Prof. Theodoros Kargidis at the Department of Marketing at Alexander Technological Educational Institute of Thessaloniki for reviewing my work and for their constructive feedback and guidance on legal and business issues respectively.

I would like to thank my colleague and friend, Dr. Dimitrios Michalopoulos at the Department of Applied Informatics of University of Macedonia for his valuable support during this journey.

I would like to thank Prof. Athanasios Belidis, for encouraging me to pursue a doctoral path and supporting me throughout the completion of my research.

Last but not least, my family deserves all credit for providing me a shelter of comfort and encouragement at all circumstances.

Abstract

In modern times, enterprises are required to thrive in a highly complex and challenging landscape. Particularly, challenges such as embracing good corporate governance practices to confront financial scandals and minimize information asymmetry among stakeholders, the diversity of enterprise risks and regulatory requirements, put pressure on enterprise management to come up with holistic solutions. In this respect, the Governance, Risk and Compliance (GRC) concept has emerged as a common framework which encompasses into a single and unified approach solutions towards enterprise challenges. Despite research progress on GRC, this concept remains premature upon implementation and lacks organizational involvement. In this dissertation, we decompose GRC into its components and describe essentials, approaches and literature review for each component. Moreover, inspired by the fact that one can manage only what can be measured, we research on security metrics, as means to provide automated and isomorphic management of the GRC content. Particularly, specifications from the Security Content Automation Protocol (SCAP) and similar, identified as SCAP-like, are presented. The aim is to support homogeneous data analysis including identification, measurement, reporting and compliance checking. In addition, since software vulnerabilities are regarded as one of the key reasons for information infrastructure risks and malfunctions, the focus is to prioritize software vulnerabilities depending on their severity for corrective actions. In this respect, in the context of security metrics we place emphasis on vulnerability scoring methods in terms of qualitative, quantitative and hybrid methods by analyzing characteristics and indicating strengths and weaknesses. Considering the growing requirements for mature software that can be trusted during application, we elaborate on the context of software trust. Particularly, we describe software development models and emphasize on the Software Maturity Assurance Model (SAMM) as an open guide to improve software maturity because the higher the level of software maturity the higher the quality of the organization's process. Moreover, we present software quality standards that can be used to develop and refine software practices. In addition, based on empirical evidence, we examine the behaviour of software versions based on vulnerability analysis to test when each version is considered mature enough, thereby causing the least possible defects, so as to evoke trust among users. Last but not least, unforeseen risk is always a reality hard to deal with and getting passive measures after risk occurrence is no longer an acceptable explanation when enterprise assets, and shareholder value is at stake. Therefore to avoid unpleasant surprises, enterprises should be at the forefront of predicting and mitigating the consequences of future risks. Taking for granted that vulnerabilities will inevitably continue to be discovered after the software is released and since there is no guarantee that software will be vulnerability-free during its life-cycle, it appears compelling to improve on vulnerability and risk prediction studies. In this respect, we develop a novel risk prediction methodology on the basis of vulnerability prediction using standardized security metrics. The aim is to proactively anticipate vulnerability, threat and risk trends in a real-time basis.

Contents

Contents 6

List of Figures 8

List of Tables 9

1. Introduction 10
1.2 Motivation 13
1.3 Objectives 14
1.4 Research areas 15
1.4.1 GRC concept 15
1.4.2 Security metrics 16
1.4.3 Software trust 16
1.4.4 Risk prediction 17
1.5 Structure of the dissertation 18
2. GRC concept 19
2.1 Introduction 19
2.2 Governance 19
2.2.1 Governance essentials 19
2.2.2 Literature review 20
2.2.3 Governance approaches 23
2.3 Risk management 26
2.3.1 Risk management essentials 27
2.3.2 Risk management phases 32
2.3.3 Risk management approaches 38
2.3.4 Literature review 46
2.4 Compliance 47
2.4.1 Compliance essentials 47
2.4.2 Compliance function in e-banking 49
2.4.3 Literature review 52
2.5 Audit 54
2.5.1 Introduction 54
2.5.2 Internal audit 54
2.5.3 External audit 56
2.5.5 Literature review 58
2.6 Chapter summary 59
3. Security metrics 60
3.1 Introduction 60
3.2 SCAP specifications 60
3.3 SCAP- like specifications 80
3.4 Vulnerability scoring methods 89

3.4.1 Qualitative vulnerability scoring methods 90
3.4.2 Quantitative vulnerability scoring methods 93
3.4.3 Hybrid vulnerability scoring methods 96
3.5 Chapter summary 97
4. Software trust 98
4.1 Introduction 98
4.2 Software maturity 99
4.3 Software quality standards 102
4.4 Case study 103
4.5 Related work 108
4.6 Chapter summary 110
5. Risk prediction methodology 111
5.1 Introduction 111
5.2 The proposed methodology 112
5.2.1. Platform identification 112
5.2.2. Vulnerability history 113
5.2.3. Vulnerability prediction 113
5.2.4 Risk prediction 114
5.3 Implementation example 117
5.3.1. Platform identification 118
5.3.2 Vulnerability history 118
5.3.3. Vulnerability prediction 121
5.3.4. Risk prediction 126
5.4 Related work 127
5.5 Evaluation 130
5.6 Discussion 133
5.7 Chapter summary 134
6. Conclusions 136
6.1 Summary of the contributions 136
6.2 Future work 137
6.3 Closing remarks 138

Appendix 139

References 143

List of Figures

Figure 1. Evolution of IT risk management 27
Figure 2. Decomposition of enterprise risk 29
Figure 3. Factors affecting risk scenarios 33
Figure 4. Control life cycle phases 37
Figure 5. CCE example 63
Figure 6. Components of a CPE name 64
Figure 7. CVSS flow chart 65
Figure 8. CVSS v2 metric groups 65
Figure 9. CVSS Temporal metric group 67
Figure 10. CVSS Environmental metric group 68
Figure 11. XCCDF function 69
Figure 12. XCCDF and OVAL interoperability 70
Figure 13. OVAL flow chart 71
Figure 14. Base metric group 72
Figure 15. Temporal metric group 73
Figure 16. Environmental metrics group 74
Figure 17. Temporal metric group characteristics 75
Figure 18. Environmental metric group characteristics 76
Figure 19. Structure of ARF 78
Figure 20. Example of TMSAD 80
Figure 21. CWSS metric groups 83
Figure 22. CWSS metric groups 84
Figure 23. STIX work flow 86
Figure 24. CVSS v1 metric groups 94
Figure 25. OpenSAMM framework 100
Figure 26. 2.2v 105
Figure 27. 2.3/2.4v 106
Figure 28. 2.0v 106
Figure 29. 1.3v 107
Figure 30. Average 107
Figure 31. BBN topology and CPTs (a to e) 114
Figure 32. APA 122
Figure 33. IIS 122
Figure 34. LIN 123
Figure 35. SOL 123
Figure 36. WS 124

List of Tables

Table 1. Comparative evaluation of governance approaches 26
Table 2. High level risk management phases..... 32
Table 3. Risk factors 34
Table 4. Pros and Cons of qualitative analysis 35
Table 5. Pros and Cons of quantitative analysis 35
Table 6. Factors influencing the selection of risk indicators 36
Table 7. Enterprise controls 37
Table 8. Risk management methods comparison 45
Table 9. SCAP evolution. 61
Table 10. CVSS v2 Base metric group 66
Table 11. AI controlled vocabulary 79
Table 12. SCAP-like specifications 81
Table 13. Number of vulnerability occurrences for Apache versions 105
Table 14. Maturing points and interval calculation 108
Table 15. E-banking sector 118
Table 16. Historical rate of vulnerability occurrences per platform 119
Table 17. CVSS base metric group frequency of occurrences for APA 120
Table 18. K-S test results based on distribution fitting..... 125
Table 19. CPTs results 127
Table 20. Related work comparative study..... 129
Table 21. Semester class for APA 131
Table 22. Semester class for IIS 131
Table 23. Semester class for LIN..... 131
Table 24. Semester class for SOL..... 131
Table 25. Semester class for WS 131
Table 26. Application of the two-phase evaluation method per platform 132
Table 27. Accuracy of results 133

"Patience is bitter, but its fruit is sweet"
Jean-Jacques Rousseau

Chapter 1

1. Introduction

Today (and most probably tomorrow), enterprises (from banks and government agencies to SMEs) rely for their existence, operation and profitability on the management of risks. This appears to be undeniable since risks exist in various forms, such as information security, compliance, reputational and outsourcing risks, and can damage the sustainability of an enterprise. If those risks are not identified, assessed and mitigated in a proper and proactive manner then, it is high likely an epidemic type of risk will occur, starting from a single enterprise or business unit and distribute to others causing systemic disruption or damage at a sector level. This implies that holistic solutions are required that will enable management of enterprise activities including governance, risk and compliance issues.

In this respect, we introduce the reader to the Governance, Risk, and Compliance (GRC) concept as a holistic approach to enterprise management. The aim is to identify, measure and monitor enterprise risks on an ongoing basis. However, this may not seem enough since the GRC concept is not a plug-in solution that solves perpetual enterprise challenges. In this respect, having a GRC program that does not achieve business objectives is a value drain tool that can cause market share loss, customer dissatisfaction and increased operating costs. For this reason, audits should be performed regularly to ensure that a GRC program performs and produces results as intended.

Moreover, a GRC program requires procedures that will automate both the implementation and the monitoring of its performance. This implies that automated security metrics are required to support isomorphic data analysis otherwise the application of a GRC program alone is likely to fall short of expectations. In this respect, we present specifications from the Security Content Automation Protocol (SCAP) and similar, identified as SCAP-like specifications. The latter performs a similar role to official SCAP specifications however, at the time of the press of this dissertation they do not belong in the SCAP family.

Both SCAP and SCAP-like specifications allow for security content automation including asset identification, compliance checking and reporting among other activities. Since regulations and legislations guide corporate behaviour and continue to evolve by becoming stricter in nature, the benefits of including automated security metrics allows for real-time evaluation of compliance status, justification of assets and

controls and strategic alignment of the regulatory, business and information security environment. To complement this research on automated security metrics, we present the reader with vulnerability scoring methods to assist in vulnerability severity measurement. Particularly, we present qualitative, quantitative and hybrid vulnerability scoring methods by outlining characteristics, limitations and benefits for each type of method.

Having presented the GRC concept and automated security metrics, we focus on the risk management acronym. Particularly, we consider that risk management is linked to trust because both are closely related in purpose. Specifically, trust improves the confidence on using an asset and risk management improves the protection of this asset. Therefore, considering that information infrastructures are assets that enterprises rely on to conduct business, we research on software trust by examining the behaviour of software versions through vulnerability analysis to indicate when each one is mature enough so it can be trusted. This study is accompanied with literature review on software maturity, quality standards and trust.

On the grounds of risk management, we introduce a novel risk prediction methodology, based on SCAP specifications and vulnerability analysis, for e-banking information infrastructure software platforms. By software platforms, we express a series of products (i.e. software versions of Microsoft IIS 5.0, 6.0, 7.5) that derive from a particular production chain (i.e. Microsoft IIS). The purpose is to develop a proactive and real-time approach to vulnerability, threat and damage trends and at the same time predict risk in security properties, namely confidentiality, integrity and availability.

In the following bullets, we provide easy to understand definitions for the most common terms used throughout this dissertation to allow the reader get acquainted with basic knowledge so as to facilitate a holistic understanding of the GRC concept, security metrics and prediction.

- *Governance* is the set of policies, processes, procedures, laws and behaviour on how information and relationship among stakeholders should be managed (Da Veiga and Eloff 2007; Rastogi and Von Solms, 2006).

- *Risk management* is the set of activities required to direct an enterprise, such as identifying, assessing, mitigating and monitoring risk events that can have a negative impact (Benini and Sicari, 2008; Mathrani and Mathrani 2013).

- *Compliance* is the act of complying with external regulations as means to provide evidence of adherence to the regulatory environment (Asnar and Massacci, 2011).

- *Internal audit* is defined as an independent, objective assurance and consulting activity designed to add value and improve an organization's operations (IIA, 2009).

- *Security metrics* are a series of metrics that enable collection, analysis and measurement of information security (NIST, 2008).

- *Security Content Automation Protocol* is an umbrella term that consists of automated specifications that aim to automatically manage the security content (Waltermire et al., 2011).

- *E-banking* is an umbrella term that describes the delivery of banking products and services via remote access in a technologically advanced manner using a variety of delivery channels from ATMS, and the Internet to mobile banking (Kondabagil, 2007).

- *Vulnerability* is a weakness, bug or flaw within an application, system, device, or service that could lead to a failure in security properties (Schiffman, 2007).

- *Threat* is the malicious act which intends to exploit vulnerability and cause damage in security properties (Koons and Minoli, 2010).

- *Damage* is the impact on security properties, based on qualitative characteristics of vulnerability and expressed in terms of significance (Koons and Minoli, 2010).

- *Impact* is the damage, usually expressed in monetary terms caused by a threat exploiting vulnerability (Koons and Minoli, 2010).

- *Risk* is the product of the probability a threat exploiting vulnerability and the impact (Koons and Minoli, 2010).

- *Confidentiality* is the security property of protecting any private or business data being divulged to a third party without proper authorization (Koons and Minoli, 2010).

- *Integrity* is the security property of protecting any private or business data from modification without proper authorization (Koons and Minoli, 2010).

- *Availability* is the security property of protecting any private or business data from being blocked or prevented from access (Koons and Minoli, 2010).

- *Software trust* is the degree of existing confidence that the software will satisfy individual needs (Amoroso et al., 1991).

- *Risk prediction* is a prognostic approach of assessing risk that facilitates proactive decision making.

In the remainder of this chapter and in section 1.2, we present the motivation for this dissertation and in section 1.3 we describe the research areas. In section 1.4 we outline the objectives and in section 1.5 we present the organization of chapters.

1.2 Motivation

Inspired by the growing need to manage the inherent complexity in modern enterprises which derives from governance risk and compliance requirements, we chose to research on the GRC concept. Particularly, we break GRC concept into its own components and analyze each one with the aim to offer partial or complete solutions to enterprise management and increase the usefulness and implementation of the GRC concept as a whole. Such an integrated approach is missing from existing doctoral studies (Kowalski, 1994; Yngström, 1996; Zuccato, 2005; Bjorck, 2005; Bakari, 2007).

While this concept is not new, having its origins back in early 2000, research is still in infant stages (Racz et al., 2010). The same authors defines GRC as "an integrated, holistic approach to organization-wide governance, risk and compliance ensuring that an organization acts ethically correct and in accordance with its risk appetite, internal policies and external regulations through the alignment of strategy, processes, technology and people, thereby improving efficiency and effectiveness". Considering this definition as broad and general, we emphasize on each acronym separately to establish the GRC purpose.

Implementing the GRC concept in an enterprise environment is not a trivial task and may seem useless if it is not accompanied with security metrics that produce standardized, repeatable and isomorphic results including activities such as asset identification, vulnerability scoring and compliance checking. This motivated us to research on security metrics that fulfil certain requirements such as a) being standardized, b) make their formulas available to the public, c) update on a regular basis, d) comply with best practices and standards and e) allow mix and comparison with other tools and methods. To satisfy this motivation, SCAP specifications and similar are analyzed and utilized. Moreover, within the context of security metrics, we present a state of the art research on vulnerability scoring methods driven by the increasing interest in measuring the severity of vulnerabilities (Huang et al., 2013; Liu et al., 2012; Liu and Zhang, 2011).

On the grounds of risk management and motivated to know when a software platform is at its best so it can trusted and considering that the majority of researches focus on software development before public release (Casey, 2010; Pressman, 2010), this dissertation examines the behaviour of a software platform on the basis of its versions through vulnerability analysis. The reason is to find when software versions are mature enough in order to optimize performance of the software and evoke trust among users.

Finally, driven by the increasing interest in zero-day vulnerabilities and vulnerability prediction studies, we develop a vulnerability prediction approach which acts as a basis for a novel risk prediction methodology. By zero-day vulnerability we refer to a weakness or a hole in software that is unknown and can be exploited before the vendor becomes aware (Bilge and Dumitras, 2012). Considering that the majority of researches focus on vulnerability prediction alone (Alhazmi et al., 2007; Chowdhury and Zulkernine, 2011; Shar and Tan 2013; Shin and Williams, 2013; Venter and

Eloff, 2004; Woo et al., 2011; Zhang et al., 2011), inspired us to introduce the measurement of zero-day risk, a term absent from existing literature, dedicated to the measurement of unknown risk which can be defined as the uncertain loss caused by a damage event and the corresponding impact for each security property, namely confidentiality, integrity and availability.

The practicality of the methodology is demonstrated with an implementation example from the e-banking sector. The reason to research on e-banking sector and supporting information infrastructure platforms is because recent studies on the progress and adoption of e-banking recommend a proactive approach to risk management for e-banking (Aduda and Kingoo, 2012; Aggelis, 2005; Angelakopoulos and Mihiotis, 2011; Gikandi, 2010). This notion is heightened by the increasing number of vulnerability occurrences, the rate of phishing techniques and intelligent attack strategies related to this sector (Khatri and Budhiraja, 2013; Kondabagil, 2007; Osunmuyiwa, 2013; Shah and Clarke, 2009; Shah and Siddiqui, 2006).

Overall, given the range of enterprise complexity and the variety of risks, it is becoming increasingly important to provide holistic solutions to enterprise management. Therefore, to achieve a smooth and complete implementation of the GRC concept, we connect it with automated security metrics in terms of SCAP, SCAP-like and vulnerability scoring methods. Moreover, incentives such as the increasing importance of risk management, the requirements for trusted software products and the lack of risk predictive studies, inspired us to research on software versions' maturity behaviour and develop a new risk prediction methodology for e-banking supporting platforms.

1.3 Objectives

The advent of internet and networks accompanied with increasing regulations has forced a strong dependence on information infrastructures. Although a significant amount of applications and tools exist to cover the GRC concept (see Chapter 2) this variety is proposed as ad-hoc solutions without specific applicability. The same notion exists for security metrics, which fail to link security requirements with business goals and governance/compliance initiatives. This implies the criticality to adapt security metrics that will enable automated security content management accompanied with additional functionalities such as policy compliance checking and reporting.

Therefore, the objectives of our work can be summarized as follows. Firstly, in the context of a holistic approach to enterprise management, we systematically break GRC components into a series of objectives, compare most reputed methods that apply for each component and present the role of audit function within the GRC concept. Secondly, in the context of security metrics, we present SCAP and similar specifications to assist in the implementation of a GRC program. Within the same context, we elaborate on vulnerability scoring methods in order to communicate the characteristics of vulnerabilities and measure their severity.

Thirdly, aiming to optimize software security and minimize risks, we elaborate on software trust. Particularly, we present software development models that aim to improve on software maturity, illustrate which software quality standards can aid in refining software security practices and examine with a case study when software versions are optimized to use. Moreover, in the context of software trust we present an extensive literature review to present the reader with studies related to the effectiveness of achieving software trust. In the context of risk management and considering the importance of managing unforeseen risk, we develop a, first of the kind, risk prediction methodology for information infrastructure protection through vulnerability analysis. The objective is to assist auditors towards justification of controls and decision making towards security trends and future risks.

1.4 Research areas

The contributions of this dissertation are centred on the following topics: a) GRC concept, b) security metrics c) software trust and d) risk prediction. The aforementioned topics are placed in the wider context of a holistic approach to enterprise management and information infrastructure protection. In the remainder of this section, we present the contributions in more details.

1.4.1 GRC concept

The first contribution of this dissertation is the breakdown of GRC into its components. The aim is to highlight the objectives for each GRC component, accompanied approaches and literature reviews in order to discover the benefits and misfits for each component. Hence, in the context of GRC concept, we elaborate on the following topics.

Governance. We address the first component of the GRC concept by describing essentials in terms of governance terminology and objectives. Moreover, we present governance approaches and compare them against specific criteria. In addition, we present an extensive literature review about the application of governance in enterprise environments.

Risk management. For the second GRC component, we present essentials in terms of risk management evolution and decompose enterprise risks into a portfolio of other risks. Moreover, we elaborate on risk management phases to introduce the reader with the characteristics for each phase. Then, we present reputed enterprise IT risk management approaches and compare them against selected criteria. We conclude with an extensive literature review.

Compliance. For the last GRC component, we describe essentials for the compliance function, principles and challenges. In addition, we focus on compliance requirements in the electronic banking sector due to the increased interest and revenues this particular sector generates. A literature review follows which describes how the compliance with regulations affects an enterprises' environment.

Audit. Closely related to the GRC concept is the audit function which may not be as a discrete GRC component however it is designed to measure the performance of a GRC program and add value to the enterprise. For this reason, we analyze the role of internal and external audit and provide a literature review in terms of internal audit.

1.4.2 Security metrics

Another important contribution of this dissertation is within the context of automated security metrics. Particularly, we present and describe SCAP and SCAP-like specifications. Such specifications allow for synergy among enterprises that are (inter)dependent and aim to improve data analysis and communication, compliance checking and reporting. Moreover, we describe vulnerability scoring methods and divide them into types to assist in the selection of the proper method upon measurement of vulnerability severity. Specifically, in the context of security metrics, we elaborate on the following topics.

SCAP specifications. Driven by the requirement to manage the security content in a consistent and isomorphic manner, we present the purpose of adopting automated security metrics, the evolution of SCAP versions and describe the purpose and application of each specification.

SCAP-like specifications. In an attempt to complement the usability of SCAP specifications, we categorize and describe specifications that perform a similar role to SCAP specification but do not belong in the SCAP family as in the time of press of this dissertation.

Vulnerability scoring methods. Considering that vulnerability severity reflects the significance each vulnerability posses to future risk, we present and discuss the most reputed vulnerability scoring methods in terms of qualitative, quantitative and hybrid types.

1.4.3 Software trust

As the demand for quality and usable software products has increased, the need for empirical examination of software trust has grown. Motivated to make software friendly to use and trusted, we present how software maturity is shaped based on

software development models with emphasis given on the Software Maturity Assurance Model (SAMM). Moreover, we present quality standards that aim to refine software practices and conduct a study on software versions' maturity to examine when each version can be trusted. Specifically, in the context of software trust, we elaborate on the following topics.

Software maturity. We present how software maturity is attained under the perspective of software development models given emphasis on the Software Maturity Assurance Model. All models have the same objective to improve on software quality and trust however, this is approached in a different way.

Software quality standards. We present and outline the importance of the software Quality Assurance (SQA) concept and provide a literature review on software quality standards. Moreover, we introduce similar research fields that perform a similar role to SQA.

Software versions' maturity. We identify versions of a software platform and analyze their behavior based on the number of vulnerability occurrences. Particularly, based on a second degree polynomial function, we define three maturity points and two maturity intervals to describe the maturity phases for each version after public release. The aim is to indicate when a software version is mature enough so it can be trusted.

Software trust. An extensive literature review about software trust is presented.

1.4.4 Risk prediction

Considering the importance of managing unforeseen risk, we introduce a novel methodology for the protection of information infrastructure platforms based on utilization of SCAP metrics and a combination of stochastic approaches. Particularly, in the context of risk prediction, we elaborate on the following topics.

Risk prediction methodology. We present a novel risk prediction methodology for software platforms that support e-banking information infrastructures. The methodology consists of four steps and utilizes standardized metrics and stochastic approaches. The contribution is to aid the auditor during decision making in response to potential risks on a real-time basis and in advance of public disclosure, applying appropriate protective actions.

Vulnerability prediction evaluation. We test the accuracy of our vulnerability prediction results which is the core of the proposed risk prediction methodology. To achieve this, we introduce and apply a method that consists of two phases. Based on the evaluation results, the proposed methodology offers increased accuracy in predicting vulnerability trends.

Vulnerability prediction review. We provide a literature review on existing vulnerability prediction studies.

Discussion. We provide benefits and limitations of the proposed risk prediction methodology

1.5 Structure of the dissertation

In this chapter, we introduced the reader to the main topics of this dissertation and provided an overview on basic terms. Moreover, we outlined the motivation, clarified the objectives and illustrated the research areas. The remaining chapters are organized as follows:

Chapter 2 breaks down GRC concept into its components and presents essentials, approaches and literature reviews.

Chapter 3 presents security metrics in terms of SCAP specifications, similar and vulnerability scoring methods.

Chapter 4 discusses software trust from the perspectives of software development models and quality standards. Moreover, a study on software versions' maturity is presented and a literature review on software trust concludes this chapter.

Chapter 5 illustrates our risk prediction methodology along with the evaluation method of the proposed methodology. A literature review on vulnerability prediction studies and discussion of benefits and limitations of the methodology concludes this chapter.

Chapter 6 summarizes the contributions of this dissertation and outlines future research initiatives.

"The man who knows it can't be done counts the risk, not the reward"
Elbert Hubbard

Chapter 2

2. GRC concept

2.1 Introduction

This chapter discusses the GRC concept and analyzes its main components. Specifically, for each GRC component we present essentials and a literature review. For the components of Governance and Risk Management, an analysis and comparison of approaches based on selected criteria is provided. The aim is to support the application of GRC among enterprises as a holistic management approach to complex challenges such as the increased responsibility of stakeholders, diversity of risks and requirements for compliance. In addition to the discussion of the GRC concept, we outline the role of audit as a function that allows performance measurement and monitoring of a GRC program. The remainder of this chapter is organized as follows. In sections 2.2, 2.3 and 2.4, Governance, Risk management and Compliance are analyzed respectively and section 2.5 concludes this chapter.

2.2 Governance

In the following, we present the first component of the GRC concept, Governance. Particularly, essentials, literature review and presentation and comparison of approaches are presented.

2.2.1 Governance essentials

Given the growing value of stakeholders and the emergence of social enterprises, the concept of governance is making a significant contribution in enterprises. The meaning of governance in enterprise environments can best be described with the term Corporate Governance (CG) which defines the set of relationships among various stakeholders providing the structure through which business and security objectives are set, attained and monitored (Mason et al., 2007). Based on governance studies (Da Veiga and Eloff, 2007; IT Governance Institute, 2006; Kritzinger and Von Solms, 2006; Moulton and Coles, 2003; Rastogi and Von Solms, 2006), governance in enterprise environments consists of the following principles:

- Unifies business and security goals
- Clarifies roles and responsibilities among stakeholders
- Raises security awareness and support a risk-oriented corporate culture and
- Ensures regulatory compliance and commitment to ethical code of conduct

After setting governance principles, we study on related terminology since there is many terms in literature that create controversy around the governance function in an enterprise environment. Specifically, the concept of governance dates back in 1970s however, the evolution of the term until today is based on similar terms that either specialize or generalize the function of governance (Cheffins and Brian, 2011). Beginning from specialized terms, ITSG (Information Technology Security Governance) emphasise on the relationships among the Board of Directors and outlines the requirement for compliance, privacy in customer records and secure exchange of business information (Tan et al., 2010). Similar in philosophy, the term ITG (Information Technology Governance) expresses the growing dependency on IT, where enterprises use technology to manage, develop and communicate intangible assets such as information and knowledge (Gremberger, 2004). In a more general manner but in the same perspective, the term ISG (Information Security Governance) highlights the significance of decision making and enhances stakeholders' value via accountability and ownership (Rao et al., 2007). The final term, known either as CG (corporate governance) or EG (enterprise governance), attempts to broaden the application of governance within and outside the enterprise, including transparency in business transactions, ownership of stakeholders and initiation of security awareness programs as part of a risk-oriented culture (IFAC, 2004).

2.2.2 Literature review

This section describes the application of governance in an enterprise environment from several studies. Particularly, Tiwari and Singh (2013), outlined the need for a trustworthy user authentication technique to solve the demand for a strong ISG

program. Towards this perspective, authors suggest biometrics as means to establish user identity and mitigate risk from authentication problems which are considered objectives of a successful ISG program.

Yaokumah (2013), empirically examined how the security governance objectives, namely strategic alignment, value delivery, resource management, risk management, and performance measurement, affect ISG. The author uses a random sampling technique to conduct a web survey from Ghanaian organizations. Based on a regression model, the results reveal a statistical positive relationship between the security governance objectives and security governance effectiveness. The highlights include that resource, performance and risk management are good predictors to security governance and allow for alignment between business and information security.

Stoll and Breu (2012), highlighted the growing importance of ISG and proposed the integration of standard-based management systems such as ISO9001 with information security governance framework, such as ISO/IEC 27001 as means of a holistic approach to ISG. Authors describe the proper implementation of this unified approach to several organizations and suggest that this approach helps improve business strategy, regulatory requirements and security obligations.

Von Solms and Louwrens (2006) examined the relationships between digital forensics and the different forms of governances such as corporate, information technology and information security governance. Authors found that there is an overlap in content among the governance disciplines and consider digital forensics as a vital ingredient for holistic information technology governance.

Nicho (2013), proposed an information system security governance model as means to provide a holistic approach to information security, assurance, audit, governance and compliance. Based on the analysis of reputed information security governance frameworks, models and concepts that adhere to the specific domains, such as COBIT, ITIL, ISO 27002, Risk IT and PCI DSS standard, the author extracts the best practice from each solution as means to achieve strategic integration and secure information security assets in an organization.

Moreover, research on ISG proposed a reference model which aims to help the chief information officer identify and organize information security activities in a prudent manner and allocate responsibilities to the staff. The reference model is based on the Agile Governance Model and is designed to help the cross functional flow of information into the organizational designs (Korhonen et al., 2012).

Another research focused on the role of IT governance on information security regarding the electronic business environment (Dieter et al., 2013). Authors based on the description of reputed IT governance frameworks such as those provided by the IT Governance Institute and the National Cyber Security Partnership, proposed a model which achieves integration through risk management in order to balance value generation and threat reduction.

Gemma and Minero (2013) compared the most reputed legislations against intellectual property law protection in the context of information security. Authors

argue that each method has its own characteristics and there is no plug and play solution to comply with legal issues in favour of intellectual property rights. Governance in IT and business extends far from a password or copy-protection rights which mean that awareness programs, training and maintenance of a mature security culture may bring bigger benefits from a technological tool.

Gelbstein and Kellermann (2012) investigated the standards and best practices, such as ISO 38500 and the initiatives of the Information Technology Governance Institute, which are considered the "best things" towards ISG and discovered that best standards can lead to false practices if the business context is not harmonized and integrated in the governance approach.

Research has also been conducted on whether corporate governance generates financial benefits for an enterprise (Fuenzalida et al., 2013). Specifically, this study examines the relationship between a good corporate governance index (GCGI) and the generation of profits on the Lima Stock Exhange (LSE). Based on an event study on Peruvian firms, authors conclude that good corporate governance practices yield a positive abnormal return of 1% on the announcement day and a monthly return of 3% based on a 4 year evaluation.

Pasquinucci (2007) proposed a practical approach to ISG because, as it is claimed, most traditional governance approaches are not usually followed correctly. This practical approach consists of three elements namely due diligence, compliance, and enablement. The author supports the concept of a capability maturity model as a reasonable and affordable way to discover high risk areas.

Weber (2013), given that there is not an official Internet of Things (IoT) supranational organization with authoritative legal activities, proposed guidelines for the IoT governance framework. These guidelines are summarized in the following pillars: a) legitimacy and representation of all stakeholders, b) transparency, c) accountability, d) IoT infrastructure governance and e) competition law. The notion of IoT governance is an emerging topic proposed by the European Commission.

Von Solms and von Solms (2006) proposed an information security governance model based on the control-direct cycle of the corporate governance. The model is based on two main principles a) the importance of measurability in order to control property and b) directives that represent the expectations and responsibilities of the stakeholders. These two principles apply to three levels namely operational, tactical and strategic level of an enterprise.

Von Solms (2005) investigated whether COBIT and ISO 17799 can co-exist to offer a more complete approach to ISG principles. The author provides a mapping of activities between these two frameworks and demonstrates with examples and scenarios the integration of the frameworks for a more holistic approach to ISG.

Recently, research from Tariq and Abbas (2013) focused on the efficacy of Pakistani code of CG. Authors, based on panel data derived from 119 firms for an 8 year period, found out that a) there is a positive impact of compliance on financial performance and technical efficacy, b) high compliant enterprises are less profitable

than a low compliant enterprise and c) mandatory compliance can have a negative impact on business objectives.

Munisi and Randøy (2013) examined the extent to which African countries have adopted good corporate governance practices. The study is based on hand-collected data on non financial enterprises in Sub-Saharan African countries to conclude that there is a positive association between governance and accounting performance but negative association between governance and market valuation. The audit function and committee are positively associated with accounting performance however there is not a strong indication that all governance practices lead to a better company performance.

2.2.3 Governance approaches

In the following, a brief description of the most reputed governance approaches for enterprises is provided to assist the reader on the selection of the proper approach. A comparison of approaches is presented in Table 1 based on selected criteria such as a) Type of approach, b) Terminology, c) Interoperability, d) Complexity and e) Holistic approach.

Sherwood Applied Business Security Architecture (SABSA)

SABSA (Sherwood et al., 2009) is a methodology that allows the development of security architectures that support business objectives. Specifically, SABSA represents a six-layer framework which covers areas of modern enterprise such as a) strategy, b) design, c) implementation and d) management and operations. The aim is to simplify the complexity of a modern enterprise and assist in the governance of business systems. The approach allows interoperability with TOGAF (The Open Group Architecture Framework).

The deployment on different perspectives from stakeholders and decomposition of enterprise into six layers allows the identification and assessment of component architectures, in terms of understanding protocols in use, system configuration and component interactions. SABSA is regarded as a holistic approach to governance, including risk and compliance perspectives and allows synergy with COBIT, ITIL and ISO/IEC 27001.

Control Objectives for Information and Related Technology (COBIT)

COBIT 4.1 (IT Governance Institute, 2007) provides guidance on IT governance via 34 IT processes on how to control and manage each process. COBIT acknowledges the responsibility of different users, from Executive board (focus on balancing risk

and control investment) to end users (focus on policies). COBIT updated in version 5 encompassing guidelines for IT governance, including those found in Val IT and Risk IT. COBIT has emerged as an IT audit and control framework to become a process reference model that focuses on governance, compliance and management.

COBIT also appeals to different users namely from Executive management (to obtain value from IT investments and balance risk and control investment), to auditors (to validate their opinions and provide advice to management on internal controls). Simply stated, COBIT 5 uses the following principles to optimize information and technology investment and foster shareholders interests. Those principles are as follows: a) meeting stakeholder needs, b) covering the enterprise end to end, c) applying a single integrated framework, d) enabling a holistic approach and e) separating governance from management

The Open Group Architecture Framework (TOGAF)

TOGAF 9.1v (Open Group, 2009) is a framework which consists of the following tools: An architectural development method (ADM), a theorectical base (The enterprise continuum), a technical reference model (TRM) and a standards information base (SIB). In simple terms, TOGAF provides an architectural description of an enterprise into various layers. Each layer is district from the other based on the function and formulation. Layers may vary due to business requirements, engineering specs and technological environment.

Typically, there are four (4) layers such as a) business, b) information, c) system solution and d) technology. In each layer there is a blueprint of processes and assets that should be governed. In summary, TOGAF seeks to become a universal approach to architectural development and governance. Because of its flexibility to cover different types of structures, it can be used by the majority of organizations including governance, large or medium size companies.

ISO 27002

This standard (ISO/IEC 27002:2005) is a code of practice which aims to provide guidance on information security management in the organization with a focus on governance, compliance, security policies and business continuity planning. The content of this standard includes the following: a) Introduction (recommended use of the standard), b) Scope (outlines the purpose of managing information security), c) Terms and Definitions (vocabulary), d) Structure of the standard (the core of the standard consists of a set of controls and implementation example) and e) Risk assessment and treatment (discussion of risk management objectives).

Revised by ISO/IEC 27002:2013, the focus is on selection, implementation and management of controls taking into consideration the organization's information security risk environment. ISO 27002, formally known as ISO/IEC 17799:2005, allows interoperability with ISO/IEC 27001:2005 a standard known for guidelines

about an Information Security Management System (ISMS). The combination of standards provides a holistic approach to information security governance (ISG) with a focus on control requirements.

National Cyber Security Summit Task Force (TASK)

This union proposed a framework and guidelines to help organizations assess business performance and assist in an ISG program implementation (National Cyber Security Summit Task Force, 2004). The principal aim is to identify roles and responsibilities within the management structure. It acknowledges three key elements of governance namely: people, process and technology.

Moreover, it outlines the importance of senior management involvement as means for a more complete governance program and describes how an organization can benefit by reaching a higher level of compliance. The implementation of the framework is based on an IDEAL model, named after the five phases it describes: Initiating, Diagnosing, Establishing, Acting and Learning. Highlights of the framework are recommendations about aligning cyber security with corporate governance and verification/compliance recommendations.

ISO 38500

This standard (ISO/IEC 38500:2008) consists of principles for directors related to use of IT within the enterprise. Recently updated by ISO/IEC 38500:2015, refers to the governance of management of business processes as means to help decision making. The focus is on the control of these processes by internal or external service providers that include senior managers, monitoring specialists, external auditors, product vendors, consultants and professional bodies.

Overall, the objectives of this standard can be summarized as a) stakeholders' assurance towards corporate governance of IT, b) guide directors in governing the use of IT, c) provide a basis for objective evaluation of the corporate governance of IT and d) put into operation a management system with policies and processes that support governance principles.

Criteria/Approaches	SABSA	COBIT	TOGAF	ISO 27002	TASK	ISO 38500
Type of approach	Me	F	F	S	F	S
Terminology	CG	ISG	CG	ISG	ISG	ISG
Interoperability	Y	L	L	Y	L	N
Complexity	L	H	L	L	L	L
Holistic approach	Y	N	N	N	N	N
Certification	Y	Y	N	Y	N	Y

Table 1. Comparative evaluation of governance approaches

Legend:

1) Type of approach (F for Framework, S for Standard, M for Model, Me for Methodology)

2) Terminology (ITSG, ITG, ISG, CG)

3) Interoperability (Y for Yes, N for No, L for Limited)

4) Complexity (L for Low, H for High)

5, 6) Holistic approach, Certification (Y for Yes, N for No)

Based on Table 1, existing governance approaches differentiate based on how well they satisfy the selected criteria. The criterion Complexity is scored either L or H based on the easiness to adapt and the time required for implementing. A holistic approach is given a Y or N indication based on whether the approach supports risk management and compliance issues or not.

For example, SABSA methodology consists of risk and policy guidance which implies that presents a holistic approach. An organization should leverage its structure based on a single approach however, for maximum results, it seems more appropriate that a combination of governance approaches may best suit an enterprise opting for the ultimate solution in the governance domain.

2.3 Risk management

Historically, the field of risk management has been dominated by theoretical discussions, practical misfits and indecipherable algorithms all of them adding to

complexity and little in essence. Recent corporate failures, such as the collapse of Lehman Brothers which caused severe consequences including economic turndown and an extended systemic risk in every sector or industry, reveal the failure to identify and manage risk at an enterprise level.

In the following essentials, we present the evolution of IT risk management and decompose enterprise risk. In addition, we introduce the reader with phases that consists a risk management process and give emphasis on enterprise controls to assist in the protection of enterprise assets. A literature review on studies related to ERM is presented and we conclude this section with a presentation and comparison of enterprise IT risk management approaches.

2.3.1 Risk management essentials

Modern enterprise risk management (ERM) depends on IT to perform its purpose of control and protection. During time, this relationship has evolved but to what extent? Evolution reveals that first attempts on managing risks in enterprises started as isolated and stand-alone process before becoming fully integrated with the business processes. Figure 1 demonstrates the evolution of IT risk management.

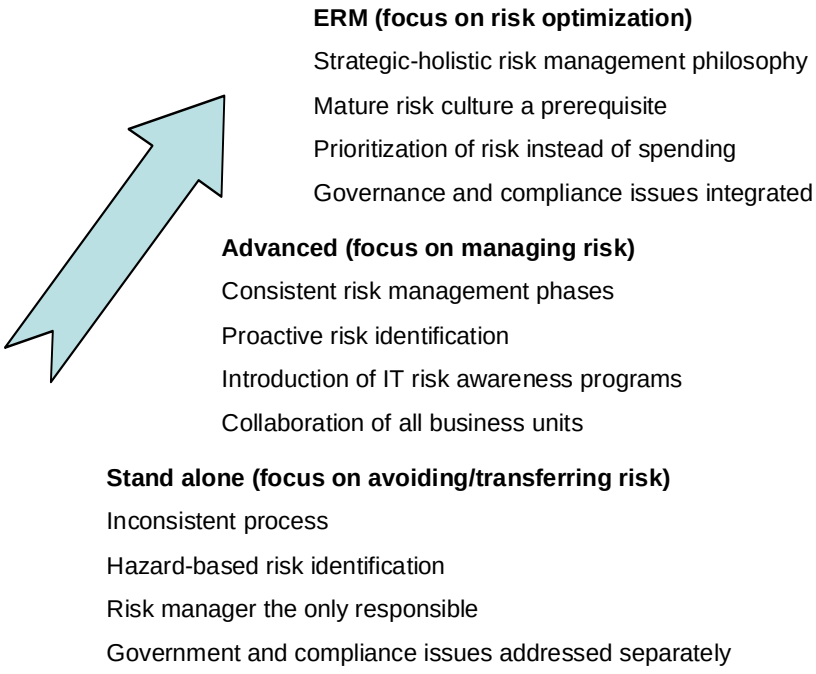

Figure 1. Evolution of IT risk management

According to Figure 1, first, there was the philosophy that risk should be avoided at all costs. This notion was supported by the fact that the majority of enterprises transferred business and IT risk to third party insurance companies. This notion became quickly outdated since business community started to realize that managing enterprise IT risk is not an individual responsibility and transferring risk is not a viable option. Therefore, enterprises started to align IT risk management as part of business activities with sight of managing risk rather than avoiding it.

This brought up the need for IT security awareness programs and training as well as the involvement of all business units. However, there were missing parts, such as governance and compliance issues. Towards this perspective, the term Enterprise Risk Management (ERM) emerged to address the limitations of previous notions, such as static risk management procedures and the need to include governance and compliance issues into a unified approach (Hampton, 2015). ERM principles are summarized in the following bullets.

- Satisfy stakeholders' goals
- Surpass static risk methodologies
- Increase transparency of operations
- Add value and communicate decision making across the organization
- Provide differentiation and competitive advantage

Successful ERM requires a solid grasp of what is happening within and outside the enterprise. COSO defines ERM (COSO, 2004) as "a process, affected by an entity's board of directors, management and other personnel, applied in a strategy setting and across the enterprise, designed to identify potential events that may affect the entity, and manage risk to be within its risk appetite, to provide reasonable assurance regarding the achievement of entity goals". Sound examples, such as General Motors, Lewis and Wal-Mart, use ERM initiatives to strengthen governance processes via the internal audit function. Organizations have the opportunity to gain a competitive advantage by using compliance efforts to build balance controls that are sustainable and add long term value to the organizational structure (Grace et al., 2014; Lam, 2014).

Developing effective ERM strategies requires the collection of data from various stakeholders from the enterprise's environment. Multiple feedbacks allow for the development of an ERM strategy that differentiates and provides the competitive advantage by nurturing a risk-oriented culture capable to add value to the enterprise and become a proactive solution to IT risks. Towards this philosophy, stakeholders should develop a high level of competence reflecting the skills and know-how to perform assigned tasks (Hoyt and Liebenberg, 2011).

Decision making should be delegated for a more decentralized and flexible operational work flow however this may increase the number of undesired events and

affect the internal environment if individuals are not hold accountable for their actions. In this regard, segregation of duties (SoD) is considered a key component to maintain a strong internal control environment because it delegates responsibility to those individuals capable to accomplish a task and avoid a fraudulent activity (Taylor, 2014).

Addressing enterprise risk holistically is not trivial because of the diversity of enterprise risks. Consequently, this raises a major question such as *what kind of risks faces an enterprise?* The answer to this question lies on decomposing the nature of enterprise risk into other types of risks that are all considered IT-related risks. Figure 2 illustrates how enterprise risk is decomposed into other types of risks and a brief description follows in bullets.

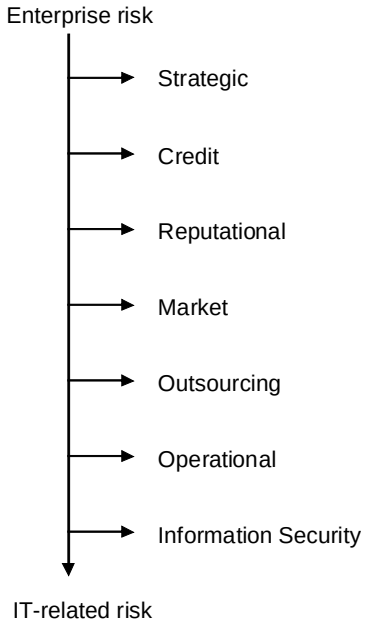

Figure 2. Decomposition of enterprise risk

- **Strategic** risks are mainly associated with the board of directors and management decisions. Factors affecting strategic risks are summarized as follows:
    - Planning and investment decisions
    - Design, delivery and pricing of services and products
    - Competition issues
    - Third party/outsourcing agreements
    - Customer support services

- **Credit** risks derive from the inability of counterparties to meet financial obligations. Factors affecting credit risks are summarized as follows:
    - Changes in interest rates
    - Political instability
    - Volatility in the banking and finance
    - Liquidity issues

- **Reputational** risks involve issues related to public opinion and to actions that create and promote negative public relationships. Increased reputational risks impair customer relationships and also damage profits and competitive position. Factors affecting reputational risk are summarized as follows:
    - Loss of trust due to unauthorized activity on customer accounts
    - Failure to deliver marketing plans
    - Increased customer complaints
    - Hacking on an enterprise website

- **Market** risks derive from the industry and competition within which the enterprise operates. Factors affecting market risks are summarized as follows:
    - Market recessions
    - Terrorist attacks
    - Political instability
    - Mergers and acquisitions
    - Changes in demographic and customer profile
    - Currency changes

- **Outsourcing** risks derives from third party dependencies and services agreements. While enterprise outsourcing may provide a number of advantages such as cost benefits and optimization of in-house activities, this also becomes a great source of other risks including strategic, reputation, compliance, operational, exit strategy and systemic risk. Factors affecting market risks are summarized as follows:
    - Incomplete third party contracts

- Inability to comply with third party regulatory environment
- Reputation of the partner

- **Compliance** risk emerges from violations or non-conformity with laws, regulations, practices or ethical standards. Non-compliance has serious consequences including financial penalties, damage to reputation, rating downgrades and removal of authority to operate. Factors affecting compliance risk are summarized as follows:
    - Staff expertise in following specific policies
    - Regulatory requirement within the industry the enterprise operates
    - Internal and external audit results
    - Third party dependencies.

- **Operational** risk derives from inadequate or failed processes, people, or systems affecting the enterprises' ability to deliver products and services. This type of risk has a direct impact on customer services. Factors affecting operational risk are summarized as follows:
    - Internal and external fraud
    - Lack of training and misuse of confidential information
    - Business disruption and damage to physical assets

- **Information Security** (IS) risk derives from threats exploiting vulnerabilities that reside in an information infrastructure of an enterprise. Every information infrastructure is composed from enterprise information systems, such as hardware, software and applications. Factors affecting IS risk are summarized as follows:
    - Human involvement
    - Evolution of technology
    - Misconfigurations of settings
    - Adequacy of controls
    - Loss in security properties, namely confidentiality, integrity and availability (CIA).

2.3.2 Risk management phases

The road to implementing an enterprise IT risk management process may seem vague and overwhelming if guidelines do not exist to follow. In this respect, in Table 2 we present the three high level phases, identification, assessment and monitoring, to introduce the reader with what needs to be followed.

Phase	Brief description
Identification	Requires collecting data from decomposing enterprise risk based on risk scenario analysis
Assessment	Requires analyzing risk, taking into consideration the business relevance of risk factors
Evaluation	Requires maintaining a risk profile as an inventory of threats, vulnerabilities and their attributes, as well as monitoring their status over time

Table 2. High level risk management phases

The **first phase** is the identification of IT-related risks. A core approach to risk identification is the use of risk scenarios as means to decompose the complex nature of enterprise risk. A risk scenario can be described as the happening of an event that can lead to a business impact, if and when it occurs. In practice, the combination of generic and customized risk scenarios is the preferable solution to identifying risk. Figure 3 illustrates an indicative list of components that synthesize risk scenarios.

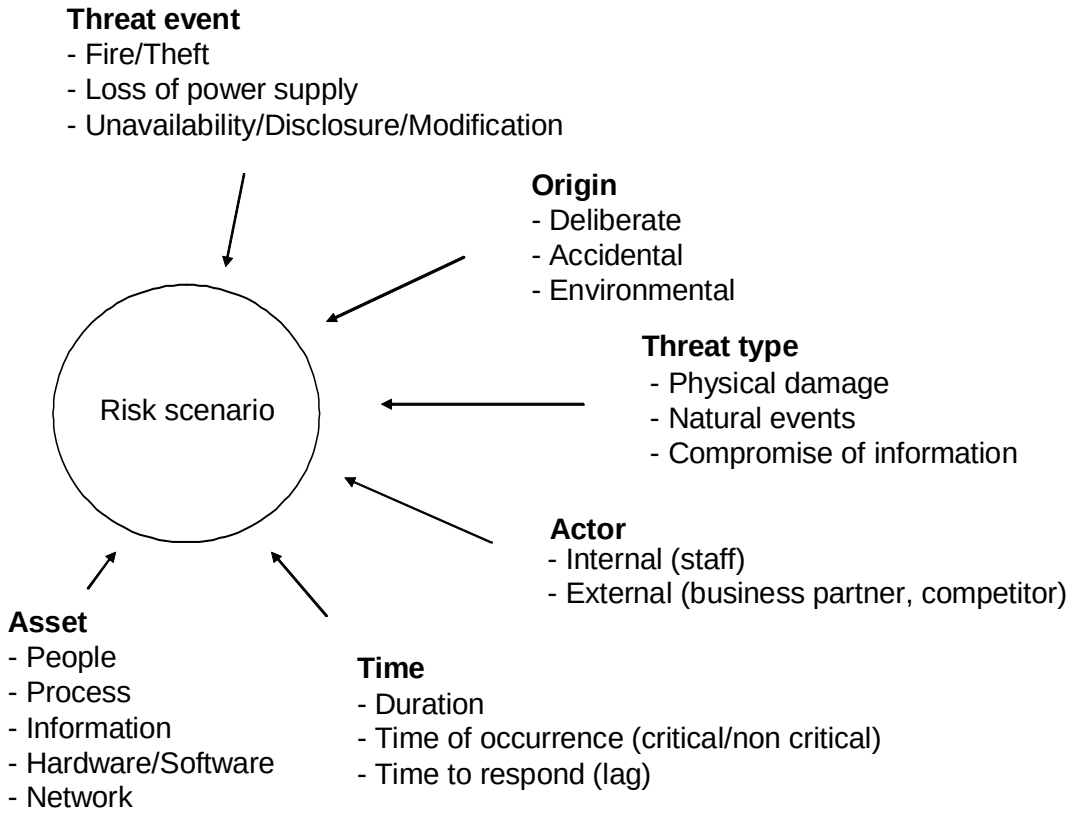

Figure 3. Factors affecting risk scenarios

Ideally, risk scenarios should include threats and vulnerabilities of current and future conditions including trends in technology, compliance and governance as well as changes in the business environment. The realism of risk scenarios depend heavily on the recognition and importance of risk factors. Risk factors are the conditions that shape the likelihood and impact of risk scenarios. Table 3 presents indicative risk factors that affect risk scenarios.

External	Internal
Market/economic status	Organizational structure
Rate of change	Complexity of information infrastructure
Industry/competition	Maturity level of culture towards risk
Geopolitical situation	Risk management philosophy
Regulatory environment	Business model
Technology trends	Change management capability

Table 3. Risk factors

After building a list of risk scenarios, the **second** phase is the assessment of the identified risk scenarios. This means that analysis should be conducted in terms of the likelihood and impact each scenario might have on enterprise goals. Towards this endeavour, risk practitioners use either qualitative or quantitative risk analysis.

Qualitative risk analysis is based on a scale of comparative values, such as low, medium, high or on numeric scale such as 1 to 10, to reveal the criticality of risk. This kind of analysis is based on subjective judgment and personal experience rather than on monetary values or statistical data. The aim is to provide an easy-to-understand result and is basically used to evaluate intangible assets such as reputation or image recognition. Typical qualitative methods include scorecards, likelihood - impact matrix, Delphi forecasting and failure modes and effects analysis (FMEA). Table 4 demonstrates the pros and cons of using qualitative analysis.

Pros	Cons
Low cost	Subjectivity/Bias
Easy to understand	Unsuitable for cost-benefit analysis
Consensus-based	Fuzzy ranking
Suitable for intangible assets	Low validity

Table 4. Pros and Cons of qualitative analysis

On the other hand, quantitative risk analysis is based on numerical and statistical techniques to calculate risk. This type of measurement produces more precise results in financial terms and is highly recommended when a cost-benefit analysis of controls is required. However, challenges such as the desired data format or the requirement of standardized historical data are not always available for analysis. Typical quantitative methods are the internal loss data, business process modelling (BPM) and statistical analysis methods. Table 5 demonstrates the pros and cons of using quantitative risk analysis.

Pros	Cons
Precision of results	Time-consuming
Statistically reliable results (objectivity)	Data collection in the desired format is a challenge
Allows for cost-benefit analysis	Increased cost
Provides anonymity of results	Requires higher staff expertise

Table 5. Pros and Cons of quantitative analysis

Recently, there is an increased interest on hybrid methods that combine both quantitative and qualitative analysis. Therefore, semi-quantitative/qualitative methods classify risk using a combination of numerical values (Nikolic and Ruzic-Dimitrijevic, 2009). Usually, this type of analysis starts with qualitative measurement based on opinions and follows with quantitative analysis of data.

The **third** phase is risk monitoring which is to control and maintain the performance of enterprise IT systems. This phase requires risk indicators to monitor business activities. This implies that risk thresholds (i.e. risk limits) should be clearly defined to address the acceptable levels of risk the enterprise is willing to accept and manage. Each enterprise varies in risk appetite and tolerance hence risk thresholds should be based on strategic orientation, on the size and complexity of information infrastructure and the type of market in which the enterprise operates. Risk thresholds are supported by risk indicators that act as warnings to risks exceeding a specified level. Factors influencing the selection of risk indicators are depicted in Table 6.

Factors	**Brief description**
Stakeholders	Risk indicators are selected from stakeholders to ensure that satisfy business goals and security requirements
Balance	A balance of indicators include: • Lag indicators (post-indication of risk) • Lead indicators (preventive indicators) • Trend indicators (analyzing trends and insights)
Root cause	The effectiveness of indicators depends on the capability to trace the root cause of events, not just the consequences
Cost-benefit	Indicators should be based on a cost benefit analysis in order to optimize security spending and maximize business performance

Table 6. Factors influencing the selection of risk indicators

When risk exceeds pre-defined thresholds, a risk indicator triggers a warning that enables stakeholders to take appropriate actions. In this respect, stakeholders should decide what controls should put in place in order to remediate risks. The term "enterprise control" describes the set of policies, procedures and behaviour designed to mandate the operation of an enterprise by specifying what actions are, or are not,

permitted (Lam, 2014). Table 7 describes the most common enterprise control categories.

Category controls	Description
Compensating	Controls designed to make up for the weakness in an existing control structure of the enterprise *Example:* Adding an additional verification procedure in an existing weak access control mechanism
Corrective	Controls designed to remediate errors after detection *Example*: Back up and restore procedures
Detective	Controls designed to provide warnings of attempted violations *Example:* Intrusion detection methods
Deterrent	Controls designed to deter a potential compromise *Example:* Login screens
Directive	Controls designed to direct the behaviour within the enterprise operates *Example:* Policies
Preventive	Controls designed to inhibit violation of a security policy *Example:* Access control methods

Table 7. Enterprise controls

Controls' effectiveness significantly influences the enterprises' risk profile. Therefore, the mix and importance of controls will be unique for each enterprise. In order to maximize the operability of controls, the risk practitioner must ensure that controls are properly managed throughout the various phases in the control life cycle. Figure 4 shows the phases of the control life cycle and each phase is briefly described.

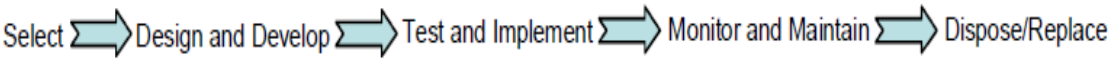

Figure 4. Control life cycle phases

- **Select**: The selection of enterprise controls depend on factors such as a) total cost of ownership (TCO), b) time constraints, c) personnel expertise, d) business priorities and e) security requirements.

- **Design and Develop**: This phase has to consider the "breadth" and "depth" of controls. The term "breadth" corresponds to the flow of information across multiple applications and "depth" corresponds to the different layers on which controls apply, as well as the dependence of a control to the operation of other controls.

- **Testing and Implementation**: Control effectiveness can be assessed by quantitative and qualitative testing to determine how well the control objectives are satisfied. Criteria that can be used to test and implement enterprise controls are, but not limited to, control sustainability, scalability, customizability, performance measures, interaction with other controls, complexity and return on investment (ROI).

- **Monitor and Maintain**: This phase includes monitoring the performance of enterprise through controls and reporting on the effectiveness of controls as individual enterprise units. Controls should be updated regularly to maintain secure operation according to vendors' configuration.

- **Dispose/Replace**: The enterprise is responsible for ensuring that only the necessary controls are in place to monitor the status of an enterprise. In this respect, control effectiveness and cost benefit analysis should be conducted regularly to ensure whether a particular control is operational and sustainable at the time of the analysis. Optimization of resources requires that an enterprise utilizes controls that are perceived justifiable in monetary terms and satisfy business and security goals.

2.3.3 Risk management approaches

This section aims to assist the implementation of an IT risk management method within the enterprise environment. Generally, an effective IT risk management method should be based on well-defined steps, from identification to mitigation, underpin all enterprise activities and follow a formal governance process. According to literature (Koons & Minoli, 2010; Landoll, 2006; Strecker et al., 2011; Wheeler, 2011), most common challenges existing IT risk management methods share are summarized in the following bullets.

- **Disparity of risk terminology**. It is common that every method, technique, standard or other, tend to differentiate. This differentiation includes inventories of assets, vulnerabilities, threats and controls.

- **Complex (inter)operability**. The majority of methods is deemed too complex to be deployed in real terms because either they require expertise personnel to be deployed or is based on an overwhelming theoretical background which is more confusing rather than practical. Moreover, the lack of interoperability usually derives from qualitative risk analysis methods that lack the accuracy to compare results with other methods due to ambiguous risk scales (e.g. a low score may be conceived differently among users).

- **Lack of a holistic approach**. Most methods focus on the technical side of risk, such as risk analysis of vulnerabilities and threats, leaving aside governance and compliance issues which imply that risk methods lack of unified approach to enterprise risk management.

To confront the challenges, eleven risk management approaches, in terms of eight evaluation criteria, are compared. Key requirements for each criterion are presented in the following bullets. A short description of the IT risk management methods is provided after the criteria analysis.

- **Risk identification**. The key requirements are as follows:
    - Identify and anticipate assets, threats and vulnerabilities
    - Set the risk appetite and determine the level of acceptable risk
    - Adopt a common risk language and support a risk oriented culture
    - Identify compliance requirements

- **Risk assessment**. The key requirements are as follows:
    - Analyze and prioritize risks
    - Indicate the relationship among risks (dependencies)
    - Incorporate opinions from various stakeholders
    - Evaluate past data incidents

- **Risk monitoring**. The key requirements are as follows:
    - Implement risk indicators
    - Tracking the source of risk
    - Engage key stakeholders
    - Reporting

- **Risk mitigation**. The key requirements are as follows:
    - Induce risk responses
    - Avoid
    - Reduce
    - Share/transfer

- Accept
- Justification of controls (cost-benefit)
- Accountability of controls

- **Compliance**. The key requirements are as follows:
  - Provide evidence of compliance
  - Justify compliance financially
  - Enable measurement of compliance
  - Integrate compliance with business and security goals
  - Educate personnel/allocate responsibilities

- **Complexity**. The key requirements are as follows:
  - Usable and repeatable results
  - Time to learn and adapt to the method
  - Easy to use
  - Add practical value to the enterprise operation

- **Interoperability**. The key requirements are as follows:
  - Allow synergy with other methods
  - Consider PEST factors
  - Support a common risk terminology

- **Governance**. The key requirements are as follows:
  - Align accountability to different stakeholders
  - Specify roles and individual responsibilities
  - Provide strategic direction through decision-making
  - Enable protection and enlargement of stakeholder value

- Focus on resource management

In the following, a brief description of the most reputed enterprise IT risk management approaches is provided to assist on the selection of the proper approach. A comparison of the approaches is presented in Table 8 based on the aforementioned criteria.

OCTAVE (Operationally critical threat, asset, and vulnerability evaluation)

OCTAVE is an engineering-oriented, risk management technique which aims to support business goals and security priorities by a) identifying critical assets, b) assessing risks to critical assets and based on the assessment it c) provides adequate reasoning about decision making (OCTAVE, 2003). The approach is context-driven and self-directed requiring a small interdisciplinary team to perform the gathering, analysis of results and recommendations on risk management strategies. OCTAVE uses for analysis catalogs of information known as OCTAVE criteria. The catalogs are an inventory of best security practices, threat profiles and vulnerabilities.

OCTAVE provides spreadsheets to support its process through a three-phase approach. Particularly, phase 1 includes asset-based threat profiles development (organizational evaluation), phase 2 includes identification of infrastructure vulnerabilities (information infrastructure evaluation) and phase 3 the development of security strategy and plans (risk prioritization). The next generation is the OCTAVE Allegro (Caralli et al., 2007), a method suitable for enterprises with a limited investment in terms of time, human resources, and other resources.

COBIT (Control Objectives for Information and Related Technology)

COBIT 4.1 and recently announced fifth version, is a process reference model for business and IT – related goals (IT Governance Institute, 2007). Based on clear metrics, the model allows data collection based on the risk appetite and tolerance of an enterprise. The fifth version is composed from distinct processes that are split into two dimensions: a) enabler dimension and b) enabler performance management.

The first dimension involves issues and processes about a) stakeholders, b) achievement of goals, c) control life cycle and d) recommended security practices. The second dimension involves the metrics and processes that allow the measuring of the first dimension. COBIT 5 provides a complete basis for an audit, compliance, and governance mindset such as that found in the financial services sector. COBIT allows interoperability with COSO, an internal control framework for enterprises.

CRAMM (CCTA Risk Assessment and Management Methodology)

CRAMM version 5 is an advanced risk analysis method which is supported by the Central Communication and Telecommunication Agency (CCTA) (British CCTA, 2003). CRAMM consists of a comprehensive range of risk assessment tools that support security management, creation of security policies, documentation, business continuity management and certification or compliance to ISO 27001. Particularly, CRAMM is compliant with ISO 27001:2005 a standard that defines requirements for an Information Security Management System (ISMS).

The method itself is composed of a software tool (aka CRAMM tool) which performs the risk assessment. The risk assessment based on input of data, such as asset identification, analyzes the security infrastructure with asset dependency modelling and business impact analysis. The assessment of risk leads to a recommendation of controls. CRAMM also provides a cost-benefit analysis based on estimated cost of controls.

ISO/IEC 27005

ISO 27005 is a risk management standard, revised as ISO/IEC 27005:2011, which provides guidelines about the risk management phases (ISO/IEC 27005:2008). The standard has normative references the ISO/IEC 27001:2005 and ISO/IEC 27002:2005, referring to security requirements and codes of practice respectively. ISO 27005 offers advice on risk identification, analysis and assessment, and remediation in terms of risk treatment to all kinds of enterprises.

It recommends qualitative risk analysis in terms of risk matrices as well as inventories for asset, vulnerability and threat. The standard offers a vocabulary based on ISO/IEC Guide 73:2002 and the highlight is the general description of information risk assessment process which includes input, action, implementation guidance and output guidelines. The standard outlines the dependencies of assets on business processes and recommends modification of asset values due to dependency modes.

ISAMM (Information Security Assessment and Monitoring Method)

ISAMM is a quantitative method for identifying, assessing and supporting decision making about controls and risks (ISAMM, 2002). ISAMM supports an Information Security Management System (ISMS) for obtaining ISO 27001 certification and consists of four phases namely: a) the scope of the assessment which includes identification of assets, threats and controls, b) assessment of compliance (vulnerability) and threats, c) validation of compliance and threats and d) analysis and reporting.

The method follows the set of controls from the ISO/IEC 27002 and provides realistic improvement through simulation and residual risk evaluation. Moreover, the analysis is based on monetary metrics such as annual loss expectancy (ALE) and return on investment (ROI). ISAMM also gives emphasis on decision support for acceptability of risks and selection of safeguards.

FRAP (Facilitated Risk Assessment Process)

FRAP is a business-led, qualitative risk assessment process with purpose to assess information security risks (Petlier, 2000). The method allows for a) threat identification, b) likelihood of threat and impact evaluation, c) risk level determination and d) controls' recommendation. FRAP does not identify assets in terms of asset inventory but on threats' materialization. However, this implies that the risk assessment of a threat is calculated based on a defined asset.

The process itself uses concepts derived from ISO/IEC Technical Report 13335-3:1998 which is a standard referring to techniques for the management of IT security. The process is established via a framework which shows how to prepare the risk assessment approach. The highlight of FRAP is that it uses the business profile of the organization to perform the risk assessment process in order to produce findings that are valuable by business owners.

ETSI TVRA (European Telecommunication Standardization Institute Threat Vulnerability and Risk Analysis)

This method uses qualitative analysis in the form of risk matrices to determine the risk and is best suited for networking applications (European Telecommunications Standards Institute, 2006). TVRA consists of 7 distinct phases: a, b) identification of security properties and requirements, c) identification of assets through an asset inventory, d, e) classification of vulnerabilities and threats and evaluation of likelihood and impact, f) determination of risks and g) countermeasures guidance.

In 2010, an updated method consisting of 10 phases is introduced. TVRA allows interaction with ISO 15408, a standard also known as Common Criteria, an IT security evaluation methodology for systems and products. The highlight of this method is the last step which includes specification of countermeasures in order to reduce the likelihood and impact of an attack.

NIST SP 800-39

This is special publication from the National Institute of Standards and Technology that outlines the need for integrated, organization-wide risk management (NIST, 2011). It describes risk as a strategic capability that entails compliance and governance to fulfil the risk management process. This publication allows interoperability with other NIST and ISO publication such as NIST SP 800-37, a guide for applying the risk management framework to Federal information systems and ISO/IEC 27005 a risk management standard.

NIST SP 800-39 proposes risk-based strategies that aim to a) manage the risks, b) determine input from the board of directors, c) justify the selection of controls and e) monitor the level of organizational risk over time. The publication highlights include a) stakeholder accountability b) a Risk Management Framework (RMF) and c) outsourcing relationships. For the outsourcing relationships, the standard recommends trust relationships between external service providers and the organization.

MAGERIT (Methodology for Information Systems Risk Analysis and Management)

MAGERIT is a Spanish method and the foundation derives from the need to control governance agencies dependence on information infrastructures (MAGERIT, 2006). MAGERIT places emphasis on risk analysis based on assets, threats and safeguards identification. Specifically, the MAGERIT model refers to inherent and residual risks based on risk analysis. The methodology supports security awareness training programs in order to foster the cultivation of a risk-based culture.

MAGERIT v2, published in 2005, consists of three books a) the methodology itself, b) an inventory of criteria and risk modelling and c) risk analysis techniques. The method proposes a holistic approach to risk management, operation and change management, incidents reporting and audit certification. There is no specified compliance with a certain standard however; the method is recognized from the Organization for Economic Cooperation and Development (OECD) which includes good practices for internal controls, ethics and compliance. Recently, MAGERIT updated in its version 3 in 2012 (in Spanish) to manage e-government principles.

PTA (Practical Threat Analysis)

This is a threat modelling methodology developed by the PTA technologies to enable users find the most cost-effective countermeasures to secure critical assets (PTA technologies, 2005). PTA is known for its own libraries which are inventories consisting of pre-defined assets, threats, vulnerabilities and controls. PTA libraries allow compliance with standards such as ISO 27001 and 27002 and PCI Data Security Standard 1.1 (Payment Card Industry, 2010).

The PTA threat model consists of four distinct phases namely a) asset identification b) vulnerabilities identification, c) safeguards cost-benefit analysis and d) threat scenarios and mitigation plans. The highlight of the methodology is the cost-benefit analysis of safeguards because the output of the analysis aims to develop a plan that enables optimized selection of safeguards on the basis of vulnerability and threat results.

ISO 31000

ISO 31000 belongs to the ISO family and is dedicated to generic risk management (ISO 31000:2009). It defines risk management as an architecture that is used to handle risk. It proposes a risk management framework that comprises of foundations and organizational arrangements. The former includes policy, objectives mandates and commitment and the latter includes relationships and accountabilities towards risk management.

The standard outlines the need for a risk-based attitude and provides guidelines on monitoring risks however; its theoretical and generic base does not allow for a detailed guide in an IT domain and lacks the notion of monitoring risks. The interoperability of this standard is limited to the ISO/IEC Guide 73, which a risk management vocabulary. The concept of governance is described in terms of stakeholder commitment and policies. The overall objective of this standard is to improve decision-making, balance uncertainty and chance and apply change management in enterprises as they progress.

Approaches	OCTAVE	COBIT	CRAMM	ISO 27005	ISAMM	FRAP	TVRA	NIST 800-39	MAGERIT	PTA	ISO 31000
Criteria											
Identification	✓	✓	✓	✓	✓	/	✓	✓	✓	✓	✓
Assessment	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Monitoring	x	✓	/	/	✓	x	/	✓	✓	✓	x
Mitigation	/	✓	✓	✓	✓	✓	/	✓	✓	✓	/
Compliance	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	x
Complexity	x	/	x	x	/	x	x	x	x	x	x
Interoperability	/	✓	/	x	✓	/	✓	✓	/	✓	/
Governance	x	✓	/	x	x	x	x	✓	✓	/	/

Table 8. Risk management methods comparison (✓ = yes, / = partial, x = no)

The scoring of criteria in Table 8 is based on a subjective comparison of the risk management approaches presented. For example, FRAP scores partial in the identification criterion because it does not contain a stand alone asset inventory, contrary to the PTA or MAGERIT. However, the threat evaluation is conducted with an asset under consideration. Another example, OCTAVE does entail a risk assessment process that allows identification of cause and effect relationships, however; it does not include stand-alone governance initiatives.

Overall, COBIT appears to offer a more holistic approach to ERM in terms of audit and governance initiatives however, due to the degree of customization required, the complexity to deploy COBIT may be overwhelming for small to medium enterprises (SMEs). Without doubt, there is no such thing as a perfect risk management method. Each one is recognized for a particular domain, for example ISAMM is known for its

quantitative analysis. Therefore, based on a review on risk management approaches and the outcome of Table 8, we recommend the selection of an approach that is tailored to fit particular environments, such as MAGERIT for governance agencies or ISO 31000 and ISO 27005 for general applicability.

2.3.4 Literature review

Mathrani & Mathrani (2013), examined enterprise risk management (ERM) from the perspective of enterprise IT systems, such as software applications from Oracle and SAS. Authors concluded that such systems improve organizational efficacy and manage risks through analysis and reporting. Authors, based on a qualitative research methodology concluded that enterprise IT systems functionalities extend from an ordinary activity to decision making.

Tekathen & Dechow (2013), studied how ERM can lead to accountability. The study, based on COSO principles about ERM, observes that implementation of an ERM practice does not necessarily ensure organizational risk management. Authors argue that ERM does not minimize uncertainty but sometimes creates more. In terms of personal responsibility to risk management, authors suggest that there should be a clear distinction between local and global processes.

In addition, Caron et al., (2013) examined ERM from a process mining perspective. This study explores on the applicability of process mining techniques, such as fuzzy and heuristics miner, as means to support the activities related to the different phases of risk management such as identification, assessment and monitoring. Authors demonstrate the process of data mining based on Belgian practices from the insurance industry and conclude that analysis of internal data over external data can offer more reliable and less biased information for the ERM functions.

Research from Quon et al., (2012), focused on whether enterprise risk management affects enterprise performance. Authors claim that ERM has been examined in the context of governance and internal control but the relationship to enterprises' performance has yet to be established. Taking into consideration the current financial crisis, the study concludes that an ERM concept did not predict or affect significantly business performance.

Arnold et al., (2011) focused on the role of strategic ERM towards easing regulatory compliance. The study examines whether ERM processes can react to regulatory mandates, under the SOX 404 internal control reporting requirements. Research findings, based on 113 executive responses, reveal that strong ERM processes can smooth the implementation of SOX mandates, increase organizational flexibility and IT integration.

Moreover, Saleh & Alfantookh (2011) proposed an information security (IS), risk management framework as means to offer a holistic approach to enterprise security. This framework consists of two structural and two procedural dimensions which

include the scope, assessment criteria, process and assessment tools. The highlights of the framework include the STOPE (Strategy, Technology, Organization, People, and Environment) approach and the six-sigma DMAIC (Define, Measure, Analyze, Improve, and Control) cycle.

Gordon et al., (2009) suggested that for a holistic approach towards managing risk in an enterprise, the concept of ERM is required. The study is based on a sample of more than 100 US organizations and concludes that ERM should work in conjunction with the variables that affect enterprises' performance. This performance is comprised of five factors namely environmental uncertainty, industry competition, firm size, firm complexity, and board of directors' monitoring.

Arena et al., (2010) focused on the organizational dynamics of ERM. Authors suggest that ERM is capable to organize uncertainty taking into consideration organizational dynamics in the form of control, accountability and decision making. Based on a 7 year period study, authors conclude that the rise of ERM has led to the emergence of a) Chief Risk Officer (CRO), b) internal audit function, c) different actors that induce uncertainty and d) governance and compliance issues.

2.4 Compliance

Last but not least in importance is the last acronym of GRC, Compliance. Therefore, we describe in compliance essentials its function, principles, trends and strategy. Moreover, we elaborate the compliance function in e-banking and highlight on the main challenges banks face in this domain. This section concludes with a literature review on the role of compliance in enterprises

2.4.1 Compliance essentials

The compliance function is the ability of an enterprise to conform the internal codes of conduct, business procedures and government policies to the regulatory environment. If this conformity shatters or not managed properly and proactively, then a number of risks are likely to emerge, such as reputational and financial risks. Compliance entails responsibilities that begin from the board of directors who share the leadership role in developing a compliance strategy. The board of directors usually delegates compliance tasks to senior level management in the form of policies that reflect compliance requirements.

Truth is that compliance with a series of legal and regulatory requirements can be expensive, time consuming and resource demanding if the strategy to compliance is not set according to business and security goals for each enterprise. In this regard, the

compliance strategy should be flexible enough to accommodate changes in regulations and at the same time comprehensive and rigorous to comply with existing regulations. Current trends in compliance include the shift from a rigid compliance-based supervisory approach to a more risk-sensitive approach which seeks to encourage innovation and experimentation (Usman and Shah, 2013). Particularly, to transform compliance into a competitive advantage and differentiate from competition, an enterprise should integrate compliance requirement into business and security objectives, monitor performance and adapt to compliance updates. In this respect, the task of compliance can become a reference model for others.

Important for the compliance function to become effective is to help create and foster a risk-based organizational culture that incorporates the compliance function. This includes educating top management and employees on regulation updates and user responsibilities. Compliance is also concerned with transparency of actions required to conduct business or mitigate risks. Truth is that the majority of organizations have a risk management approach which they follow however, mitigating the identified risks is not always a success because decision making is poor and controls do not always bring the desirable results (Koons and Minoli, 2010). Standards and policies do help in achieving compliance however, there will be issues of non compliance if monitoring and reporting of compliance performance is not automated. In this respect, security metrics (see Chapter 3) can automate the compliance function and integrate it with the organizational mission.

A common obstacle in law conformity is that the law can not possibly have the same flexibility when technological changes occur. Nowadays, the majority of enterprises are global in nature and business operations allow for cross-country electronic transactions. This implies that legal requirements may differ from country to country depending on the type of business, country-specific regulations, outcome of the act and others (Abdollahi and Afzali, 2014; Mermod, 2011). Therefore, it is essential for enterprises that sustain a global presence to comply with laws and regulations in all countries in which they operate.

Moreover, the organization of compliance is attained differently among enterprises. Particularly, large enterprises tend to locate the compliance function within operating business lines whereas multinational enterprises have group and local compliance officers depending on the country they reside (Kondabagil, 2007). In smaller enterprises, compliance function may be located in one unit. Placing compliance in one unit as being an independent function appears ideal for specialized areas such as data protection, prevention of money laundering and terrorist financing. Given the strong connection between business, operation and compliance functions and taken into consideration potential overlaps, the organization of the compliance function should be subject to the following principles:

- Integration with business and operational activities
    - Manage compliance risks
    - Compliance policies

    - Compliance accountability

- Availability of resources and updates on regulations
    - Ensure compliance is up to date
    - Predict compliance trends
    - Assure resources allocation

- Compliance monitoring
    - Allocate security metrics to check compliance status
    - Provide a real –time evaluation of compliance performance
    - Provide consistent compliance reporting

- Nurture of an enterprise compliance culture
    - Assign compliance responsibilities
    - Enforce compliance training and evaluation
    - Respond to compliance violations

As the borders of financial institutions grow increasingly complex, due to globalization, advent of internet and networks, user requirements for mobility and automation, so does the regulations that govern them. In this regard, moving from traditional to electronic services, we present the compliance function in e-banking. This complex electronic environment is now the new normal in banking services and compliance should follow high standards of integrity and performance to allow for legal and consistent online banking services, otherwise operational and customer experience risks are likely to emerge.

2.4.2 Compliance function in e-banking

Nowadays, there has been intense legislative and regulatory activity around e-banking to address the numerous opportunities and threats brought by the advent of electronic commerce and banking, industry competition and stakeholder demands (Mu, 2003). Financial institutions have been always on the hot spot as far as compliance challenges are concerned. This is because electronic transactions and the use of

mobile technologies increased the ability of financial institutions to extend national borders and conduct business worldwide. This has forced government agencies and regulatory authorities to develop strict requirements for the e-banking sector (Shah and Clarke, 2009).

Particularly, a series of technology-oriented legislative acts, such as the Electronic Transactions Act, Sarbanes-Oxley Act (SOX), Truth in Lending Act (TILA), Equal Credit Opportunity Act (ECOA) and Electronic Fund Transfer Act, have been enacted to provide security for all electronic transactions. In addition, the Payment Card Industry Standard (PCI) aim to enhance payment card data security and the Basel Committee on Banking Supervision (Basel Committee on Banking Supervision, 2003) sets clear guidelines for the risk management in e-banking. Considering that the majority of regulations is often overlapping in scope and appears complementary in nature, in the following bullets we describe compliance challenges for e-banking that can increase customer loyalty, expand business and save costs when managed properly.

- **Anti-money laundering**. This is a process by which criminals use e-banking to hide the criminal origin of processes and reduce the risk of being detected by the authorities. This term is closely related to terrorist financing and this process normally involves three stages: placement, layering and integration. Placement is the process of inserting the proceeds of crime into the e-banking whereas layering is moving money around with the use of e-banking to hide its origin. Integration is the method of paying back laundered funds to criminals. The primary risk mitigation approach for anti-money laundering is the "Know Your Customer (KYC)" concept. The Basel committee on e-banking has prescribed guidelines on KYC such as

    - Customer acceptance policy
    - Customer identification
    - Record maintenance

- **Privacy of customer information**. This is a cornerstone for the success or failure of banks and reflects the long term success of a bank. Misuse or unauthorized disclosure of confidential customer data may expose a bank to reputational and legal risks and cause systemic risk. In the e-commerce environment in general and particularly in e-banking, there is always the risk of breach in the privacy of customer information, denial of service, hacking and other errors that usually derive from software and hardware failures.

- **Dependence on service providers**. An additional challenge is the increased reliance on service providers to support e-banking services. This makes almost impossible for a bank to maintain full control of customer information within their own computer network and databases due to data transfers. This causes

- **Personnel issues.** Safe and sound e-banking transactions require skilled personnel to suit the use of technology. In this respect, crucial elements to support personnel integrity and performance are considered a sound risk management process to ensure the integrity and measures to minimize staff turnover. From one hand, a risk management process is vital to maintain and monitor staff operation and apply measures to counter frauds from operational risk. From the other hand, a high staff turnover may disrupt workflow, diminish the quality of service and increase training costs. It follows that clearly defined duties and responsibilities along with personnel policies and practices promote smooth and continuous operations.

- **Technical issues.** It is obvious that the reliability and completeness of information platforms that support e-banking services is one of major concerns in managing operational and compliance issues. The degree a financial institution standardizes its information infrastructure is a management responsibility, requires standardization and measureable controls. Deficiencies in design, maintenance and development constitute prime causes for the manifestation of compliance risk. In such a technology-intensive environment, such as e-banking, constant changes to transaction processes, controls and staff, require an effective change management function. This function should allow for up to date patches, business continuity and controls to reduce risks associated with human or technical error.

- **Segregation of duties.** Operational, reputational, legal and other risks can be managed through internal audits. A popular method of internal audits to ensure that compliance is attained is the segregation of duties (SoD). Common practices to establish segregation of duties include the following guidelines.

    - Information systems designed to conduct electronic transactions should prevent employee or outsourced service provider to enter, authorize or modify a transaction

    - Segregation should be maintained between those developing the web page content and those administrating e-banking systems. The same applies for staff responsible for the information infrastructure and those developing compensating controls

The migration from traditional to electronic banking has brought benefits to both consumers and financial institutions. However, this transformation raises compliance concerns that may be the source of multiple risks. Above all, it is customer's responsibility to take appropriate security precautions against personal data

disclosure. Towards this endeavour, customer education through security awareness programs is recommended as means to enhance customer confidence and improve the compliance function.

2.4.3 Literature review

The purpose of this literature review is to document compliance studies in enterprise environments and examine how compliance is affected. Ifinedo (2012) studied how information systems security policy (ISSP) compliance is affected by the theory of planned behaviour and the protection motivation theory. The study builds hypothesis to test which factors affect ISSP compliance and found, based on a survey of 124 experts and analysis with the partial least squares technique, that factors such as a) self-efficacy, b) attitude towards compliance, c) subjective norms, d) response efficacy and e) perceived vulnerability, have a positively influence on ISSP behavioural compliance intentions of employees.

The same author in a complementary study (Ifinedo, 2014) investigated on employees' ISSP compliance behavioural intentions within organizations from the perspective of social bonding, influence and cognitive processing. The author, based on the theory of planned behaviour, social cognitive and social bond theory uses hypothesis to test how the socio-organizational factors affect the individual's attitude toward ISSP compliance and subjective norms. Results indicate that social influence and the individual's perceptions of their control and competence with regard to IS security issues have a positive effect on ISSP compliance behaviours.

Another study suggests that a higher understanding in organizational power implies better compliance with security rules (Kolkowska et al., 2013). Authors, based on an empirical qualitative study conducted in a Swedish social service organization, analyze the influence of different dimensions of organization power on complying with security policies. Specifically, it is emphasized how the power of resources, processes and meaning influence actions, awareness and values towards security rules as means to induce strategic changes.

Vance et al. (2012) has been motivated from the employee's failure to comply with IS security rules to examine, on the basis of socio-cognitive theories such as the protection motivation theory and habit theory, how past behaviours influence decision making towards compliance. Authors based on results from a qualitative research on a Finnish municipal organization conclude that there is a strong relationship between past and future behaviour towards compliance.

Harris and Furnell (2012), studied on the potential effect of peer-shaming when compliance fails. Based on a scenario-based survey, authors build compliance and non compliance levels to gather responses about behaviour towards compliance. Results show that the relationships between co-workers and the life outside the organization receive the same influence on compliance as inside working conditions.

Moreover, von Solms (2005) examined the difference in terminologies used to describe governance and compliance. The author introduces the term Information Security Governance (ISG) as a broad term that contains IT and corporate governance. This term is focused to preserve the security properties (confidentiality, integrity, availability) at all times. Authors support that compliance and IT should have distinct departments, due to different functions and in order to achieve good governance.

Gragido and Pirc (2011) made a research on how regulatory compliance has deteriorated the status of information security. Authors based on an analysis of international standards such as ISO, PCI, HIPAA and FERPA, suggest that cyber criminals are aware of what has to be protected; therefore regulatory compliance may act as a silent killer against information security properties with keeping update the security invaders with new policies and rules that should exist for an "ideal" information security status.

Vroom and von Solms (2004) emphasises on the role of auditing and related terminology towards information security behavioural compliance. Authors focus on the audit of the behaviour of the employee, showing interest in human factor and organizational behaviour and culture. It is concluded that policing the behaviour, following audit guidelines from COSO and COBIT, helps change the business culture into a more mature and security-oriented.

Moreover, Borrett (2013) highlights the human factor and the need to manage change in behaviours. The author suggests that the security status should be reassessed regularly and communicated in simple terms to users and management. Compliance through policies should limit the change in human behaviour in order to improve the corporate security environment.

Bunbury (2009) argued that it is more efficient to build and use a risk-based security model which will focus on risk assessment rather than a compliance-based model which is not usually followed. The highest risk derives from the human factor and especially from the staff. Monitoring and auditing the network should be priority as it is regarded by the author the weakest areas in the majority of organizations.

Fitzgerald (2011) examines on frameworks and standards towards compliance as means to recommend compliance strategies for companies that can comply with regulatory rules and save costs at the same time. In this regard, this study analyzes the main characteristics of most reputed compliance standards such as COSO, COBIT, ITIL, and ISO 17799 to propose an "11-Factor Security Compliance Manifesto" as best practice towards compliance.

Cook (2012) approaches the concept of compliance from a third party perspective. The study focuses on the supply chain of an enterprise such as customer house brokers, distributors, agents, legal firms, carriers, consultants and other entities that play part in the business. The author emphasizes on the fact that if an enterprise is to operate globally, should conform to a set of rules that respect international business and this notion should be followed by the service provider.

Le Grand (2013) suggests that new technologies can keep an enterprise updated and competitive and that the corporate governance should direct the strategy for secure use of technology. The author claims that the upper management, such as the Board of Directors is responsible to guide the management of risk throughout the enterprise and realize the benefits or shortcomings of compliance.

Farrell (2010) concentrated on the governance, risk and compliance discipline (GRC) from a cloud perspective. In this respect, the author outlines the security issues that exist in a cloud environment and suggest that even if the GRC concept remains problematic, it will be the one that will minimize risk. Key policy factors that need to be taken into consideration is a) research, b) privacy and security, c) access to technology, d) e-government, e) intellectual property, f) electronic surveillance and e) consumer protection. Thereby, the cloud adopter should prioritize related GRC issues before the cloud implementation takes place.

2.5 Audit

2.5.1 Introduction

One of the fundamentals of GRC effectiveness is the role of the audit function. Business and IT professionals are under pressure from regulators, stakeholders and external auditors to assure that GRC activities are performed as appropriate and that this task should be accomplished with an eye on sustainability. For this reason, we describe the role of internal and external audit in subsections 2.5.2 and 2.5.3 respectively. Moreover, we present a literature review that describes various studies on the audit function with emphasis placed on the internal audit.

2.5.2 Internal audit

The audit function is a universal activity that aims to add value to existing operational activities and ensure that the GRC concept functions as intended. Considering the differences in applying audits in enterprise environment, the internal audit depends on factors such as industry, regulations and organizational structure. The Institute of Internal Auditors (IIA, 2012), an official body on the development and evolution of internal audit, supports the following activities:

- Recommend best practices of internal audit

- Provides a general framework which reflects the value of internal audit
- Supports the measurement of internal audit activities
- Encourages proper management of the organizational processes

According to IIA (2009), internal audit is defined as "an independent, objective assurance and consulting activity designed to add value and improve an organization's operations. It helps an organization accomplish its objectives by bringing a systematic, disciplined approach to evaluate and improve the effectiveness of risk management, control, and governance processes". In addition, internal audit can be defined as an independent, objective assurance and consulting activity designed to help the enterprise accomplish its objectives by bringing a systematic, disciplined approach to evaluate and improve the effectiveness of three processes, namely risk management, control and governance processes (IIA, 2000; Spira and Page, 2003). This definition creates the need for a strong GRC discipline (Humphreys, 2008; Karagiorgos, 2010; Tarantino, 2008).

The importance of internal audit lies on the organization and evaluation of the enterprise performance in terms of financial, information security and operational issues. Specifically, internal audit can be described as an umbrella term which defines best practices in terms of governing operations, complying with laws and regulations, preventing the realization of errors and attacks by monitoring actions and report findings to the management. Another role of internal audit is to verify whether transaction accounts have been entered correctly in the accounting books and whether there are irregularities in the accounting records.

Internal audit usually involves an array of analytical auditing procedures that intend to examine and compare relationships among both financial and nonfinancial information (Internal Auditors Statement on Internal Auditing Standards No. 8). A sound example of an analytical auditing procedure is to statistically estimate future financial and non financial risk based on a number of variables. Such variables include but are not limited to technological, operational and environmental changes, staff turnover, governance, compliance and stakeholders requirements and controls efficiency.

Moreover, internal audit analytical procedures can develop projections about current and future events enabling comparison of results and assessment of existing conditions. The benefits of having an internal audit function are summarized in the next bullets:

- Enables the examination of single cases as part of enterprise management
- Identifies redundancies and recommends proper use of resources with an eye on sustainability and performance

- Unifies different departments towards an integrated approach to risk mitigation, recovery from failures, governance and compliance requirements.

- Allows for greater synergy towards cost savings and controls optimization.

- Improves communication between personnel and board of management by establishing smooth internal communication mechanisms.

Complementary to internal audit, is the role of external audit which aims to verity, and in some cases provide a type of certification, about whether internal audit performs adequate controls over enterprise activities in terms of compliance, governance and risk management.

2.5.3 External audit

To support the internal audit function (or cover its deficiency), the external audit function comes into place. External audit can be defined as an independent, third party, audit activity, which is performed to confirm that the internal audit is optimized and to provide a type of verification or certification of compliance (Chartered Institute of Management Accountants, 2006). The same source indicates three types of external audit namely a) statutory audit, b) management of audit guidelines and c) consultancy.

The first type refers to the auditor's role to perform independent audit activities as means to assess whether the enterprise environment complies with certain requirements. The second type refers to supervisory role of external audit in terms of reporting and communication to senior management. Consultancy refers to the role of consulting senior management about best practices and trends in an area of expertise as means to develop strategic perspectives.

In fact, the GRC concept has broadened the role of audit into a dual role that considers different audit functions. For this reason, there are the internal auditors that focus on evaluating the effectiveness of control mechanics and assure compliance with business and security requirements and there are the external auditors that contribute by providing feedback on the effectiveness of this process. This relationship between internal and external audit is not only responsible for the effectiveness of the internal audit but also shapes the risk environment of the enterprise by affecting decision making. The professional competence of external audit should complement the internal audit and ensure compatibility between those two functions (Johnson and Goetz, 2007; Pink Elephant, 2008).

Examples of third party security consultant are the HISP (Holistic Information Security Practitioner) and PinkVERIFY. HISP promotes a holistic approach to information security management by providing through a security system program certification for security practices such as the ISO 17799 (ISO 27002), ISO 27001, ISO 20000, COBIT, COSO and NIST guidelines. PinkVERIFY is a worldwide

independent assessment program that supports the workflow requirements of IT service management tool that certifies Information Technology Infrastructure Library (ITIL) compatibility with operational security processes. This program is developed from Pink Elephant which consists of a group of qualified and experienced IT management consultants who provide external audit services. Overall, the benefits of external audit can be summarized in the next bullets:

- Ensures consistency of internal operations with strategic goals
- Certifies the level of compliance
- Verifies security objectives
- Provides consulting services to senior management
- Evaluates the adequacy and reliability of internal control systems
- Reviews controls' effectiveness and monitoring procedures

It is important to notice that the benefits of adapting an internal and external audit function may be numerous, however, there are hidden drawbacks when the two audits combine functions. Below, we describe the major challenges the combination of internal and external audits may hidden.

- **Resource demanding.** Having two types of auditors assessing the enterprise environment and preserving records is money, resource and time consuming.
- **Outdated reporting.** Due to the historic nature of reporting, the time the audit functions require to communicate and agree on providing records to the management, such as annual audit reports, information may not be as relevant as the time of the audit took place. This fact may impair decision making.
- **Sampling base.** Audits require a large size of information for consistent and reliable results.
- **Human nature.** Due to the instability and bias of human behaviour, from preparation of accounting records to testing of controls and reporting, human behaviour is prone to errors.
- **Audit nature.** Internal and external audits make propositions based on evidence collected. This evidence is of persuasive nature rather than conclusive, which may lead to wrong conclusions.
- **Learning curve.** Hiring an external auditor and building a cooperative but independent relationship with internal audit, requires investment in time and learning experience.

- **Privacy.** The external audit to function properly requires access to confidential information. This may put private information at risk, even in the case of a confidential agreement, because data need to circulate from internal to external audit and back.

2.5.5 Literature review

This literature review presents studies on the factors affecting the internal audit function fulfill its purpose and the relationship of it with risk management. Particularly, Wang and Li (2011) examined the relationship between risk management and audit from an Analytical Hierarchy Process (AHP). Authors suggest that an internal audit function is vital for the risk management function and helps improves inconsistencies and core competitiveness. Authors do not clarify the function of risk management within the audit function however it is claimed that an audit function does help in risk management.

Moreover, Knechel (2007) presents the history of audit, placing emphasis on the structure of the audit function and the relationship with risk management. The author argues that the opportunistic behaviour by auditors combined with the lack of professionalism, the systemic failure of the corporate governance system and stakeholders' demands led to inefficient internal audit processes over time. The author suggests the business risk audit function as a strategic process that places internal audit to the client's systems dynamics.

In addition, Sarens et al., (2009), based on the comfort theory, examined the drivers of internal audit function. The authors, based on four Belgian case studies, outline that internal auditors seek comfort in inter-personal and behavioural skills as well as specialized knowledge in risk management. This comfort is enhanced via collaboration between internal and external auditing.

Research on internal audit, examined the relationship of internal audit and information security functions (Steinbart et al., 2012). Authors conducted a set of structured interviews at four organizations in the education industry and suggested that auditor's technical knowledge, communication skills, auditors' attitude and top management support are the factors that affect the relationship between the two functions.

Abdolmohammadi and Ross (2010) examined the factors that play role by the internal audit functions (IAFs) on information technology (IT) audits. Based on multivariate regression analysis in a sample of 1029 chief audit executives, authors discovered that four variables, namely the certified information system auditor (CISA) certification, IAF age, training, and the number of organizational employees are significantly and positively associated with IT audits by the IAFs. Interesting is the fact that factors

such as the educational level of the auditor, the country of residence and other certifications do not affect the internal audit function.

Another research examined the effects of internal audit in terms of reporting on fraud risk assessments when the level of fraud risk varies (Norman et al., 2010). Based on survey of 172 internal auditors and using ANCOVA analysis, authors found that when fraud risk is high, internal auditors receive more personal threats when they report directly to the audit committee compared to the top management which leads to reduce assessed risk levels when reporting. The same research found also that fraud risk assessment decomposition has different appealing to internal and external auditors.

Recent research on internal audit has linked its function with ethics and the moral role of auditors (Nickell and Roberts, 2013). Authors connected the Everett and Tremblay's analysis of ethics with Nils Brunsson's model of organized hypocrisy in order to propose a series of organizational outputs - talk, decision and action - which can help the internal auditor build a professional profile and expand on corporate governance and risk management.

Stoel et al. (2012), conducted a factor analysis to discover which factor has the greatest impact on the quality of an IT audit process. The top five factors are a) adequate audit plan, b) ethical standards among team members, c) communication skills, d) sufficient resources and e) risk-based audit approach within the audit plan. Interesting is to see that audit certifications, such as CISA, did not receive high score. The research concluded the difference in perceptions between IT and financial auditing members.

Finally, research on internal audit focused on how audit fees vary during reformation of accounting standards (Zhu and Sun, 2012). Based on a survey in China, authors found that a reformation of an accounting standard creates market risk which in turn creates audit risk. This leads to an increase in audit fees among audit professionals.

2.6 Chapter summary

In this chapter, the GRC concept is broken down to its components. Governance is liable to produce multiple benefits for an enterprise based on integrated organizational behaviour. It may also produce inconsistencies if the governance program is not properly utilized. Risk management is the foundation of establishing processes for managing a variety of enterprise risks, such as market and outsourcing risk. The availability of IT risk management methods complicates rather than simplifies the risk management selection process. Compliance is a predominant factor in the success of the GRC concept. Indispensable part of the GRC effectiveness is the audit function, however, inconsistencies may arise if harmonization between external and internal audit is not achieved.

"The price of light is less than the cost of darkness"
Arthur C. Nielsen

Chapter 3

3. Security metrics

3.1 Introduction

In modern society, the requirements for automated security metrics are increasing. Especially in the case of GRC, this concept can be implemented more smoothly and monitored more efficiently when automated security metrics are in place to support isomorphic data analysis. This chapter presents and analyzes a series of specifications from the Security Content Automation Protocol (SCAP) and similar, defined as "SCAP-like" specifications. The distinguishing feature is that specifications identified as SCAP-like do not belong within the SCAP family as in the time of press of this dissertation. Such specifications can aid the management of security content automatically and integrate business, government and compliance issues at an enterprise or sector level. In this chapter we are looking for security metrics that have the following requirements:

- Support a unified treatment of risk variables, as is often the case of isomorphism
- Measurement formulas are available to the public
- Incorporate quantitative metrics
- Consist of automated and standardized processes
- Communicate with a common language (syntax and semantics).

3.2 SCAP specifications

Firstly, we present the evolution of SCAP versions and secondly we provide a description of SCAP specifications. Specifically, SCAP started as 1.0 version which consisted of six components. The latest version is 1.2, which was finalized in 2011 with the addition of five components. In Table 9, the evolution of SCAP and upgrades for each specification is presented.

Version	**SCAP 1.0**	**SCAP 1.1**	**SCAP 1.2**
Year	**2009**	**2010**	**2011**
Specifications	CVE	CVE	CVE
	CCE 5.0	CCE 5.0	CCE 5.0
	CPE 2.2	CPE 2.2	CPE 2.3
	XCCDF 1.1.4	XCCDF 1.1.4	XCCDF 1.2
	OVAL 5.4	OVAL 5.8	OVAL 5.10
	CVSS 2.0	CVSS 2.0	CVSS 2.0
		OCIL 2.0	OCIL 2.0
			CCSS 1.0
			ARF 1.1
			AI 1.1
			TMSAD 1.0

Table 9. SCAP evolution.

In the following, specifications from SCAP 1.2 version (Waltermire et al., 2011) are illustrated. The aim is to increase interoperability among enterprise information systems and manage security content homogeneously. Roussey et al., (2010), defines interoperability as the information systems' ability to share information and other applications. On this basis, SCAP allows information exchange, from asset identification to policy checking, in a uniform and automated process. Using SCAP, an enterprise is liable to relate data on a one-to-one relationship among enterprise IT systems and thereby, accelerate data analysis.

CVE (Common Vulnerabilities and Exposures)

Most software attacks are the result of exploiting vulnerabilities in software products, which are essential for the operation of any company, organization or even for the security of a nation. In addition, vulnerabilities in software can endanger the intellectual property, consumer confidence and business operations and services. In recent years, the number of vulnerabilities has increased significantly as the level of sophistication which attackers use their techniques, enabling them to gain access to often highly sensitive information. Many experts on security issues and organizations, such as the SANS, provide tips and best practices for risk prevention and safety, including setting up systems with safety regulations, protection of sensitive data through encryption and removing security vulnerabilities in software.

Particularly, information security is often seen as a fast-pace race between hackers that try to exploit vulnerabilities in an information system and vendors and risk practitioners that try to cover from such vulnerabilities by applying patches and updates. Therefore, CVE has emerged as an official vulnerability identifier, supported by the MITRE and found in the National Vulnerability Database (NVD) vulnerability database. A CVE identifier contain for each vulnerability a) an identity (e.g. CVE-2003-0818), b) a standard description for a vulnerability (e.g. default password enables remote command execution) or an exposure (e.g. improper settings in an operating system), and c) references to other standards (e.g. OVAL - ID). The CVE list is dictionary of information security vulnerabilities and aims to provide common names for publicly known problems. Hence, CVE is as follows.

- A name for one vulnerability
- A standardized description for each vulnerability
- A dictionary rather than a database
- The way in which different databases and tools can "speak" the same language
- A basis for assessing software tools and databases
- The basis in which vulnerability severity is scored

When a new vulnerability is publicly announced, a new CVE identifier is created to represent the vulnerability and CVSS base attributes are computed and added in the NVD. The CVE specification identifies two types of vulnerabilities: entries and candidates. Entries are the vulnerabilities that have been published to the CVE list, and candidates the vulnerabilities under review for the CVE list. Overall, this specification distinguishes and enumerates vulnerabilities and is available without any restrictions.

CCE (Common Configuration Enumeration)

An effort similar to the CVE is the CCE specification. Particularly, CCE assigns a unique, common identifier with an associated “configuration guidance statement” and “configuration control.” The first specifies required settings or policies for the information system under testing (e.g. the required permissions for the directory System32\Setup should be assigned to the “Administrator account” only.) A configuration control describes a control unit referring to the conceptual security model of an information system such as the access permissions for files and directories such as System32\Setup in Win32 Libraries.

Currently, the focus of CCE is on software products configuration. Each configuration issue gets an identity consisting of a) a number (e.g. CCE-1234-123), b) a description

(e.g. operating system) c) conceptual parameters for its implementation (e.g. specifications and settings, d) technical mechanisms for a given configuration issue (e.g. availability and download of an update), e) references to reports, tools or documents that support in more detail the configuration issue under consideration (e.g. OVAL - ID). In the following Figure 5, an example of CCE is presented.

Example CCE:

- CCE-3260-7
- Definition: The "Log Dropped Packets" option for the Windows Firewall should be configured correctly for the Domain Profile.
- Parameter: Enabled or Disabled.
- Technical Mechanisms:
  - HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\Domain Profile\Logging\LogDroppedPackets
  - Computer Configuration\Administrative Templates\Network\Network Connections\Windows Firewall\Domain Profile\Windows Firewall: Allow Logging - Log Dropped Packets
  - Control Panel\Windows Firewall\Advanced\Security Logging\Logging Options\ Log dropped packets

Figure 5. CCE example (Source: Mann, 2008)

The creation of CCE has motivated five different cases of use in managing security settings in an information infrastructure. The most important use case of CCE is the internal lifecycle management settings. This case can be described as a process that consists of four parts: a) the design of the system before installing the testing, b) development, c) evaluation and d) rehabilitation. The remaining four cases using CCE can be regarded as "rays" from the earlier described case. These four use cases include: system design, the control configuration, audit report and compliance. A common theme among all these cases is to use the CCE to facilitate faster and more accurate correlation of information for some settings between different, but so closely related practices. Overall, CCE specification provides unique identifiers to security-related system configuration issues for various software and hardware products in order to improve correlation of configuration data across the enterprise.

CPE (Common Platform Enumeration)

Identifying and describing vulnerabilities and configuration settings may seem worthless if it not related to a particular IT platform. For this reason, the CPE specification emerged to improve on the standardization of IT platforms. Therefore, CPE describes the characteristics of an IT platform (hardware, software and applications). The aim of CPE is to foster automation towards identification of the IT platforms to which a vulnerability or element of guidance applies.

The most common is version 2.2 (Buttner, 2009), a specification which is included in SCAP 1.0 and 1.1 versions. The CPE format describes platforms into specific fields, such as part, vendor, product, version, update, edition and language. CPE evolved into

the 2.3 version, part of SCAP 1.2 version, with main differentiation the deployment of four additional, edition-related, fields (Cheikes et al., 2011). A CPE product dictionary represents a list of official CPE names and is provided to the public and supported by the NVD (National Vulnerability Database, 2013), a U.S. government vulnerability data resource. By using CPE names to distinguish IT platforms, provides a uniform way to classify IT platforms with a common format. Current use of this standard is limited to naming software (e.g., vendor, title, version). In the following Figure 6, components of a CPE name are presented.

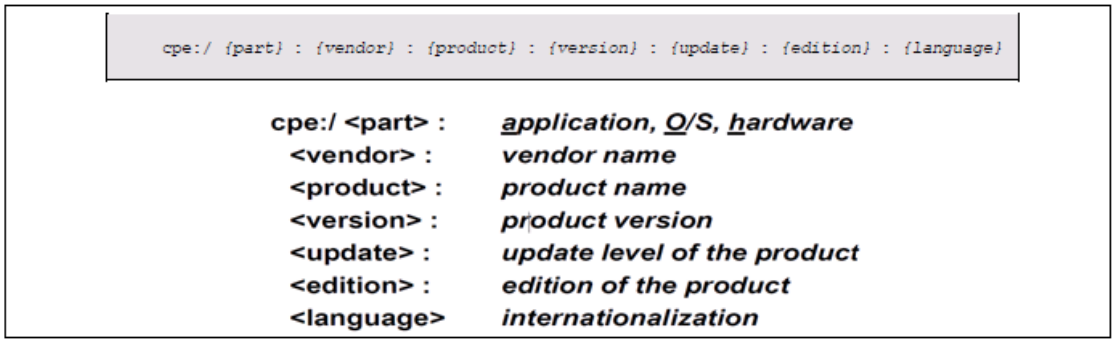

Figure 6. Components of a CPE name (Source: Cheikes et al., 2011)

Specifically, a CPE name is a unique collection of components given to a specific platform type, such as <cpe://microsoft:windows:2000>. Using CPE brings the following benefits: a) eliminates huge discrepancies in the IT industry on how a software or hardware will be named or a version will be referred, b) people have no difficulty in interpreting different names that exist for commercial use for each product (e.g. Microsoft Windows 2000, Win2K, Windows 5.0) and c) achieves security automation by standardizing the naming scheme for each software platform.

Moreover, CPE supports its own dictionary as the official collection of CPE names. Its purpose is to provide a source of all known CPE Names, descriptions of these names and various diagnostic tests. Overall, CPE is a structured naming scheme for Information Technology (IT) systems, platforms and packages and contributes by provides a set of identifiers and dictionary for platform or product naming.

CVSS (Common Vulnerability Scoring System)

The CVSS is a standardized scoring system for vulnerabilities. Its purpose is to highlight what vulnerability is the most important with a view to give priority to addressing the most critical vulnerabilities. Specifically, it features three measurement sets: a) Base (obligatory) which describes the fundamental characteristics of vulnerability that are constant over time, b) Temporal (optional) which describes characteristics that vary in time and c) Environmental (optional) for characteristics

that are relevant to the user interface. In the following Figure 7, CVSS flow chart and metrics for each metric group are presented.

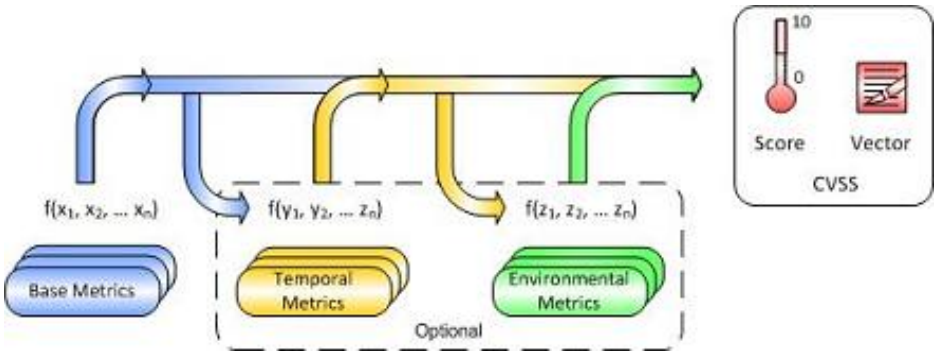

Figure 7. CVSS flow chart (Source: Mell et al., 2007)

For each metric group, there is a vulnerability severity formula. The basic formula, that can act as a stand alone vulnerability severity formula is the one based on the base metric group. This formula produces a CVSS score as a decimal number ranging from 0 to 10 where the value 0 means vulnerability has negligible severity whereas 10 means vulnerability has critical severity.

Formulas from temporal and Environmental metric groups are used to update the Base formula scoring. At the time of press of this dissertation, there are two versions of CVSS, v1 and v2, whereas v3 is under development. Here, we describe CVSS v2 and accompanied metrics. Therefore, in the following Figure 8, CVSS v2 metric groups are depicted.

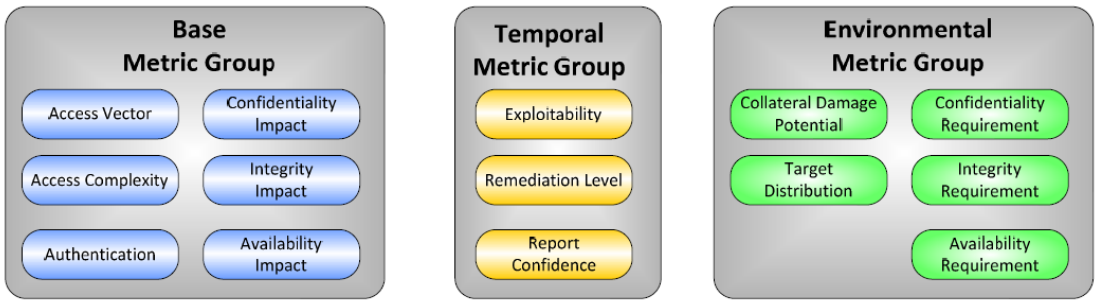

Figure 8. CVSS v2 metric groups (Source: Mell et al., 2007)

The Base metric group consists of i) exploitability sub-score composed of access vector (EV), access complexity (EC) and authentication (EU) metrics, and ii) impact sub-score composed of confidentiality (IC), integrity (II) and availability (IA) metrics. The EV metric measures how someone took advantage of a vulnerability or how the attack happened. The more remote an attacker, the higher the rating score. The EC

metric measures the complexity of an attack. The higher the attack complexity, the lower the rating value. The EA metric measures the number of times an attacker must authenticate to a target system to exploit vulnerability. The IC, II and IA metrics measure the impact on confidentiality, integrity and availability respectively. EV is measured in terms of local (*lo*), adjacent network (*ad*) or network access (*ne*) values; EC metric is measured in terms of low (*lw*), medium (*me*) or high (*hi*) values; EU metric is measured in terms of none (*no*), single (*si*) or multiple (*mu*) values. The base impact metrics (IC, II, IA) are all measured under none ($nn_{c,i,a}$), partial ($pa_{c,i,a}$) or complete ($co_{c,i,a}$) values. Each metric value has a corresponding rating value. In Table 10, CVSS Base metric group is presented.

Metric name	Metric values	Rating values
EV	lo, ad, ne	$0.395, 0.646, 1$
EC	hi, me, lw	$0.35, 0.61, 0.71$
EU	mu, si, no	$0.45, 0.56, 0.704$
IC	nn_c, pa_c, co_c	$0, 0.275, 0.660$
II	nn_i, pa_i, co_i	$0, 0.275, 0.660$
IA	nn_a, pa_a, co_a	$0, 0.275, 0.660$

Table 10. CVSS v2 Base metric group (Source: Chatzipoulidis et al., 2015b)

The Base metric group severity formula is described in the following formulas:

$$BaseScore = round\_to\_1\_decimal(((0.6 * Impact) + (0.4 * Exploitability) - 1.5) * f(Impact)) \quad (1)$$

$$Exploitability = 20 * EV * EC * EU \quad (2)$$

$$Impact = 10.41 * (1 - (1 - IC) * (1 - II) * (1 - IA)) \quad (3)$$

$$f(Impact) = 0 \; if \; Impact = 0, \; 1.176 \; otherwise \quad (4)$$

The Temporal metric group consists of three metrics. The Exploitability metric measures the current state of exploit techniques or code availability. The Remediation

Level measures the existing level of remediation of vulnerability. The less official and permanent a fix, the higher the rating value is. The Report Confidence measures the degree of confidence in the existence of vulnerability and the reliability of the technical detail. Note that the temporal score will produce a score no higher than the base score, and no less than 33% lower than the base score. In Figure 9, CVSS Temporal metric group is presented.

```
TemporalScore = round_to_1_decimal(BaseScore*Exploitability
                *RemediationLevel*ReportConfidence)

Exploitability    = case Exploitability of
                         unproven:              0.85
                         proof-of-concept:      0.9
                         functional:            0.95
                         high:                  1.00
                         not defined:           1.00

RemediationLevel = case RemediationLevel of
                         official-fix:          0.87
                         temporary-fix:         0.90
                         workaround:            0.95
                         unavailable:           1.00
                         not defined:           1.00

ReportConfidence = case ReportConfidence of
                         unconfirmed:           0.90
                         uncorroborated:        0.95
                         confirmed:             1.00
                         not defined:           1.00
```

Figure 9. CVSS Temporal metric group (Source: Mell et al., 2007)

The Environmental metric group consists of three metrics. The Collateral Damage Potential measures the potential for loss of life or physical assets through damage or theft of property or equipment. The Target Distribution measures the number of vulnerable systems that will be affected by the exploitation of vulnerability. The final metric corresponds to security properties and enables the analyst to customize CVSS scores depending on which security property (confidentiality, integrity and availability) is more important.

Note that the environmental equation, if employed will combine the environmental metrics with the temporal score to produce an environmental score ranging from 0 (negligible) to 10 (critical) vulnerability severity. The Environmental equation will produce a score no higher than the temporal score. In Figure 10, the Environmental metric group formula, metric and rating values are presented.

```
EnvironmentalScore = round_to_1_decimal((AdjustedTemporal+
(10-AdjustedTemporal)*CollateralDamagePotential)*TargetDistribution)

AdjustedTemporal = TemporalScore recomputed with the BaseScore's Impact sub-
equation replaced with the AdjustedImpact equation

AdjustedImpact = min(10,10.41*(1-(1-ConfImpact*ConfReq)*(1-IntegImpact*IntegReq)
                 *(1-AvailImpact*AvailReq)))

CollateralDamagePotential = case CollateralDamagePotential of
                                 none:            0
                                 low:             0.1
                                 low-medium:      0.3
                                 medium-high:     0.4
                                 high:            0.5
                                 not defined:     0

TargetDistribution        = case TargetDistribution of
                                 none:            0
                                 low:             0.25
                                 medium:          0.75
                                 high:            1.00
                                 not defined:     1.00

ConfReq          = case ConfReq of
                        low:              0.5
                        medium:           1.0
                        high:             1.51
                        not defined:      1.0

IntegReq         = case IntegReq of
                        low:              0.5
                        medium:           1.0
                        high:             1.51
                        not defined:      1.0

AvailReq         = case AvailReq of
                        low:              0.5
                        medium:           1.0
                        high:             1.51
                        not defined:      1.0
```

Figure 10. CVSS Environmental metric group (Mell et al., 2007)

Overall, CVSS is a standardized vulnerability scoring system that enables the analyst to score individual vulnerabilities. Values from the Base metric group are available for the vulnerabilities identified under the CVE specification in the NVD. Temporal and Environmental metric groups are optional to calculate.

XCCDF (Extensible Configuration Checklist Description Format)

An XCCDF document is a structured collection of rules for system configuration in the form of security checklists that aim to support automated policy compliance testing and scoring. The purpose of XCCDF is to provide a uniform basis for the expression of security checklists and other configuration guidance, and therefore trying to promote the application of good security practices.

The XCCDF created to document the technical and non-technical security lists using a standard form. Each list is an organized collection of rules for a particular system or platform, captured in a XML specification language and is used to examine whether or not a system is vulnerable to a particular type of attack.

The primary audience of the XCCDF specification is government and the secondary audience is industry security analysts and product developers. The use of XCCDF is mainly technical security checklists which can reduce the vulnerability exposure of a system. Specifically, XCCDF goals are to generate documentation, express policy-aware configuration rules, support complex systems that may require complex rules, support compliance scoring and customization, and perform as a vulnerability scanner.

Additional features provided by the XCCDF are as follows: a) ensure compliance with multiple policies (e.g. Federal Information Security Management Act, the Security Technical Implementation Guide, Health Insurance Portability and Accountability Act, and others), b) allow faster and more automated definition and implementation of safety standards, procedures, guidance, warnings, recommendations and remedial measures, c) allow the unified management of security controls, d) allow the combination of safety rules and tests from different groups and vendors and e) Rate the condition of the test system and enable reporting and monitoring of the security situation and compliance with security policies. In Figure 11, the XCCDF function is presented.

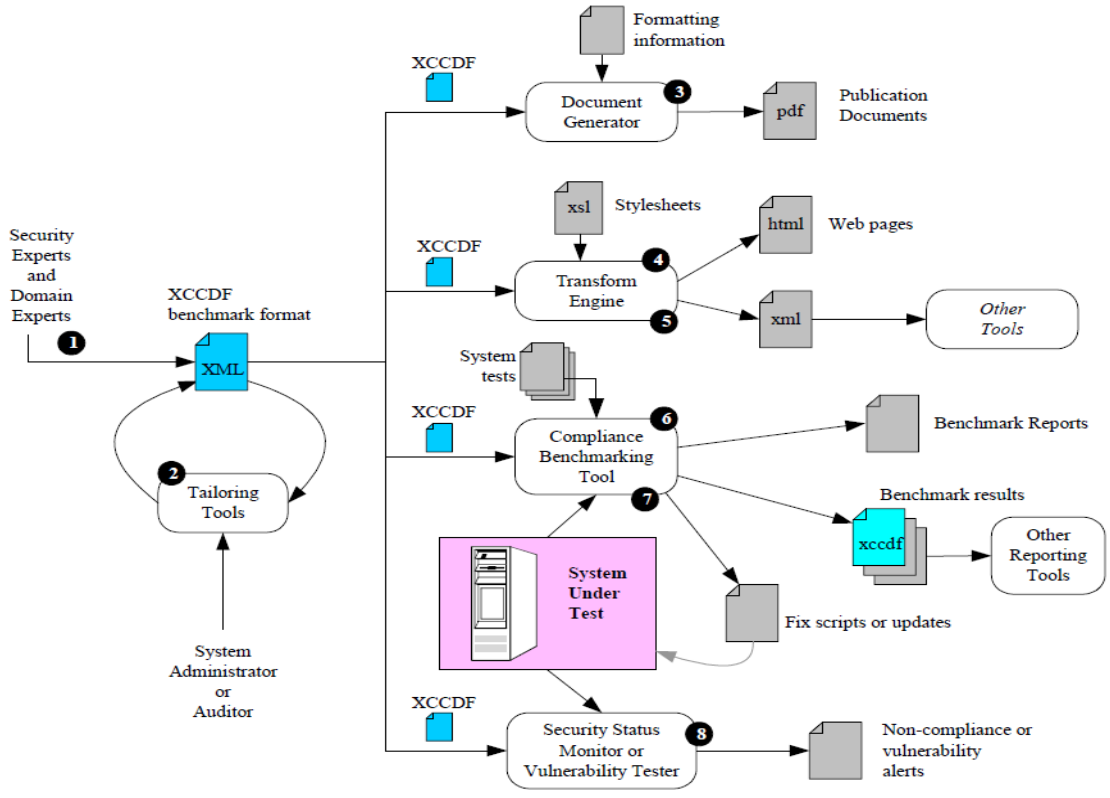

Figure 11. XCCDF function (Source: Ziring, 2005)

In Figure 11, the points of interest are 1,2,6,7 and 8. At first, various specialists in safety produce the so-called best practices (1), then the XCCDF document is created (2) which performs benchmarks for compliance under the test system with best

practices (6 and 7) and produces a report containing the results of the benchmarking exercises. Finally through XCCDF and evaluations we can see if the evaluated system suffers from a vulnerability (8).

XCCDF cooperates with OVAL in order to perform tests to IT platforms from a target system. Specifically, the XCCDF contain rules based on which a system is compliant with a security policy, while the OVAL contains the tests which will check whether the examined system follows a specific security policy or if it suffers from a vulnerability. In the following Figure 12, XCCDF and OVAL interoperability is presented.

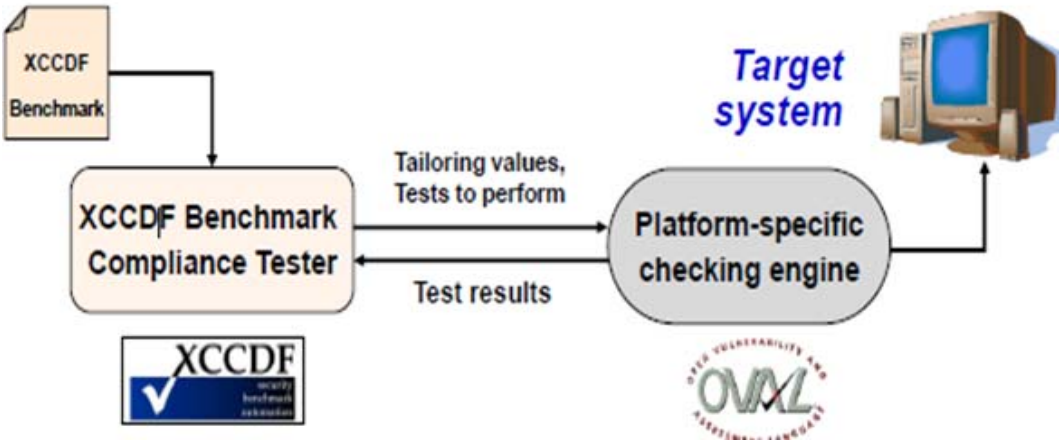

Figure 12. XCCDF and OVAL interoperability (Source: Waltermire et al., 2011)

Overall, the objective of XCCDF is to enable security analysts and IT specialists create efficient, interoperable and automated checklists and support the use of these lists with a wide variety of tools.

OVAL (Open Vulnerability and Assessment Language)

OVAL is a method for performing structured tests for reporting purposes. It supports, homogenizes and transfers data across the entire spectrum of security tools and services. OVAL includes a language (in XML format) which is used to encode the details of the system and has a safety profile which collects the data that exist in the community of intelligence. The aim is to conduct tests to check if the values of a system (e.g. registry key) satisfy the security policies of XCCDF.

OVAL uses a language (in XML format) for storing system configuration information in local systems and its actual use is similar to a common risk assessment process namely a) identify and collect configuration data of the system under test (OVAL System Characteristics), b) analyze the system security incidents (OVAL Definition schema) and c) document and report the final results about the state of a system

(OVAL Results schema). OVAL definitions are posted under a unique identifier (OVAL-ID).

Version 5.10.1 released on 20 January 2012 and version 5.11 is in the planning phase. The oldest version of OVAL that exists in the website of MITRE, is version 3. This version dates back in June 2004 and relates to Microsoft Windows, Sun Solaris, Red Hat and Linux operating systems. The OVAL today covers a wide range of operating systems, network components and some web or applications servers. The community contributes to the development of OVAL language by participating in the creation of the OVAL language, the OVAL Forum Developers, and writing definitions for OVAL Repositories via the OVAL Forum.

An OVAL Board consisting of representatives from a broad spectrum of industry, academia and government organizations from around the world who oversee the OVAL Language and monitor the definitions posted and hosted on the site of OVAL. This means that the OVAL, which is funded by US-CERT at the US Department of Homeland Security, reflects the expertise that is delivered through the leading professionals in the field of safety and management systems worldwide. In the following Figure 13, the OVAL flow chart is presented.

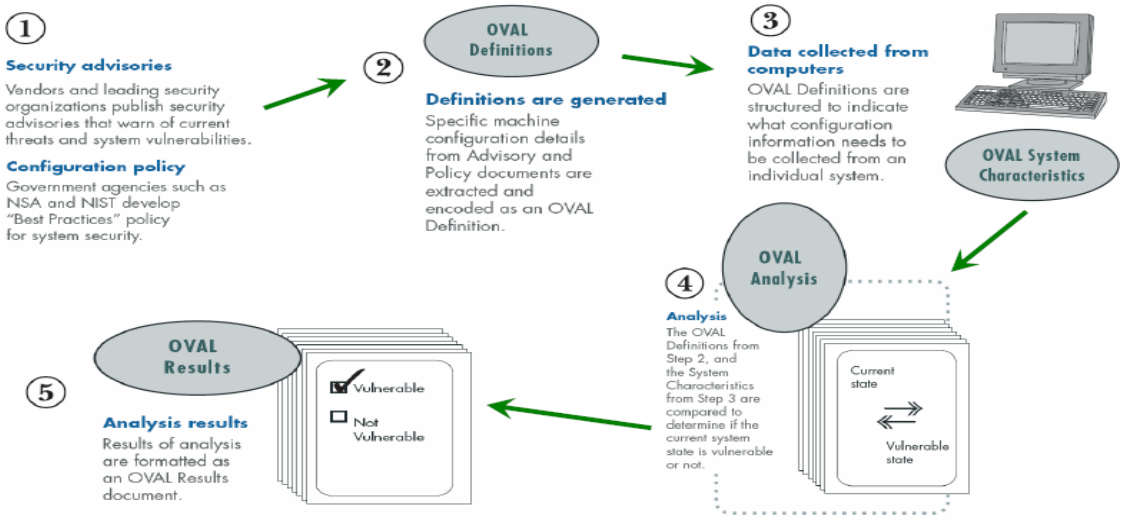

Figure 13. OVAL flow chart (Source: MITRE, 2006)

Overall, the benefits of using OVAL can be summarized as a) through OVAL can be determined if there is vulnerability or an issue in system configuration security settings or if a patch is needed in the system, b) provides a schema that describes the necessary security that must exist in the system and information about the relevant settings, c) through the XML encodes the exact details of a particular safety concern and d) it is backed and updated by a community of experts on security, system administrators and software developers.

CCSS (Common Configuration Scoring System)

A similar scoring system to CVSS is the CCSS which differentiates from CVSS in that it scores not the vulnerability severity but software configuration issues. Considering that the majority of configuration settings have the capability to increase security at the expense of reducing usefulness and operability, it follows that the most secure configuration setting may not provide the best solution.

Therefore, CCSS can assist organizations in making correct decisions as to how they should be treated the security configuration issues (security configuration issues) and provide data used in quantitative assessments of the overall security of the information system. CCSS uses the same scoring range as the CVSS where 0 means negligible severity and 10 critical severity of the configuration setting under evaluation. However, there are also significant differences in the details of the specifications.

The CCSS measurements are organized in three groups: Base, Temporal, and Environmental. The Base measurements describe the characteristics of a configuration problem that remains stable over time in the user's environment. The Temporal measurements describe the characteristics of the configuration issues that may change over time but remain constant in the user's environment. The Environmental measurements are used to adjust the Base and Temporal scores based on the characteristics of a particular user interface. In the following Figures 14, 15 and 16, show how the Base, Temporal and Environmental results are calculated from the three measurement groups.

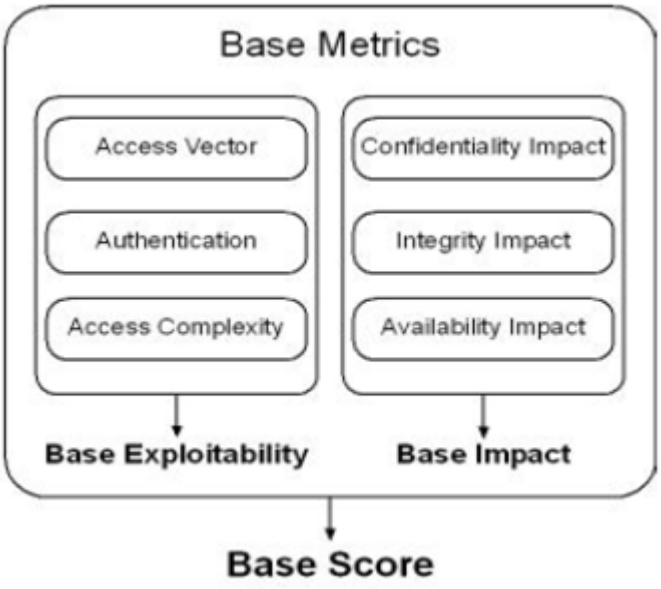

Figure 14. Base metric group (Scarfone and Mell, 2010)

The Base metric group counts two main aspects of affect vulnerability severity: exploitability (Exploitability) and Impact (Impact). The CCSS Base metrics are the same as the CVSS with the only difference being the measurement of Exploitation Method in the measurement of Access Vector, Authentication and Access Complexity added. The Exploitability Method refers to the types of vulnerabilities caused by misadjusted configuration settings in two ways either Active (A) or Passive (P). Based on the type of vulnerability (Exploitability Method), Exploitability metrics are scored and based on this scoring CCSS scores are generated.

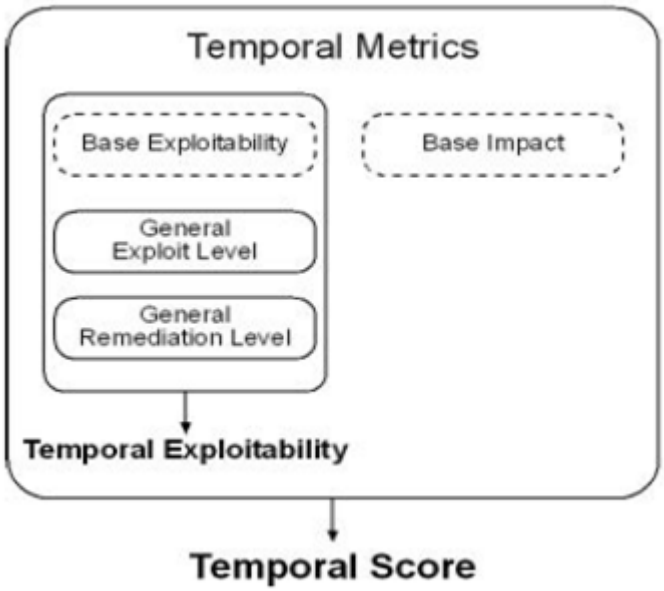

Figure 15. Temporal metric group (Scarfone and Mell, 2010)

The Temporal metric group describes the characteristics of the security configurations that can change over time but remain constant in the user's environment. The temporal measurements are not required to calculate, such as the Base, and contain two measurements: a) General Exploit Level (GEL) and b) General Remediation Level (GRL). The first measures the prevalence of attacks against security configuration and the second measures the availability of remedies that can mitigate the vulnerability without changing or disabling some features of the configuration settings.

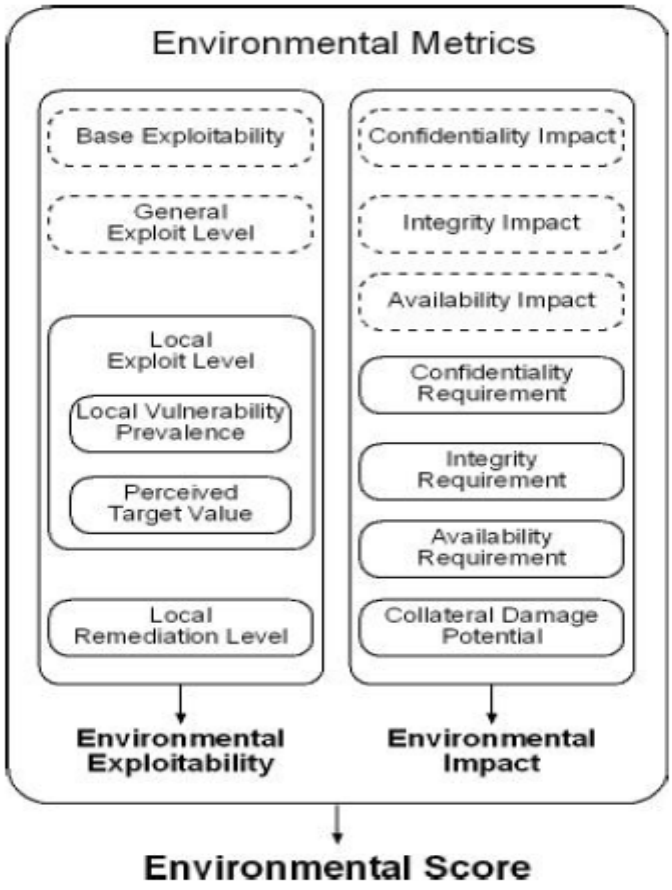

Figure 16. Environmental metrics group (Source: Scarfone and Mell, 2010)

The differences between the environments can have a significant effect on the risk that a vulnerability can pose to a particular organization and its parts. In this respect, the Environmental metric group measures the characteristics of vulnerability related to a specific IT environment by adjusting the Base metric values. Particularly, the Environmental metric group measures three aspects: Local Exploit Level (LEL), Local Remediation Level (LRL) και Local Impact (LI).

As seen in Figure 13, LEL is composed on two metrics: Local Vulnerability Prevalence (LVP) and Perceived Target Value (PTV). LVP measures the prevalence of vulnerable systems in a specific environment and the PTV measures the probability of an attack exploiting a security issue in an environment in relation to vulnerable systems in other environments. The LRL measures the level of protection against a vulnerability in the local IT environment and determines how widespread and effective the mitigation application is.

The LI consists of seven environmental metrics. The first three (Environment Confidentiality Impact, Environment Integrity Impact and Environment Availability Impact) take the place of the corresponding measurement of the Base Impact (Confidentiality Impact, Integrity Impact and Availability Impact). The fourth, Collateral Damage Potential (CDP) increases the three measurements of environmental impact. The remaining three Environmental metrics (Confidentiality

Requirement, Integrity Requirement and Availability Requirement) are used to calculate the weights applicable to the three metrics of environmental impact. The CDP and the Confidentiality, Integrity, Availability Requirements (CR, IR, AR) are exactly the same as the CVSS. The Environment Confidentiality, Integrity and Availability Impact metrics allow the analyst to adapt the Environmental score if the benefits to the environment differ considerably from best practices associated with the vulnerability.

In the following Figures 17 and 18, the Temporal and Environmental characteristics in terms of formula, metric values and rating values are presented respectively. Note, that the Base formula, metric and rating values are the same as in the Base metric group of CVSS.

```
TemporalScore = round_to_1_decimal(((0.6 * Impact) + (0.4 * TemporalExploitability) – 1.5) * f(Impact))

TemporalExploitability = min(10, Exploitability * GeneralExploitLevel * GeneralRemediationLevel)

GeneralExploitLevel = case GeneralExploitLevel of
                none: 0.6
                low: 0.8
                medium: 1.0
                high: 1.2
                not defined: 1.0

GeneralRemediationLevel = case GeneralRemediationLevel of
                none: 1.0
                low: 0.8
                medium: 0.6
                high: 0.4
                not defined: 1.0
```

Figure 17. Temporal metric group characteristics (Source: Scarfone and Mell, 2010)

```
EnvironmentalScore = round_to_1_decimal(((0.6 * EnvironmentalImpact) + (0.4 * EnvironmentalExploitability) –1.5) *
f(Impact))

EnvironmentalImpact = min(10, 10.41*(1 - (1 - EnvConfImpact * ConfReq) * (1 - EnvIntegImpact * IntegReq) *
(1 - EnvAvailImpact * AvailReq)) * CollateralDamagePotential)

EnvironmentalExploitability = min(10, Exploitability * GeneralExploitLevel * LocalExploitLevel *
LocalRemediationLevel)

LocalExploitLevel = LocalVulnerabilityPrevalence * PerceivedTargetValue

EnvConfImpact = case EnvironmentConfidentialityImpact of
                    none: 0.0
                    partial: 0.275
                    complete: 0.660
                    not defined: ConfImpact
EnvIntegImpact = case EnvironmentIntegrityImpact of
                    none: 0.0
                    partial: 0.275
                    complete: 0.660
                    not defined: IntegImpact
EnvAvailImpact = case EnvironmentAvailabilityImpact of
                    none: 0.0
                    partial: 0.275
                    complete: 0.660
                    not defined: AvailImpact

ConfReq = case ConfReq of
                    low: 0.5
                    medium: 1.0
                    high: 1.51
                    not defined: 1.0
IntegReq = case IntegReq of
                    low: 0.5
                    medium: 1.0
                    high: 1.51
                    not defined: 1.0
AvailReq = case AvailReq of
                    low: 0.5
                    medium: 1.0
                    high: 1.51
                    not defined: 1.0

CollateralDamagePotential = case CollateralDamagePotential of
                    none: 1.0
                    low: 1.25
                    low-medium: 1.5
                    medium-high: 1.75
                    high: 2.0
                    not defined: 1.0

LocalVulnerabilityPrevalence = case LocalVulnerabilityPrevalence of
                    none: 0.6
                    low: 0.8
                    medium: 1.0
                    high: 1.2
                    not defined: 1.0

PerceivedTargetValue = case PerceivedTargetValue of
                    low: 0.8
                    medium: 1.0
                    high: 1.2
                    not defined: 1.0

LocalRemediationLevel = case LocalRemediationLevel of
                    none: 1.0
                    low: 0.8
                    medium: 0.6
                    high: 0.4
                    not defined: 1.0
```

Figure 18. Environmental metric group characteristics (Source: Scarfone and Mell, 2010)

Overall, CCSS was developed to measure the severity of security software configuration issues due to imperfections of software. The CCSS can assist organizations in making correct decisions as to how they should be treated the security configuration issues (security configuration issues) and provide data used in

quantitative assessments of the overall security of the system. CCSS can work in conjuction with the Common Configuration Enumeration (CCE) specification.

OCIL (Open Checklist Interactive Language)

Compliance checking may sometimes be quite complex to be automated. In this effort, OCIL based on an XML language, aims to support manually compliance checking in case automated checking is not viable. Particularly, OCIL establishes a framework for the presentation of a set of questions to the user, the procedures for the interpretation of the answers to these questions and aims to develop safety checklists. The difference from XCCDF is that XCCDF supports the automated control compliance whereas OCIL the manual. OCIL describes in the form of questions to the user whether a particular process is in compliance with a set of regulations or policies. OCIL and XCCDF documents are easily communicated via the XML language and OCIL results are usually combined into a single XCCDF report.

For example, we have an XCCDF document that contains controls to be performed for a particular system. However, some of the controls require a complex assessment which can not be automated for some reason. Therefore, the author of the XCCDF document includes a reference to an OCIL document that comprises of a manual control. When the administrator wants to monitor compliance opens the XCCDF document and the OCIL document loads with a series of questions.

The user responses are collected and the results are returned to XCCDF which then continues with the remaining checks. The results of automated and manual controls are combined in a single report by the XCCDF interpreter. Overall, OCIL assists in compliance management through a manual checking in cases when XCCDF can not be used.

ARF (Asset Reporting Format)

The ARF (Asset Reporting Format) is a data model for determining the transport format of information about assets and the relationships between assets and reports. Due to the fact that data about assets exist in various forms (e.g. reports, various formats) and across different locations (e.g. databases, sensors), the need to capture and organized in a unified way data about assets has brought up the need for a data model capable to support correlation of different assets from different sources.

The ARF, in cooperation with Asset Identification (AI) specification, enables the correlation and fusing of information from disparate data sources into a standardized format in order to create a homogenized and comprehensive picture of an asset status. Assets showing conformity with ARF specifications usually makes them more commercial and interoperable.

In Figure 19 the structure of a data model for ARF is presented which consists of four (4) sections such as: a) relationship section, b) report request section, c) asset section,

and d) report section. The purpose of ARF data model is to capture disperse information about assets from various resources within the organization and present it in a unified way.

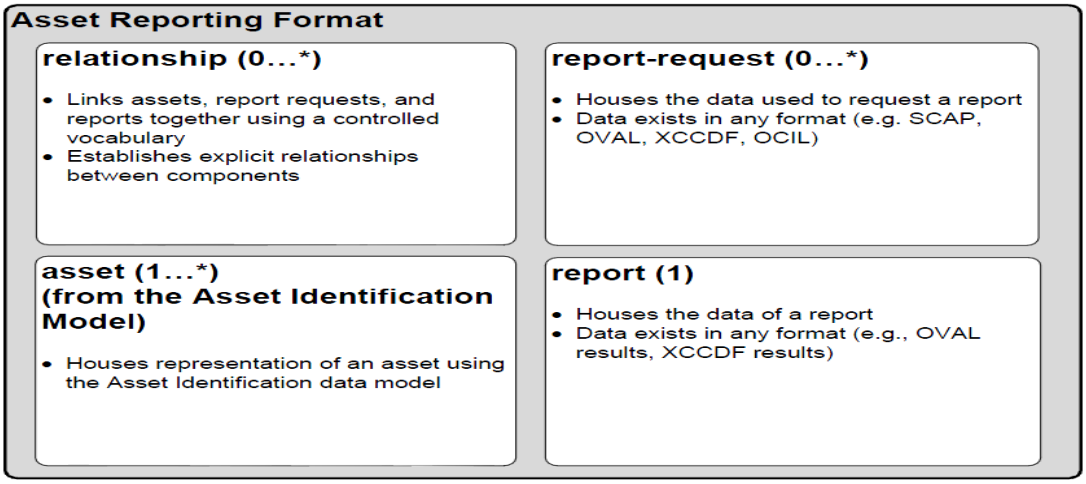

Figure 19. Structure of ARF (Source: Halbardier et al., 2011)

Overall, ARF is a specification that allows for interoperability of data for asset management across the enterprise. Several ARF reports can be generated among different business units which in turn enables deeper understanding of the real-time status of an enterprise.

AI (Asset Identification)

In order to manage assets, first you must be able to identify and define them based on a meaningful set of data. Towards this challenge, the Asset Identification (AI) is a specification which provides a unified approach to asset identification allowing for data correlation from multiple sources, reporting and interoperability with other SCAP specifications.

AI supports many types of identifiers, the most common used are literal identifiers (e.g. MAC address), relationship identifiers (assets dependency), synthetic (during a process) and extension (outside AI vocabulary) identifiers. The purpose of identifiers is, beyond the actual identification of assets, to integrate the assets.

In the following Table 11, the associations between assets that belong to the AI controlled vocabulary are presented. The column “Term” defines the name of term being used. The column “Domain” defines the relationship of a subject in relation to the object which the relationship with the subject is defined in column “Range”. The column “Description” includes a brief description of the relationship type.

Term	Domain	Range	Description
hasTerminationDevice	ai:circuit	ai:computing-device	The circuit is terminated by the device.
hasServiceProvider	ai:circuit	ai:organization	The circuit is owner/operated by the organization.
hasNetworkTerminationPoint	ai:circuit	ai:network	The circuit ends at the network.
servedBy	ai:database, ai:website	ai:service	The database or website is served up by the service.
hasServiceProvider	ai:service	ai:software	The service is provided by the software.
installedOnDevice	ai:software	ai:computing-device	The software is installed on the computing device.
connectedToNetwork	ai:system	ai:network	The system is connected to the network.
isOwnerOf	ai:person, ai:organization	ai:it-asset	The person or organization owns the IT asset.
isAdministratorOf	ai:person	ai:computing-device, ai:system	The person is the system administrator of the computing device or system.
partOf	ai:person	ai:organization	The person is in some way a part of the organization.
connectedTo	ai:computing-device, ai:system	ai:system	The computing device or system is connected to the system.

Table 11. AI controlled vocabulary (Source: Wunder et al., 2011)

Overall, the purpose of AI is an asset data management specification which enables the analyst to identify, define and correlate data for assets across the enterprise in a unified manner. Using AI brings multiple benefits to the enterprise management such as detection of duplicate assets and documentation on the relationships among those assets.

TMSAD (Trust Model for Security Automation Data)

This specification is a trust model which can be used for traceability of results, providing assurance that a set of results derive from a particular source. The TMSAD was developed to provide content integrity and authentication by ensuring that the content has not been altered since it was created. In this way, it aims to support data security automation by processing of XML documents. Conformity with this specification adds value to the security status of an enterprise or a "product".

The latter is defined as a "content-consumer" and "content-author" communication which to be secured must comply with the syntax, structural and other requirements

(e.g. TMSAD algorithms) of TMSAD in order to allow that the content has not been modified. In Figure 20, there is an example of exchanging security automation data using TMSAD. In the left side a content producer signs and exchanges data with a content consumer. The TMSAD is used to address how the content producer will sign security automation data using the private key and how the content consumer will verify this signature using the public key.

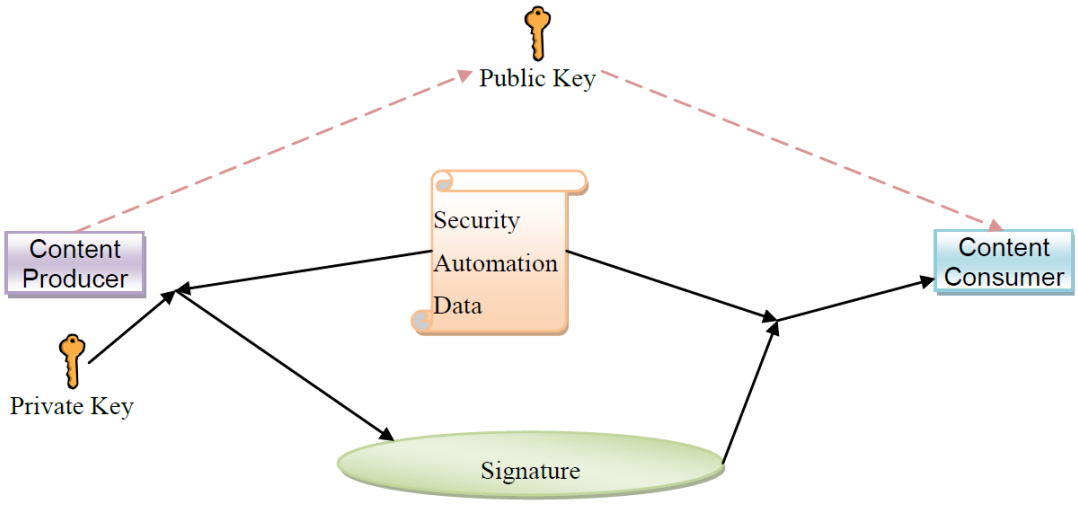

Figure 20. Example of TMSAD (Booth and Halbardie, 2011)

TMSAD uses the W3C recommendation on XML Signature Syntax and Processing. It belongs to the SCAP specification and considering that security automation is exchanged via an XML language, the focus of this model is on the processing of XML documents.

Overall, TMSAD brings multiple benefits not only to the parties that exchange XML documents but also to software vendors that want to verify that their products interoperate with others using the trust model or to an enterprise that is looking to establish security of exchanging data with other enterprises.

3.3 SCAP- like specifications

Parallel to SCAP specifications, there are others that either perform the same role or can work in conjunction with SCAP specifications to aid managing automatically the security content. In Table 12, SCAP-like specifications are presented with their primarily purpose and a description follows.

Purpose	Identification	Vulnerability Scoring	Event Characterization	Reporting/ Data Transport
Specifications	SWIDs	CWSS	MAEC	ASR
	CWE	CWRAF	CybOX	CVRF
	CAPEC	CCSS	STIX	xNAL
		CMSS		

Table 12. SCAP-like specifications

SWIDs (Software Identification Tags)

SWIDs is a software asset management tool which collects data for an asset into tags so as to provide accurate asset identification. SWIDs are files that use XML language and have been primarily developed to identify software applications in terms of name, version, edition, publisher and other characteristics. SWIDs also allow for asset monitoring in order to verify with a common language whether the installed software is updated and not compromised. SWIDs are in accordance with ISO/IEC 19770-2:2009, a standard which refers to software asset management.

SWIDs include the Software Entitlement Tags (SWEN) which intend to measure automatically whether license for software applications has been expired in order to provide assurance to the enterprise that the installed software is optimized. Currently, there are seven mandatory elements of software identification such as Entitlement Required, Product Title, Product Version, Software Creator, Software Licensor, Software Unique ID and Tag Creator. In addition to the mandatory elements, one can use more that thirty optional elements such as Product Category, Previous product of company names, Licence details and others.

Overall, SWIDs are becoming a must in managing the identification of an enterprise's software applications. If we consider that, beyond identification, SWIDs also optimize expired licensing or forgotten updates, then this tool appears a complementary solution to existing asset identification specifications such as ARF and AI.

CWE (Common Weakness Enumeration)

Similar specification to CVE is the CWE which represents a set of documents that describe, in an integrated way, the weaknesses of software covering a wide spectrum associated with the architecture and design of computing systems. This specification can be used as a guidance to find weaknesses in source code and operational systems. CWE interacts with security taxonomies such as Gramma Tech and research centres in order to provide updated information about new software weaknesses.

CWE is supported by the National Vulnerability Database. Particularly, there is a CWE vulnerability type for every CVE identifier. Currently, CWE supports a list with more than 700 identified weaknesses and more than 30 (34 in the time of press of this dissertation) CWE types. An individual who do not have the luxury of time and want to know the most serious weaknesses for software applications, there is the CWE/SANS Top 25 community consensus list with most widespread and critical CWEs. Individuals who have the time to research on weaknesses for their software applications and want to prioritize them, the Common Weakness Scoring System (CWSS) for scoring CWEs and the Common Weakness Risk Analysis Framework (CWRAF) for prioritization of CWEs with the organization's business mission are available to the public.

Overall, CWE provides a unified language of discovering the causes of software security vulnerabilities as they are found in code, design or system architecture. As in the time of press, CWE is not part of the SCAP family however, the NVD is using CWE as a classification mechanism that differentiates CVEs by the type of vulnerability they represent.

CAPEC (Common Attack Pattern Enumeration and Classification)

Challenged to build secure software, CAPEC is introduced as an inventory of common attack patterns. Supported by the MITRE, CAPEC describes attack methods and is designed to categorize the mechanism of attacks into categories. As in the time of press, the CAPEC 2.6v is available that supports 463 attack patterns, however, the top tier has the following eleven categories: Abuse of Functionality, Spoofing, Probabilistic Techniques, Exploitation of Authentication, Resource Depletion, Exploitation of Privilege/Trust, Injection, Data Structure Attacks, Data Leakage Attacks, Resource Manipulation and Time and State Attacks. These attack patterns are mechanisms that capture and communicate the attacker's perspective or descriptions of common methods for exploiting software. Each attack pattern is identified with a unique identification number along with a proper description.

CAPEC allow interoperability by combining its inventory with the Common Weakness enumeration (CWE) as means of weakness relationship type. CAPEC also provides information about attack prerequisites and resources required in order an attack to succeed. An organization being "CAPEC-compatible" means that for the management of software assets (including Web sites, databases, or other security products or services) uses CAPEC names in a way that allows it to be cross-referenced with other products that employ CAPEC names. Overall, the purpose of CAPEC is to provide an updated list of patterns employed by attackers when compromising systems using specific real-world exploit examples.

CWSS (Common Weakness Scoring System)

This is a specification for scoring software weaknesses. This scoring system aims to prioritize CWEs and support multiple usage scenarios by different stakeholders. The

total score is calculated based on 18 factors that constitute for the three metric groups such as: a) the Base Finding group captures the inherent characteristics of the weakness, confidence in the accuracy of the finding, and strength of controls, b) the Attack Surface group refers to the nature of the attack exploiting the weakness and c) the Environmental group refers to factors that apply for a certain operational environment. In the following Figure 21, the three metric groups and accompanied factors are presented.

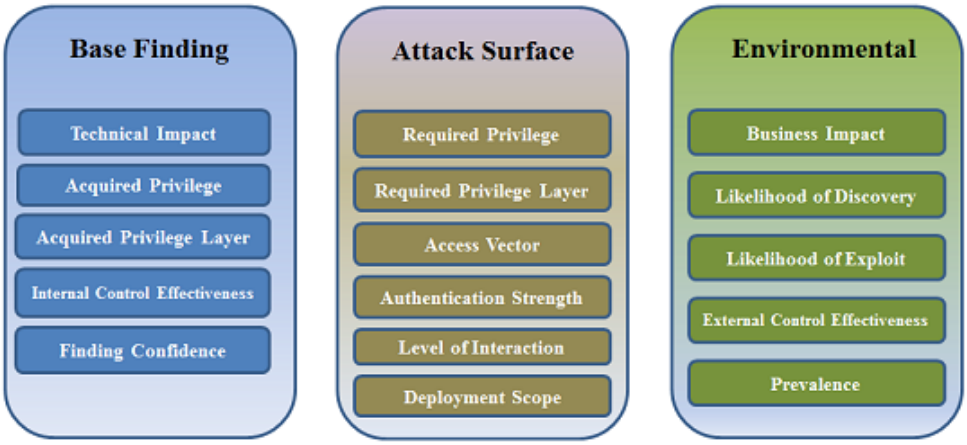

Figure 21. CWSS metric groups (Martin and Coley, 2014)

First, each factor in the Base Finding metric group is assigned a value based on the profile of the weakness. These values are converted to associated weights and the Base Finding subscore is computed. The same method is applied to the Attack Surface and Environmental metric group. In terms of scoring values, the Base Finding subscore range between 0 and 100 and the Attack Surface and Environmental group subscores range between 0 and 1. Finally, the three subscores are multiplied together, which produces a CWSS score between 0 and 100. For more information about the characteristics of factors and formulas for each CWSS metric group can be found in (Martin and Coley, 2014).

CWSS can be a great tool for software developers, acquirers, security managers and other stakeholders that show interest in scoring and prioritizing software weaknesses that could introduce risks to software products, applications and networks. Unlike CVSS, CWSS scoring ranges between 0 and 100 and is calculated by multiplying the subscores from each metric group. Overall, CWSS offers quantitative measurements of weaknesses identified with the CWE specification, provides a common framework for prioritizing weakness based on specific factors and in conjunction with the Common Weakness Risk Analysis Framework (CWRAF), CWSS allows stakeholders to identify types of weaknesses and correlate them with protection activities within the enterprise environment towards achieving software assurance.

CWRAF (Common Weakness Risk Analysis Framework)

CWRAF is a framework for scoring software weaknesses from various business domains and works in conjunction with CWE and CWSS. This specification supports the automatic prioritization of weaknesses according to an organization's business profile and also allows ranking weaknesses separate from software packages in order to indicate which weakness is more dangerous from another for an organization. With the use of vignettes, which is a semi-formal description of a scenario that defines a set of connected technology groups that perform a function within a business domain, CWRAF assesses the weaknesses inherent in an operational environment, the security requirements and the role that software plays within a particular organization.

The benefits of applying CWRAF are the following: a) a standardized mechanism for measuring the severity of weaknesses in a way that resembles an enterprise's mission b) allows for automation and customization of weaknesses scoring according to the needs of the enterprise and c) interoperates with CWSS and CWE as means to achieve software assurance. In Figure 22, the relationship of CWRAF, CWSS and CWE is presented.

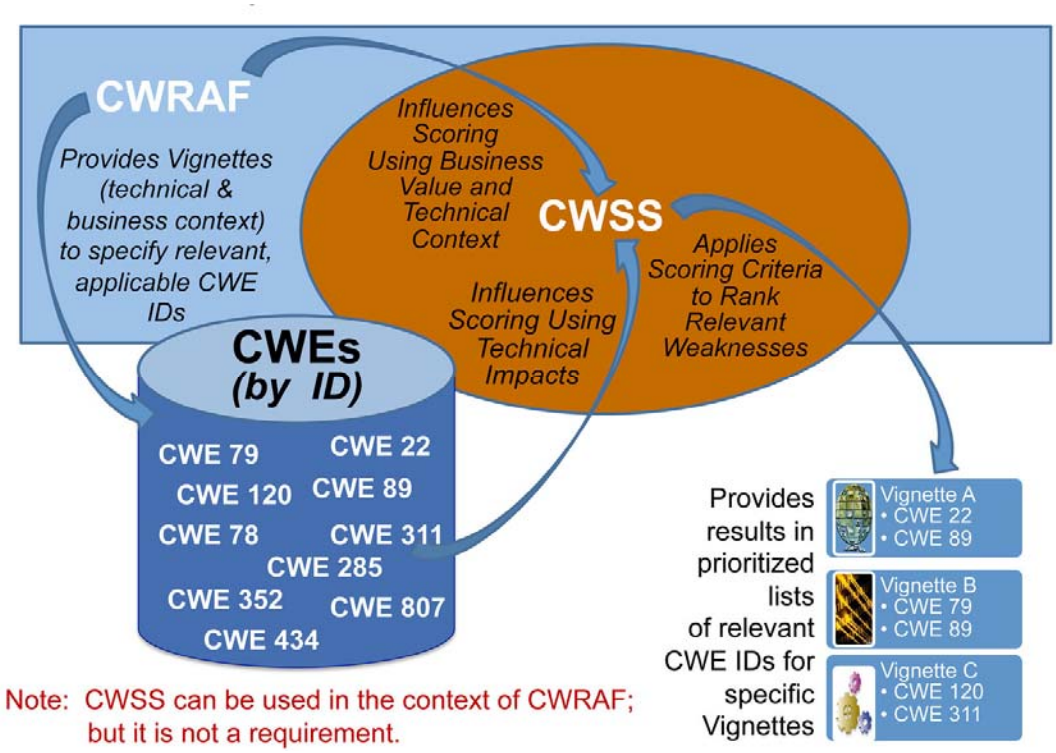

Figure 22. CWSS metric groups (Source: Martin, 2013)

Overall, CWRAF is an open framework which improves on the management and prioritization of weaknesses. CWSS and CWE are independent of CWRAF however they allow interoperability which improves on identifying weaknesses relevant to the business environment.

MAEC (Malware Attribute Enumeration and Characterization)

MAEC represents a standardized language for encoding and communicating malicious software (malware) (Kirillov et al., 2010). By offering a formal language on malware descriptions such as attack patterns and artifacts, it helps avoid duplicate data during malware analysis and also improves controls' effectiveness towards malware detection, prevention and mitigation. Moreover, the adoption of MAEC offers a) elimination of ambiguity in malware description by offering a common method of characterizing malware, b) improved organizational awareness related to malware handling and c) greater efficiency of controls related to malware management.

MAEC supports a common vocabulary and grammar for the malware domain and is used to characterize attributes of software and provide data for malware intrusion detection and assessment. Particularly, MAEC use cases assists in the fields of incident management, cyber threat analysis and intrusion detection. MAEC identifies and collects characteristics of a malware object by capturing with a dynamic or static way the total set of attributes that surround it. The collection of malware objects and characteristics are reserved in a container where they are grouped based characteristics and candidate indicators. The latter is an entity which signifies the presence of the malware instance on a host system or network. Overall, this specification allows for a unified malware management and by supporting different use cases it improves on the analysis of cyber threats and can be used in conjunction with CybOX (Cyber Observable eXpression).

CybOX (Cyber Observable eXpression)

This specification represents a standardized language for events' characterization (Barnum, 2012). It supports various use cases that respond to the following processes:

- Event management
- Attack detection
- Incident response management
- Threat characterization
- Security testing
- Digital forensics
- Information sharing

- Malware analysis
- Cyber situational analysis

CybOX allows interoperability with STIX, CAPEC, DFXML and MAEC to support communication of cyber observables across the entire operational environment in a unified and automated way. Cyber observables are specific descriptions of characteristics for an entity or event in a cyber environment such as a UNIX file or a Windows registry key. In the time of press, the latest version is 2.1 with highlights the extended support and the addition of more detailed characteristics about cyber observables. Overall, CybOX is an open community specification which enables the description of observable events occurring at the operational cyber environment of an enterprise. By supporting a common structured language, it helps organizations identify, collect and communicate cyber observables.

STIX (Structured Threat Information eXpression)

STIX is a standardized language which aims to capture and communicate cyber threat events. STIX expresses with detailed characteristics a cyber attack such as indicators (e.g. IP address), specific threat data (e.g. attacker tactics) and exploitation targets (e.g. computer system). STIX is supported by users and developers that share a standardized language to define cyber threat information. STIX language consists of 8 key elements that are depicted in the following Figure 23.

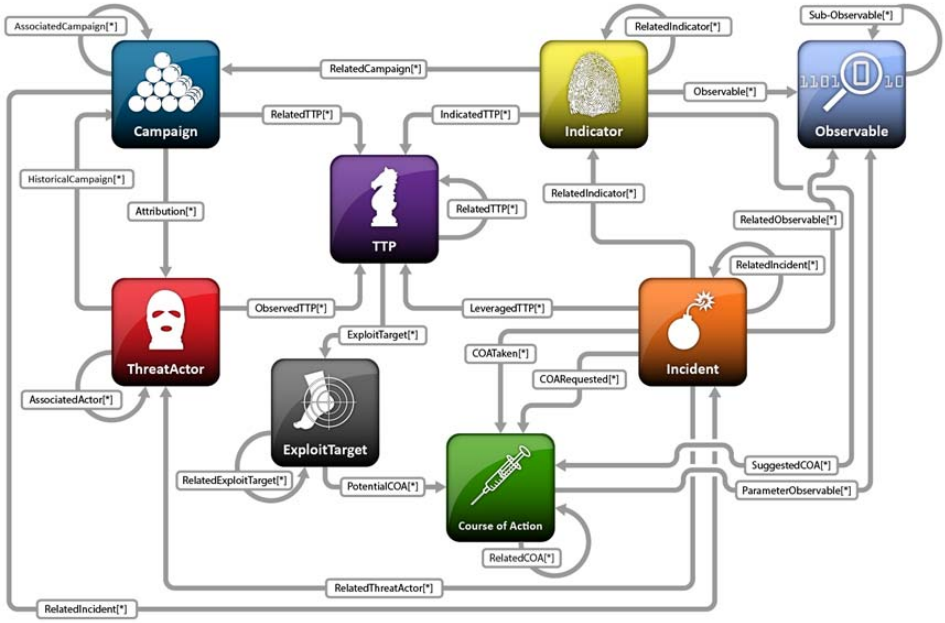

Figure 23. STIX work flow (Barnum, 2014)

- Observable refers to what has happened or might happen in the cyber environment
- Indicators refer to artifacts and behaviours of interest within the cyber security context
- Incidents refer to specific adversary actions
- TTP (Adversary Tactics, Techniques, and Procedures) refers to attack patterns, exploit techniques and other methods used by the adversary
- Exploit Targets refers to vulnerabilities, weaknesses, configurations and malware that might exist
- Course of Action refers to mitigation actions in response to an attack
- Campaign refers to a set of TTPs with shared characteristics
- Threat Actor refer to identification and description of the adversary

Overall, STIX is an open-source, community specification that uses a common language to automate the management of cyber adversaries within the cyber environment of an enterprise. STIX allows interoperation with other similar specifications such as CybOX and MAEC.

ASR (Assessment Summary Results)

ASR is similar in scope with the ARF, a SCAP 1.2v specification. As a result, ASR has the purpose of asset management that based on structured language exchanges summarized information from assessment tools about the variety of IT assets (Wunder and Baker, 2010). This automation in data exchange further improves the process of collecting and analyzing data to create a unified asset database for further analysis. The structured language in ASR is based on the Policy Language for Assessment Results (PLARR). The latter describes automation interfaces as specifications for data exchange and validation reasons.

ASR allows interoperability with all of SCAP 1.0v specifications. The work flow of ASR is similar to ARF and includes identification of assets, assessment for vulnerabilities, patches and configuration settings and sharing of results. The latter can be attained with OVAL/XCCDF specifications. The audience of ASR is primarily asset assessment managers, software vendors and security information managers. Overall, ASR assists in characterizing an asset within the enterprise environment and reports the results of asset assessment. The asset assessment includes results from vulnerabilities in the form of CVE, configuration items in the form of CCE, platforms in the form of CPE, patches in the form of patch identifiers, assessed state of compliance in the form of XCCDF and rules in the form of OVAL.

CVRF (Common Vulnerability Reporting Framework)

CVRF is specified to standardize in an automated fashion, the communication of vulnerability documentation (Schiffman, 2011). In this regard, it provides collecting and reporting on software product IDs, revision history, document status and other elements. Although there are existing specifications that identify vulnerabilities (e.g. CVE) and rank their severities (CVSS), this lack of standardization in vulnerability reporting, best practice documents or security bulletins released by different vendors created the requirement of a CVRF documents, based on XML language, support a single format of vulnerability reporting, speeding information exchange. Its structure is based on a tree-based, mind map diagram.

CVRF purpose is to gather and structure the data in the form of a report, without attention in the appearance of data to the end user. In this regard, it limits its potential only to pure technical users. Overall, CVRF was developed to fill the gap in the standardizing vulnerability reporting. It enables different stakeholders across different organizations to share security data in a unified way that is familiar to all users. CVRF uses definitions existing in SCAP specifications such as CPE, CWE and OVAL.

CMSS (Common Misuse Scoring System)

There are different ways in which vulnerabilities can be classified. Particularly, one can score vulnerabilities based on a) software flaws in the form of CVSS, b) configuration settings in the form of CCSS and c) software feature misuse in the form of CMSS. The latter is introduced here and refers to a software application that is abused from its intentional use. Software feature misuse vulnerabilities are introduced during the design of the software and allow attackers to abuse the software application from its initial purpose (LeeMay et al., 2012).

In simple terms, CMSS is another vulnerability scoring specification, closely related to the CVSS, CCSS and CWSS. The noticeable difference is that the scoring applies for what is called “software feature misuse vulnerability”. This vulnerability differentiates in nature because it emerges from the software designers’ trust assumptions about the use of a particular software product. For example, a designer who builds e-banking banners for banks newsletters made the trust assumption that the nature of the product could not be distorted by an attacker who can use such banners to attract the user into a malicious site. CMSS scoring system uses the same base, temporal and environmental metric found in CCSS (see Figures 14, 15 and 16) and the formulas are identical. Overall, CMSS takes a different perspective on vulnerability scoring considering misuse software vulnerabilities.

xNAL (eXtensible Name and Address Language)

xNAL uses an XML language to represent person names and addresses (OASIS, 2001). xNAL is governed by the OASIS Customer Information Quality (CIQ) Technical Committee and consists of two sub-standards, the extensible Address Language (xAL) and the extensible Name Language (xNL). Considering that customer data may exist in different formats and in many components, the only reliable identifier is the customer name and address. This name and address, as a data type, can receive multiple variations from country to country. Furthermore, this type of data is fragile because customers come and go, change addresses, names and so on. This fact is exaggerated considering the different cultural contexts of customer names and addresses in a global market. It follows that name and address data vary inconsistently and are prone to errors and discrepancies during integration of files from different business units.

Moreover, address information is completely subjective and heavily dependent on the person who performs the data entry. In this respect, storing the same information in different way creates an annoying duplication of data that is difficult to manage. In this respect, xNAL standardize the way name and address characteristics are described in order to provide a unified solution to collect and share this type of data among enterprises that use this standard. Overall, xNAL supports the interchange of customer data, in terms of names and addresses, using XML language and standardizing the format of this data type. Other standards that perform a similar role to xNAL is the xCIL that relates to a standard related to unify disperse customer information and the xCRL, a standard that defines and exchanges in a common format customer relationships.

3.4 Vulnerability scoring methods

All enterprises rely on information infrastructures to perform business goals. Each information infrastructure relies on a variety of platforms such as hardware, software and applications. Each platform faces multiple risks when a threat exploits vulnerability such as damage or disruption of services. According to Arbaugh et al., (2000) vulnerability is a “technological flaw in an information technology product that has security or survivability implications”. De Ru & Eloff (1996) defines software vulnerability as “a weakness in a system that can be exploited to violate the system’s intended behavior”. Therefore, in the context of security metrics, we present vulnerability scoring methods. The reasons to research in vulnerability scoring methods are the following:

- Accelerating rate of growth
- Increasing attack patterns and sophistication
- Multiple impacts on different enterprise IT systems

- Prioritization of vulnerabilities

Vulnerability scoring methods are divided into three main types: qualitative, quantitative and hybrid. For each type, accompanied methods and characteristics are presented.

3.4.1 Qualitative vulnerability scoring methods

Qualitative vulnerability scoring methods use a Likert-type scale to measure the severity of vulnerability. The actual scoring is based on subjective judgment and experience rather than an actual estimation of vulnerability characteristics and is used to provide an easy to understand result. In the following, qualitative scoring methods are described.

Symantec Security Response Threat Severity Assessment

Symantec (Symantec, 2002) focuses on software bugs (viruses, worms, Trojan horses and macros) which Symantec defines as threats. Based on identification, Symantec classifies such threats into "threat components" to determine the severity rating. The threat components are: a) "in the wild", b) the damage occurred if exploited and c) the distribution of the damage.

The first component measures the extent to which the threat is already spreading among computer users such as the number of affected computers and geographic distribution of infection. The second component measures the potential damage in terms of modified files, performance degradation and compromised security settings. The third component measures how rapidly it spreads such as large-scale email attack (worm), executable code attack (virus) and network drive infection capability.

Each threat component is measured in a scale of High, Medium and Low. When the threat analyst gathers all sub-component scorings then, categorizes each component within five (5) categories, category 5 being the most severe and category 1 the least.

Qualys vulnerability management

Qualys (Qualys, 1999) approach to scoring vulnerability comes with a range of activities such as a) identification of assets, vulnerabilities and threats, b) risk status reporting and c) compliance evaluation with the PCI standard. Specifically, Qualys performs scanning for the network environment of an enterprise via a cloud premise

(e.g. Amazon cloud) and identifies the most critical assets into business units and asset groups.

The criticality is measured in a five tier scale from Low to Critical. Based on NVD, performs identification and correlation of vulnerabilities in terms of CVEs. The highlights of Qualys include the iDefense Threat Intelligence module and the Zero-Day risk analyzer. Both enable to make predictions based on statistical data about zero-day vulnerabilities and allow interoperability with CVSS scoring.

Security Bulletin Severity Rating System of Microsoft

Microsoft (Microsoft, 2012) approach aims to score vulnerability in terms of exploitation. Microsoft associates the identification of a specific vulnerability with a CVE identifier and rates the exploitability on a three tier scale as follows: a) consistent exploit code likely, b) inconsistent code likely and c) functioning exploit code unlikely. Microsoft focus on the latest and older software releases in order to provide an aggregate exploitability assessment. This assessment is provided through an index (Microsoft's Exploitability Index) to customers.

This index, compatible with CVSS since it uses CVE to identify vulnerability, is independent to other vulnerability scoring systems. Microsoft uses a four tier rating for vulnerability severity as follows: a) Low for a vulnerability whose exploitation or impact is low, b) moderate when vulnerability can be remediated via configuration or audit, c) important when it will have an impact on confidentiality, integrity and availability and d) critical when its exploitation could allow the distribution in the network of a Internet worm.

Secunia

Secunia (Secunia, 2002) is well known for the vulnerability advisories that contain vulnerability characteristics categorized by product and vendor. Secunia Advisory is dedicated to be a premier database of vulnerability knowledge which is updated on a daily basis and communicated via personalized e-mails to recipients who want to get acquainted with the latest information about the criticality, patch status and severity of vulnerabilities.

Secunia acknowledges three different types of vulnerabilities based on attack vector. These types are as follows: a) from local system, b) from local network and c) vulnerabilities that can be exploited remotely, without access to a system or local network. Secunia's severity rating is based on five tier scale, from "not critical", such as locally exploitable denial of service, to "extremely critical" such as vulnerabilities that are exploited remotely.

Red Hat severity classification scheme

Red hat (Red Hat, 2005) has released the Red Hat Enterprise Linux 4 which is a vulnerability severity classification scheme. This scheme publicizes vulnerability ratings compared to the first version. Red Hat uses CVEs as means to identify vulnerabilities and supports advisories that are updated regularly to provide users with a tool to assess vulnerabilities based on their network environment.

Red Hat's severity classification scheme has the following characteristics: a) it is based on a technical analysis of the type of vulnerability, b) helps users assess their network environment, c) provides automated update services, via the Red Hat network, as means to minimize the risk emerging from updates and d) it is independent from other vulnerability scoring systems. Red Hat uses a four tier scale from Critical to Low. For example, Red Hat considers as critical those flaws that can be exploited remotely and as low those flaws that will cause, if exploited, minimal consequences.

Mozilla's vulnerability rating system

Mozilla (Mozilla, 2005) supports security advisories for products such as Firefox, Thunderbird and Seamonkey. Each product advisory consists of vulnerabilities that are presented with: a) title, b) impact, c) discovery date, d) reporter and e) products affected. Vulnerabilities are rated in a four tier scale, from Low, which includes vulnerabilities that have minimal consequences, such as denial of service, to Critical, such as vulnerabilities that allow an attacker to exploit and distribute the damage into the whole system.

Google's severity rating system

Google (Google, 2007) supports guidelines for security issues regarding products based on Chromium browser to help vendors rate the severity of vulnerabilities. Google uses a four tier scale as follows: a) Low, rated as Pri-3, such as a bug that allows the attacker to hang the browser, b) Medium, rated as Pri-2, such a bug that allows an attacker to collect limited amount of information, c) High, rated as Pri-1, such as vulnerability exploitation modifies confidential data and d) Critical, rated as Pri-0, such as attackers' success to gain user's privileges during the normal operation of the browser.

Vupen security

The principal aim of Vupen (Vupen, 2005) is to work in conjunction with government agencies and the community to combat 0-day and 1-day vulnerabilities and threats. Due to the increasing rising of most critical 1-day vulnerabilities in commercial programs, such as Adobe Acrobat or Microsoft Internet Explorer, Vupen, based on a series of sophisticated techniques, such as disassembly, reverse engineering, protocol analysis, and code auditing, aims to provide a proactive approach to risk. Vupen uses a four tier scale as follows: a) Low, for locally exploitable flaws which can not put the

system in danger c) Moderate, b) High and d) Critical, for remotely exploitable flaws, which could lead to system failure without user interaction.

3.4.2 Quantitative vulnerability scoring methods

Qualitative vulnerability scoring may have the advantage of interpreting vulnerability severity with a Likert type scale however, qualitative methods lack both the precision of results, due to the fuzziness of rating, and the interoperability with other methods, because the majority use own inventories or advisories. In this respect, when a user requires increased accuracy of results, then, it becomes almost necessity, the need for a quantitative vulnerability scoring method. For these reasons, the most reputed quantitative vulnerability scoring methods including formulas characteristics, terminology used and reporting aspects are presented.

Common Vulnerability Scoring System

CVSS version 1

This is, without doubt, one of the most reputed vulnerability scoring systems, which has inspired other researchers towards vulnerability scoring. Schiffman & CIAG, (2005) introduced CVSS v1 which consists of the base, temporal and environmental metric groups. Base metric group measures the inherent characteristics of vulnerability in order to capture the main profile and then based on temporal and environmental metrics calculation (which is optional), vulnerability severity is updated.

The temporal group contains metrics that evaluate the vulnerability during its lifecycle and hence, are prone to change. The environmental group contains metrics that evaluate the surrounding attributes, such as stakeholder involvement. The CVSS severity score takes values from 0 (vulnerability has no severity) to 10 (vulnerability has maximum severity). In the following Figure 24, CVSS v1 metric groups are presented.

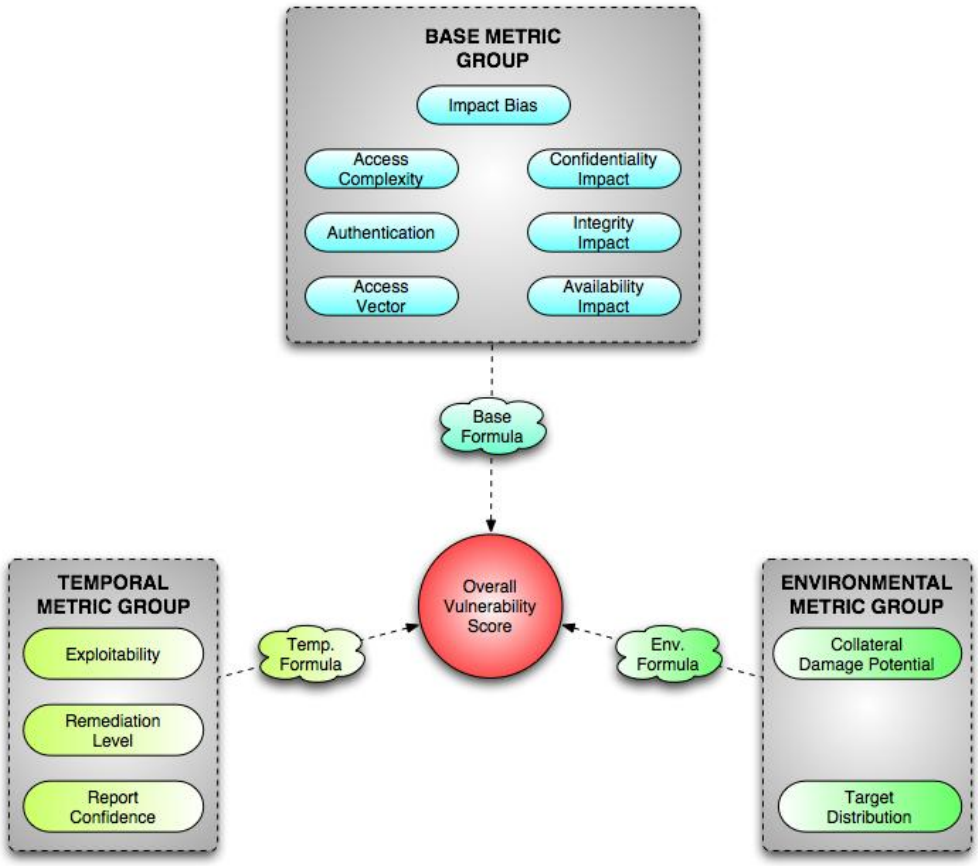

Figure 24. CVSS v1 metric groups

Comparing CVSS v1 and v2, in Figures 24 and 8 respectively, one can notice the strong similarities. However there are some differences that will be discussed. Particularly, in terms of the base metric group, all metrics are the same, expect with the addition of the metric Impact Bias. This allows the analyst to weight more or less a particular security property in terms of confidentiality, integrity and availability.

The Impact Bias metric will have no effect if the three impact metrics are all assigned the same value. Moreover, the temporal metric group remains identical in both versions and the environmental metric group has only the Collateral Damage Potential and Target Distribution metrics and misses the environmental impact metrics for security properties. Note that the metric and rating values have less granularity and the vulnerability severity formulas are different compared to the CVSS v2.

CVSS version 2

The updated CVSS v2 (Mell et al., 2007) relies on the foundations of v1 and is demonstrated on page 64. It is constructed to provide greater score diversity and better accuracy of vulnerability scoring in terms of prioritizing and reflecting the real impact of vulnerability. The difference between versions is that v2 offers a more balanced distribution scoring compared to v1. It consists of the same metric groups (base,

temporal and environmental) and uses the same severity scoring from 0 to 10, however the differences between CVSS v2 compared to v1 are highlighted as follows:

- The Impact Bias metric in the Base metric group does not exist
- Three additional metrics have been added in the environmental metric group and refer to security requirements, namely confidentiality, integrity and availability requirement.
- Metric values have become more granular
- Rating values also follow this notion
- All group metric formulas have been updated

WIVSS (Weighted Impact Vulnerability Scoring System)

WIVSS has roots in the CVSS and is proposed by Spanos et al., (2013) as means to improve even further the vulnerability scoring in terms of accuracy and balanced distribution. Authors claim that, based on statistical analysis in terms of median, mean, standard deviation and different scores, that WIVSS offer higher score diversity and spreading of scores, objectives required from the transition from CVSS v1 to v2. In the following bullets, the main characteristics that differentiate WIVSS from CVSS v2 are presented.

- Weight of Confidentiality Impact > Weight of Integrity Impact > Weight of Availability Impact.
- When "None" Confidentiality Impact then all other Impacts take value "None" = 0.
- Partial Impact = 0.5 * Complete Impact.
- The Impact Score ranges from 0 to 7.
- The 33 possible sums of the three Impact metrics must be different.
- Impact Score = Confidentiality Impact + Integrity Impact + Availability Impact.

PVL (Potential Value Loss)

Wang & Yang (2012) developed the PVL method for rating single vulnerabilities of computer networks. It uses seven indicators to describe vulnerability severity in terms

of effectiveness, usability, accuracy and orderliness. The PVL metric offers higher vulnerability score diversity and vulnerability distribution evenness compared to CVSS v2. Authors demonstrate the practicality of PVL by using PVL to rate published vulnerabilities for IP Multimedia Subsystem

3.4.3 Hybrid vulnerability scoring methods

Hybrid vulnerability scoring methods are a hot topic in research since they combine qualitative description with quantitative measurement. This allows them to be more appropriate during analysis and scoring of vulnerabilities. In the following, hybrid vulnerability scoring methods are presented.

VRSS (Vulnerability Rating and Scoring System)

In order to bridge the gap between qualitative and quantitative vulnerability scoring methods, the VRSS hybrid scoring system is proposed (Liu & Zhang, 2011). This is basically a combination of qualitative methods such as ISS X-Force and Vupen Security and quantitative method such as CVSS. Based on normal distribution fitting analysis, the ISS X-Force and Vupen Security present greater consistency compared to the CVSS. VRSS prioritizes vulnerabilities into three parts: High, Medium and Low. The objective of VRSS is to replace different vendor-specific rating systems and a much more balanced distribution scoring compared to quantitative scoring methods.

VRSS improvement

Liu et al., (2012) developed an updated version of VRSS as means to increase even further the diversity of scoring (in terms of separating vulnerabilities from each other as much as possible) and accuracy of results (in terms of prioritization). VRSS improvement is characterized, first, by enabling vulnerability type to prioritize vulnerabilities, in terms of Common Weakness Enumeration (CWE), and second, by an analytic hierarchy process (AHP) based on the VRSS. Again, authors use statistical analysis, in terms of standard deviation, mean, and distribution analysis, to support their results.

Vulnerability prioritization via fuzzy logic process

Huang et al., (2013) proposed fuzzy logic processes to prioritize vulnerabilities. Authors use a Fuzzy Delphi method which filters the subjectivity from stakeholders regarding the factors affecting security. Then, based on each factors' value, authors use a Fuzzy Analytical Hierarchy Process (AHP) to obtain the fuzzy membership values. The factor membership values are analyzed through a fuzzy synthetic decision

making model in order to show the various degrees the vulnerability is affecting the security of a software system. This method highlights the fuzziness of stakeholder involvement and how different weights of evaluation criteria, which are based on CVSS v2 metrics, affect vulnerability prioritization.

3.5 Chapter summary

In this chapter, we presented specifications from the Security Content Automation Protocol (SCAP) and similar together with vulnerability scoring methods. SCAP is a suite of specifications that allow for automatic security content automation including identification of software flaws, measurement of weaknesses, vulnerabilities and configuration settings, threat classification and compliance checking. SCAP-like specifications perform a similar role to SCAP specifications however they do not belong in the SCAP family as in the time of press of this dissertation. Vulnerability scoring methods aim to provide an assessment of vulnerability and scoring of its severity. Qualitative methods are subjective in nature, lack of a formal measurement of vulnerability characteristics and provide a simple indication of the severity status of vulnerability. Quantitative methods are objective in nature, depend of vulnerability characteristics analysis and provide a measure of assurance which can be used to compare vulnerabilities. Hybrid methods take the best from qualitative and quantitative methods to score vulnerability severity and appear as the most complete solution to vulnerability scoring.

"The biggest risk is not taking any risk"
Mark Zuckerberg

Chapter 4

Software trust

4.1 Introduction

In our information-based society, enterprises are dependent on software platforms (see definition of software platform in the introduction section, on page 11) to perform operational activities. This strong dependency brings opportunities such as greater organizational management and improved operational efficiency but also carries significant risks that arise from the unfortunate reality that software is inherently not as mature as it should be therefore lacking trust during its application. It follows that software security depends on software trust and the level of confidence it reflect from the perspective of end users.

Software trust is closely related to maturity and is affected by numerous challenges such as a) number of vulnerability occurrences, b) threat significance and c) user requirements (Casey, 2010; Pressman, 2010). Therefore, the research question becomes: *how much one can trust software platforms to perform and fulfill business, security and personal objectives*? By software trust, we define the degree of existing confidence that software will be acceptable for individual needs (Amoroso et al., 1991). This implies that software trust is established only after one has established confidence based on a meaningful set of information, that the software does not include flaws that will prevent it from meeting its requirements. Particularly, the challenge with software trust is its subjective nature. This implies that there is a growing need to develop approaches and techniques using existing data.

The interoperability among information infrastructures and sectors implies that everyone is expecting plug-in software with quality characteristics that can be trusted. However, challenges such as the complexity of deploying software right out of the box accompanied with user requirements for quality and customization, put pressure on software vendors to produce quality software platforms from the very beginning. This appears to be utopia since software from its nature is incomplete, prone to errors in the face of vulnerability occurrences and attractive to attackers.

In order to determine the availability of tools that can improve the quality of software, in this chapter, we present how software maturity is attained in terms of software

development models given emphasis on the Software Assurance Maturity Model (SAMM). In section 4.3 we discuss software quality standards and in section 4.4, we present a case study that examines the behaviour of software versions based on the number of vulnerability occurrences in order to indicate when each version is mature enough to be trusted. In section 4.5 we describe related work in terms of software trust and section 4.6 concludes this chapter.

4.2 Software maturity

In pursue of improving software trust, the interest lies on optimizing software maturity. Towards this objective, we present software development models that a) consist of standardized processes, b) provide prescriptive guidance related to improvements of software practices and c) propose a unified approach to software maturity. By software development model, we mean a model that refers to a series of processes that correspond to software engineering tasks required to transform application software into user and business requirements (Magdaleno et al., 2012). A software process can be defined as a dialogue or a "road map" which the process is communicated to the people involved (Pressman, 2010).

In the context of software development, the field of model-driven engineering (MDE) has been proposed as means to deal with complexity, improve software maturity and quality and shorten software development time (Gorschek et al., 2014). Hence, in this field, we present models that fall into the following categories: plan-driven, agile and free/open source software development (F/OSSD) models. The models that adhere to the aforementioned categories have the same objective, to improve on the quality of software applications, however, this is approached in a different way.

Plan-driven models are designed to assist large and complex environments with high costs where trust between the client and the development team has not been established. The aim of software development process in this category is to ensure consistency and predictability (Chrissis et al., 2006) and includes maturity models such as the Software Capability Maturity Model (SW-CMM), the Capability Maturity Model Integration (CMMI), the Software Assurance Maturity Model (SAMM) and software quality standards (Magdaleno et al., 2012). SAMM is presented in the following lines and software quality standards are discussed in 4.3.

SAMM is an open framework that allows for strategy formulation towards software security and development. It is vendor neutral and tailored to an organization's software security practices. The aim of this maturity model is to provide accurate metrics to measure improvements achieved, reduce complexity and support integration with other models and standards. It consists of twelve security practices that adhere to four business functions such as Governance, Verification, Deployment and Construction. Each business function is briefly described as follows. Figure 25 highlights the SAMM framework.

- **Governance**: Describes the way software development is managed within the organization

- **Construction**: Describes the way an organization defines goals and develops software projects

- **Verification**: Describes the way software development is tested

- **Deployment**: Describes the way software applications are utilized and supported in the organization

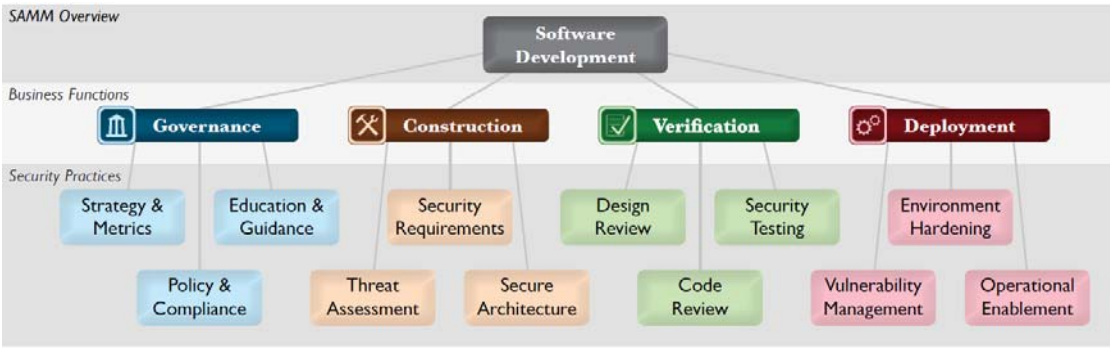

Figure 25. OpenSAMM framework (Source: OWASP, 2009)

For every security practice there are sequential goals accompanied with maturity levels that range from 0 to 3+ where 0 denotes negligible maturity and 3+ denotes maximum maturity. Each maturity level performs as an indicator for the proposed business function and reflects the current state of maturity. The utilization of SAMM is recommended with the accompanied self assessment tool, a subjective, qualitative questionnaire which allows users to score each security practice based on a yes/no option (Asterix, 2014). The final scorecard is a comparison gap between the current and the optimized maturity levels for each security practice.

Reviewing business functions, one can notice the following characteristics: a) heterogeneity, because each one derives from different security practices and b) uncertainty, because each one depends on the maturity level of the security practices that support the same business function and c) dependability because each business function depends on the maturity level of the security practices that support another business function. In other words, a low maturity level of the Construction business function does limit the potential of achieving a high maturity level in the Verification business function.

This framework's highlight, apart from traditional security practices such as Construction, Verification and Deployment, is Governance. This is because most compliance frameworks require, or at least allude, to software assurance practices. For example, PCI DSS Requirement 6 and ISO 27002 Section 12 both require assured software development practices. However, implementing a secure Software

Development Lifecycle (SDLC) is difficult without knowing where it begins and how software maturity is progressing or not. In this respect, SAMM supports a framework as a step-by-step approach to software assurance that is utilized via the homonymous questionnaire.

This tool is a standalone checklist that mimics the implementation of SAMM and can be applied to evaluate existing maturity levels of software products in one of the following options: a) default, b) independent software vendor, c) online service provider, d) financial services organization and e) government organization. Each option has different maximum maturity levels. Overall, SAMM is a comprehensive software development model that aims to improve the maturity of software. It provides actionable recommendations for improving application security and can be utilized through the self-assessment SAMM questionnaire.

The Software Capability Maturity Model (SW-CMM) is a model which provides guidelines to software organizations for constant optimization at five levels: Initial, Repeated, Defined, Managed and Optimized. Each level includes various key software process areas (Paulk et al., 1993). Although SW-CMM aims to achieve higher levels of software process maturity, the models' limitations are a) the lack to address expertise in particular application domains and b) does not include compliance or governance issues. In an attempt to improve the usability of SW-CMM even further, the Capability Maturity Model Integration (CMMI) is proposed as a successor to software development process.

The model uses five scales; Initial, Repeatable, Defined, Quantitatively Managed, Optimizing, to rate the maturity levels of software processes. The latest version 1.3 (Software Engineering Institute, 2010) incorporates principles from the concept of agile software development, allowing for integration of separate organizational functions including government initiatives. Limitations of using CMMI are the limited applicability since it may not suit for every organization and the documentation may be overwhelming and requires professional staff to initiate CMMI-based process improvement.

Agile modelling is characterized by increased flexibility to adapt to technological, business and user requirements during the software project lifecycle (Boehm et al., 2002). In this category the models that apply are the following: Extreme Programming (XP), Scrum, Crystal, Kanban, Adaptive Software Development (ASD), Dynamic Systems Development Method (DSDM), Feature Driven Development (FDD), Lean Software Development (LSD), Agile Modeling (AM) and Agile Unified Process (AUP). The common characteristic behind the utilization of agile models is the ability to deal with the cost of change, which increases nonlinearly as project progresses, as means to adapt to changes before they actually happen (Pressman, 2010; Dingsøyr et al., 2012).

F/OSSD-oriented modelling distinguishes itself by providing a distributed perspective towards software development (Scacchi et al., 2006). This implies that this type of modelling relies on the freedom it provides to users to access, apply, improve and distribute the program code in a collaborative manner. In this respect, this type of modelling relies on reinvention as means to enable continuous improvement. Specifically, it allows users and developers to use the program code on a voluntary

basis for personal or business purposes and provide feedback through the Internet. This diversity of use allows F/OSSD to improve the software product through a multiple review process during its lifecycle (Koch, 2005).

In summary, all categories towards software development have distinctive characteristics. Plan-driven modelling adds value to software development through extensive documentation as means to illustrate as many software processes as possible. The focus is to analyze software content into processes, however, this may cause a certain level of complexity to deploy. Agile modelling aims to fulfil a series of requirements through technological, business and user perspectives. The focus is on the agility to adapt to change however, this agility may not cover the software content range as detailed as the plan driven modelling. F/OSSD modelling aims to improve on the software development through the utilization of the distributed nature of the source code. This implies that it is more close to plan driven modelling, however, it relies on the feedback of users and developers.

4.3 Software quality standards

Software quality standards have the purpose to improve the development process, increase maturity and establish trust in software (Pressman, 2010). Particularly, software standards, such as the Common Criteria for Information Technology Security Evaluation (CCMB-2009-07-001; CCMB-2009-07-002; CCMB-2009-07-003), ISO 9126 (ISO/IEC 9126:1991); ISO 25010 (ISO/IEC 25010:2011) and others more technical such as IEC 61131-3 (IEC 61131-3:2013) and IEC 61508 (IEC 61508:2008), offer ad hoc solutions towards software quality which implies that such standards are subject to multiple rounds of ad hoc review and revision (Graydon and Kelly, 2013).

Other software quality standards describe either implicitly or explicitly a software development process. On one hand, software development is implicitly stated in standards such as ISO 25010 (ISO/IEC 25010:2011), which describes a quality model through systems and software quality requirements and evaluation, and ISO 25012 (ISO/IEC 25012:2008), which defines a general quality model for data retained in a structured format within a computer system. On the other hand, standards such as ISO 12207 (ISO/IEC 12207:2008) refers to software development explicitly by proposing a common framework for software life cycle processes including acquisition, supply, development, maintenance and disposal of software products. Moreover, ISO 15504 (ISO/IEC 15504-5:2012), which uses definitions from ISO/IEC 12207:2008 and ISO/IEC 15504-2, identifies and supports a Process Reference Model. This model is used for software development purposes including process and capability dimensions accompanied with assessment indicators for software processes.

According to ISO 8402 (AS/NZS ISO 8402:1994) on quality management and assurance, it is explicitly stated that in order to evaluate software quality, one must first define and identify the “needs” of the software. Those “needs” include functional and non-functional characteristics of the software and constitute for the complexity of

operating a software product. This complexity derives not only from analyzing software quality characteristics such “usability”, “maintainability”, “functionality” and others as described in ISO/IEC 9126-1 (ISO/IEC 9126-1:2001), a standard on software engineering and quality, but mainly from setting clear goals during the software development process.

In the context of software quality, the term software quality assurance (SQA) is defined as the process of improving the quality of software during its life cycle (Tompkins and Rice, 1986). A complementary definition takes into consideration different perspectives in the face of four key dimensions: technical, managerial, organizational, and economic (Rai et al., 1998). According to Sarigiannidis and Chatzoglou (2014), the relationship between SQA and software development models is strong since modern software require mature processes at an early stage of development in order to mitigate emerging risks as soon as possible and appear as a quality software project. This implies that the identification and evaluation of software processes at early stages of the software lifecycle increases the quality of the product, minimizes risk that may derive from immature software processes and improves the process quality.

A recently added term that refers to software quality is Social Software (SoSo) (Giuffrida and Dittrich, 2013). Authors claim that SoSo has the potential to improve on the software development process through communication which implies improved quality with reduced costs and time to mature. In addition, a number of studies (Losavio et al., 2004; Bonfè et al., 2013; Vyatkin and Zoitl, 2013; Martín and Yelmo, 2014) highlight that to increase software quality there must be automation in software process assessment.

Another research (Yuen and Lau, 2011), proposed a fuzzy group analytical hierarchy process for assessing the quality of software based on subjective judgments from a group of experts at different organizational levels. Authors’ approach is linked to the international standard of software quality attributes, ISO/IEC 9126-1:2001 as described in Jung et al., (2004). This approach may offer a solution to measure the level of software quality however it would be interesting to see how the proposed fuzzy group analytical hierarchy process helps decision making in the software development process.

4.4 Case study

In the following case study, we examine software products (i.e. versions of a particular production chain) through vulnerability analysis in order to indicate when a version is mature enough in order to be trusted. It is a fact that software evolves over time because it is affected by a) new vulnerabilities, b) updates and patches and c) user requirements. Evolvability is a crucial characteristic of information infrastructure platforms and refers to the stability, maturity and performance of software during time (Diane, 2006; Mannaert et al., 2011; Mannaert et al., 2012).

In this case, we examine different versions of the Apache HTTP web server based on the number of vulnerability occurrences. The aim is to discover how vulnerability occurrences affect software maturity and thereby trust. Particularly, we introduce a three point-to-point analysis that includes two maturity intervals. Each point is an indicator that expresses the maturity status of a version at a particular time. Specifically, the three points are defined as follows:

- **Zero-point**. This point is defined the time each version is released to the public.
- **Point of inflection.** This point is defined the time each version reaches its peak in the number of vulnerability occurrences.
- **Point of steady state.** This point is defined the time where the number of vulnerability occurrences does not vary greatly over time and reveals that the version is mature enough and can be trusted

To calculate the points, we used a second degree polynomial function given in the following formula

$$\alpha x + \beta = 0 \quad (5)$$

In between the points there are intervals. Those intervals reflect the maturity status for each version. Specifically, the maturity interval between zero point and point of inflection characterizes each version as immature since the number of vulnerability occurrences is steadily increasing. In the contrary, the maturity interval between the point of inflection and point of steady state reflects that each version is introduced to a maturity phase.

After the point of steady state, the number of vulnerability occurrences does not change significantly over time and therefore, the version can be considered mature enough to be trusted. In the following Table 13, the number of vulnerability occurrences, according to NVD, for each Apache version is depicted on a per semester basis.

Semesters	1.3v	2.0v	2.2v	2.3/2.4v
0	0	0	0	0
1	0	1	2	1
2	2	2	4	3
3	5	3	7	5
4	5	5	3	2
5	5	4	2	3
6	3	3	4	4
7	3	2	3	3
8	2	1	1	1
9	1	2	1	2

Table 13. Number of vulnerability occurrences for Apache versions

Note that the number of vulnerability occurrences is calculated based on the relative time the version is released (e.g. the 2.0v has first been released in 10 March 2000). The following Figures (26) – (29) show how maturity is progressing for each version and Figure 30 shows the average maturity behavior of the aggregated versions. The blue line represents how the number of vulnerability occurrences is progressing and the black line is the polynomial curve. For each Figure, the horizontal axis reflects time in terms of the number of semesters, and the vertical axis reflects the number of vulnerability occurrences.

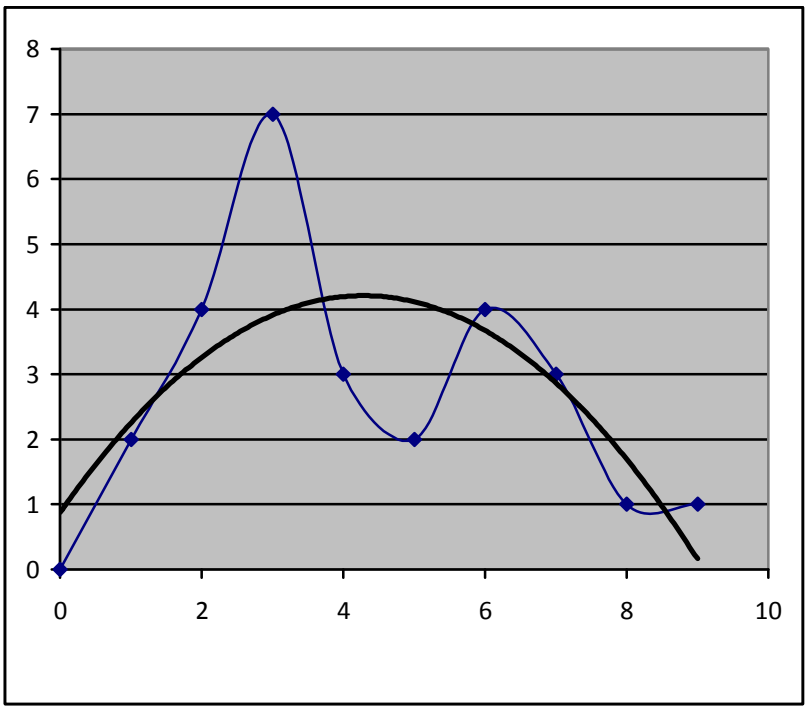

Figure 26. 2.2v

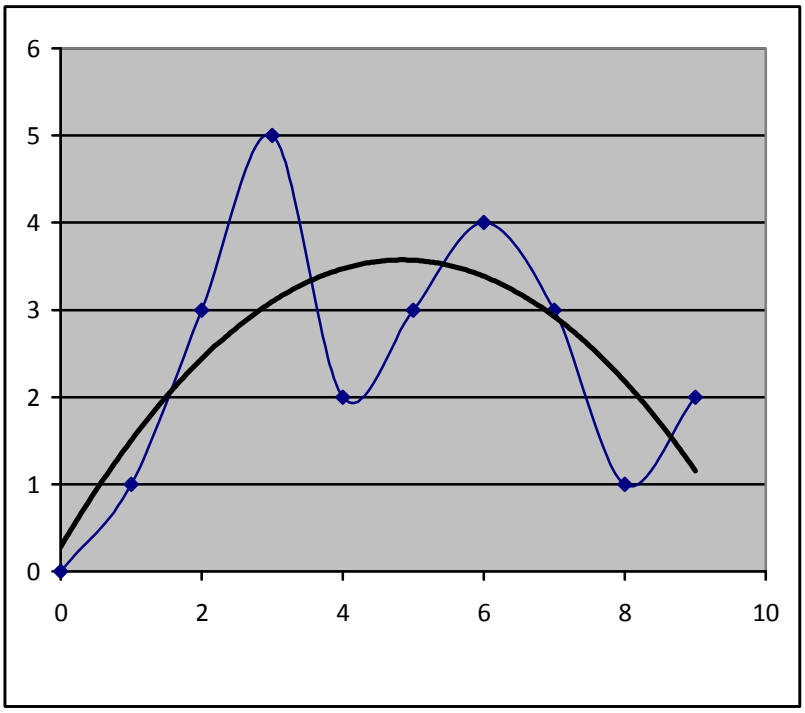

Figure 27. 2.3/2.4v

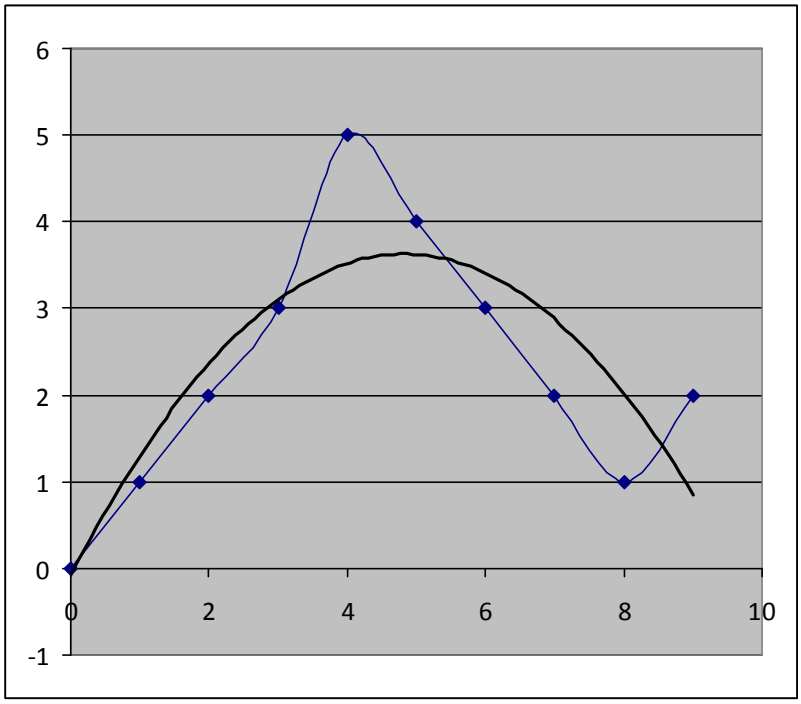

Figure 28. 2.0v

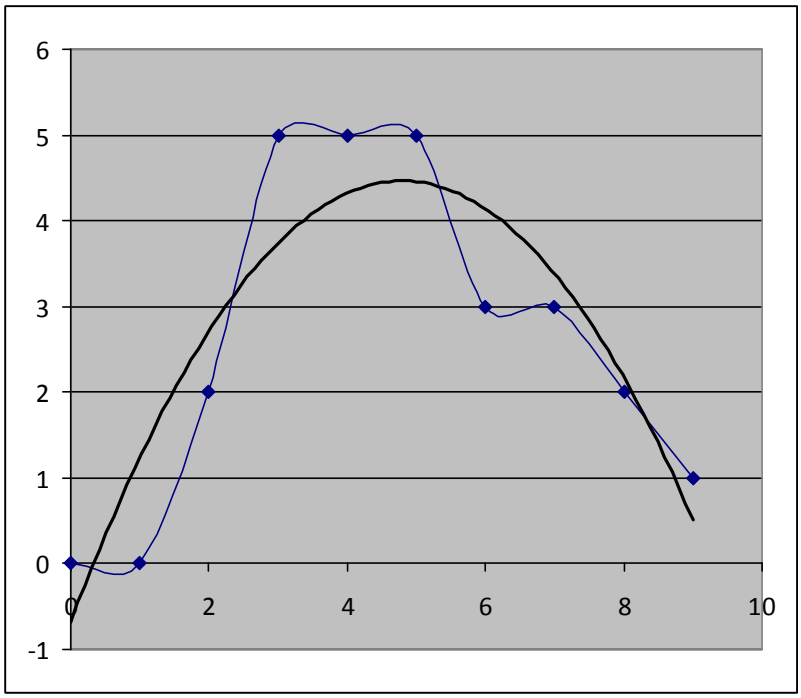

Figure 29. 1.3v

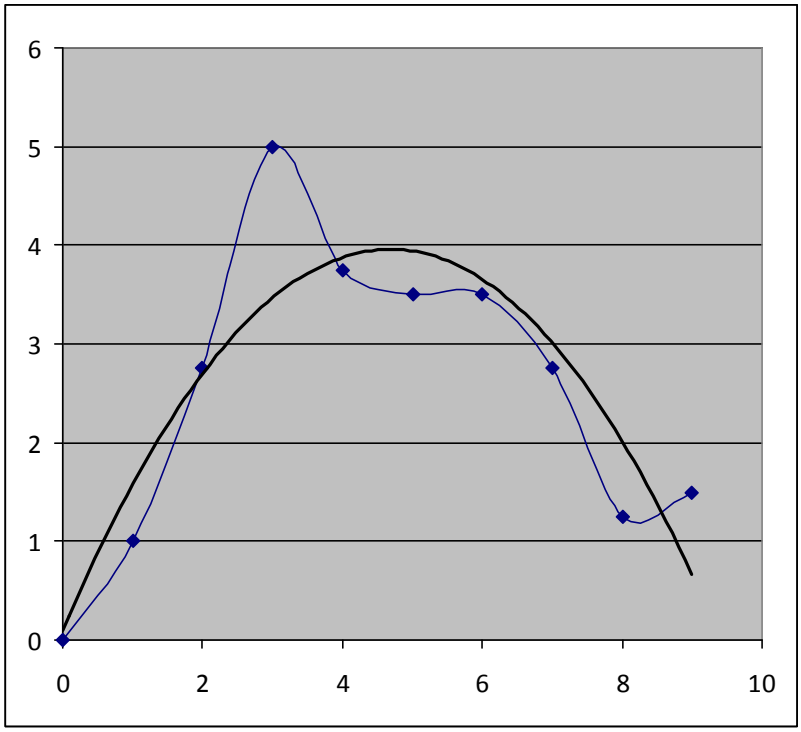

Figure 30. Average

According to Figures (26) – (30), versions 2.2v and 2.3/2.4v present the highest volatility in terms of vulnerability occurrences. Moreover, in a greater extent in Figures (28) – (30) and to a lesser extent in Figures (26) and (27), the blue line follows an inverse U curve shape. This observation was first proposed by Kuznets

(1955) in the field of economics to explain that countries developing from farm-based economies to industrial economies, their income inequality initially increases then, reach a peak, and then decreases. Another important observation is that the point of inflection occurs three times in a row in 1.3v. As a final observation, the polynomial curve presents the higher intensity in 1.3v and the lowest in 2.3/2.4v. In the following Table 28, software maturity points and intervals are calculated based on formula 5.

Version	Polynomial function	Point of inflection	Point of steady state	Maturity intervals
1.3v	f(x)= -0,446x+1,858	4,165919283	8,331838565	[0,4.16,8.331]
2.0v	f(x)= -0,318x+1,504	4,729559748	9,459119497	[0,4.72,9.45]
2.2v	f(x)= -0.181x+1,923	5,312154696	10,62430939	[0,5.31,10.64]
2.3/2.4v	f(x)= -0,28x+1,476	5,271428571	10,54285714	[0,5.27,10.54]
Average	f(x)= -0,352x+1,69	4,80114	9,6022727	[0,4.8,9.6]

Table 14. Maturing points and interval calculation

According to Table 28 results, the 1.3v is the version which both points of inflection and steady state occur first compared to the other versions which imply that the 1.3v reaches first the maturity intervals which further implies that the 1.3v can be trusted earlier compared to the other versions. On the contrary, the 2.2v is the version that trust is developed the latest.

4.5 Related work

This section presents related work for software trust. Li et al., (2012) highlighted the growing demand for trusted software and proposed a measurement model that includes process risk management and software trustworthiness metrics. Authors, based on simulation cases analyzed the effectiveness of the proposed model and concluded that risk management is critical to enhance the trustworthiness of software. The same authors outline the lag in literature review to include measurement of software trust due to the ambiguity related to this domain.

Hertzum (2002), outline that trust is a central property to software quality, thereby establishing a perceived quality of a software product is essentially a matter of establishing to what extent one is willing to place trust in it. The author describes that trust in software is affected by a) first-hand experience, b) reputation, c) inspection of attributes and d) general assumptions and stereotypes. This research concludes that software trust differs based on the amount of evidence to characterize software products.

Al-Ani et al., (2014) investigated the potential of using tools, such as software features, organizational structures and office technologies, to support trust in software engineering teams. Based on grounded theory, authors interviewed 71 employees from multinational organizations and conclude that the development of trust among software collaborative teams has a positive effect on software trust.

Amoroso et al., (1991) proposed an approach to measure software trust at some state in the software development life cycle. Based on a set of criteria classes consisting of trust principles, authors provided a scale of measuring software trust. While this approach paved the way for more research on software trust, considerations is the lack of conceiving trust not only as part of the software development process but also as part of maturity as software life cycle progresses.

Another research conducted from Schryen et al., (2011) proposed a trust measurement method for distributed environments. Based on propositional logic and probability theory, authors determine trustworthiness to distributed systems based on trust metrics given that distributed systems fulfil a particular security requirement.

Close to this research, Babar et al., (2007), conducted a survey on 12 Vietnamese software development practitioners and identified that cultural understanding, creditability, capabilities, and personal visits are important factors in gaining trust in outsourcing software relations. Particularly, client trust, cultural understanding, communication strategies, contract conformance, and timely delivery are vital factors in maintaining software trust in outsourcing relationships.

A complementary research from Siakas and Siakas (2008) explored the challenges of trust in software outsourcing. Authors conclude that transcending time, space and culture affect software trust in outsourcing projects and future research has been suggested in the field of software trust in terms of exploring software behaviour.

Moreover, Yan and MacLaverty (2006) presented an autonomic trust management solution that can specify, evaluate and set up trust in a component based software system. Based on a number of algorithms that use subjective logic, authors conducted trust assessment and maintenance during component execution. Authors define software trust from a system point of view as the assessment of the observed behaviour of software characteristics towards an intended purpose. This implies that software trust is a subjective, dynamic, and sensitive to change process which is influenced by multiple factors. The same research recommends that approaches are required not only for trust measurement but also for trust establishment and sustaining.

Bugiel et al., (2011) discuss the availability of software solutions and the lack of assessing software security as means to build software trust. Authors present a scalable approach for the trustworthiness assessment of software programs based on security history. The approach is used to automatically sort programs by their security record or to assess the trustworthiness of complex software systems in remote attestation schemes. Implementation results in a Linux system recorded promising accuracy results.

Recently, Lai et al., (2014) conducted a research on software trust analysis based on state graphs. Authors proposed a trusted modeling approach based on Hidden Markov Model and generated state transition graphs according to software functions by semantic analysis. Based on a case study for RSS software, results demonstrate that software behaviour can be analyzed for anomalies, illegal input behaviour and detection of untrusted behaviour.

Last but not least, research from Yuana and Hanab, (2011), presented a software behaviour trustworthiness measurement method including partitions based on data mining. The method compares datasets generated at software running time with static attribute features at a three stage scheme. First, defines the concept of software trust, second provides a group of formulas for measurement and comparison of the types of feature datasets and third, develops an architecture that uses an algorithm to evaluate conceptualized hierarchical software trustworthiness.

4.6 Chapter summary

In this chapter, we elaborated on the context of software trust and presented software development models and quality standards that help develop reliable software. In addition, we examined how particular software versions progress over time through vulnerability analysis. Particularly, we defined maturity points and intervals that express the maturity status for each version. This chapter concluded with an extensive literature review on software trust.

"Information is not knowledge"
Albert Einstein

Chapter 5

5. Risk prediction methodology

5.1 Introduction

In our modern society, sectors like banking, finance, government services, and communication technologies, rely on information infrastructures to perform operational activities. Each information infrastructure relies on software and hardware platforms that constitute assets vital for the maintenance of business goals and information assurance. However, information infrastructures are constantly under threat due to a number of challenges, such as the accelerating change of technology, open networks, third party dependencies, outsourcing risk, stakeholder involvement and government requirements for stricter regulation through compliance and policies (Koons and Minoli, 2010; Masera and Fovino, 2007; Theoharidou et al., 2010; Veríssimo et al., 2006). In this regard, the concept of information infrastructure protection (IIP) is of critical importance to ensure that all sectors continue to function and interoperate in an optimum way (Rinaldi et al., 2001).

The growing concern towards zero-day vulnerabilities and threats that derive from software-based platforms and challenge the sustainability of an information infrastructure is the source of the main issue we try to address. By zero-day vulnerabilities and threats we mean: a) vulnerabilities that have zero-day awareness and are unknown and b) threats that exploit zero-day vulnerabilities (Bilge and Dumitras, 2012; Zhang et al., 2011). Another matter of concern is the risk from exploiting zero-day vulnerabilities. Due to a lack of a formal definition on zero-day risk in the literature, we define zero-day risk as the uncertain loss caused by a damage event and the corresponding impact for each security property, namely confidentiality, integrity and availability.

Challenged by the fact that one can predict only what can be measured, we approach the above issues from a security metrics' perspective, as means to measure and prioritize vulnerability severity. Despite marked progress in security metrics, in terms of vulnerability scoring systems, the scoring of vulnerabilities is still a hot topic in research (Liu et al., 2012). This fact, combined with threats increasing in significance (Aburrous et al., 2010), creates the need to develop prediction techniques as defences to zero-day risk. We believe that early recognition of zero-day risk is in favour of the

IIP, because precaution is better than cure, allowing for resource optimization and sustainable allocation of security controls.

To the best of our knowledge, there is a lack in literature about measuring zero-day risk since most of the published research focus on prediction of zero-day vulnerabilities alone. In this respect, the contribution is a novel risk prediction methodology as a proactive approach to IIP. The research goal is to aid the auditor during decision making in response to potential risks on a real-time basis and in advance of public disclosure, applying appropriate protective actions. The proposed methodology aim at offering promising results based on stochastic approaches and standardized metrics. To this extent, we use specifications from the Security Content Automation Protocol (SCAP) as means to enable automated software platform and vulnerability identification, quantitative measurement and comparison of results (Waltermire et al., 2011).

This chapter is organized as follows: section 5.2 is a step-by-step analysis of the proposed methodology and in section 5.3 we present the practicality of the methodology with an implementation example from the e-banking sector. In section 5.4, we present related work in terms of vulnerability prediction and in section 5.5 we develop and demonstrate a two-phase evaluation method for vulnerability prediction. In section 5.6, we discuss the proposed methodology and outline limitations. In section 5.7 we conclude by summarizing this chapter.

5.2 The proposed methodology

In this section, we describe step-by-step the proposed risk prediction methodology through platform vulnerability analysis based on SCAP specifications (Waltermire et al., 2011). The first three steps include the platform vulnerability analysis and the fourth step is the information infrastructure risk prediction process. The steps are the following:

Step 1: Platform identification

Step 2: Vulnerability History

Step 3: Vulnerability prediction

Step 4: Risk prediction

5.2.1. Platform identification

The objective of the first step is to identify and classify information infrastructure supporting platforms. To achieve this, we use the Common Platform Enumeration (CPE) as presented on chapter 3 on pages 63 and 64.

5.2.2. Vulnerability history

The objective of the second step is to measure the historical rate of vulnerability occurrences, i.e. the rate by which the vulnerabilities of identified platforms occurred for a defined time period in the past, in our case that of a semester. To achieve this, we retrieve historical data from the NVD, in terms of the Common Vulnerabilities and Exposures (CVE) as presented on chapter 3 on pages 61 and 62. In this case, we measure the entry type vulnerabilities.

5.2.3. Vulnerability prediction

The objective of the third step is to predict the probability that the number of vulnerability occurrences will be less or more for the following semester. To achieve this, we use a distribution fitting procedure as means to estimate the statistical correlation between empirical and reference probability distributions in terms of the Cumulative Distribution Function (CDF).

Next, we verify the distribution fitting results using the Kolmogorov-Smirnov (K-S) test (Papoulis and Pillai, 2002). The K-S test can be described as a goodness of fit test that verifies whether an empirical probability distribution follows a reference probability distribution (Marsaglia et al., 2003). In this respect, we use the K-S test to assess the statistical significance of the distribution fitting results, revealing in this way future vulnerability occurrences patterns. We define the variable *v*, as the number of vulnerabilities during a semester, such as $v \in [0+\infty), v \in \mathbb{N}$. By definition, the CDF function is (Papoulis and Pillai, 2002):

$$F(x) = P(v \leq x) \tag{6}$$

where $F(x)$ is the CDF function that expresses the probability that v takes a value less than or equal to x, whereas x expresses the number of vulnerabilities of the semester preceding the semester the vulnerability prediction is conducted. As $v \geq 0$, $F(x)$ denotes the probability P_1 that the number of vulnerabilities will be less or equal to the number of vulnerabilities of the preceding semester and P_2 denotes the

probability that the number of vulnerabilities will be more and is expressed as follows:

$$P_2 = 1 - F(x) \tag{7}$$

5.2.4 Risk prediction

The objective of this step is twofold. First, we model the possible conditions to achieve information infrastructure risk prediction and second, we estimate zero-day risk in terms of security properties. To achieve the first objective, we build a risk model based on a Bayesian Belief Network (BBN) topology (Figure 31) as means to a) model the dependencies among the risk elements namely, vulnerability (*V*), threat (*T*), and damage (*D*) on information security properties, namely confidentiality (D_c), integrity (D_i) and availability (D_a) and b) allow traceability of results. The BBN is a directed acyclic graph (DAG) together with an associated set of probability tables, known as conditionally probability tables (CPTs). A DAG consists of nodes that represent variables that use arcs to define the relationships (dependencies) among the variables (Brændeland et al., 2010).

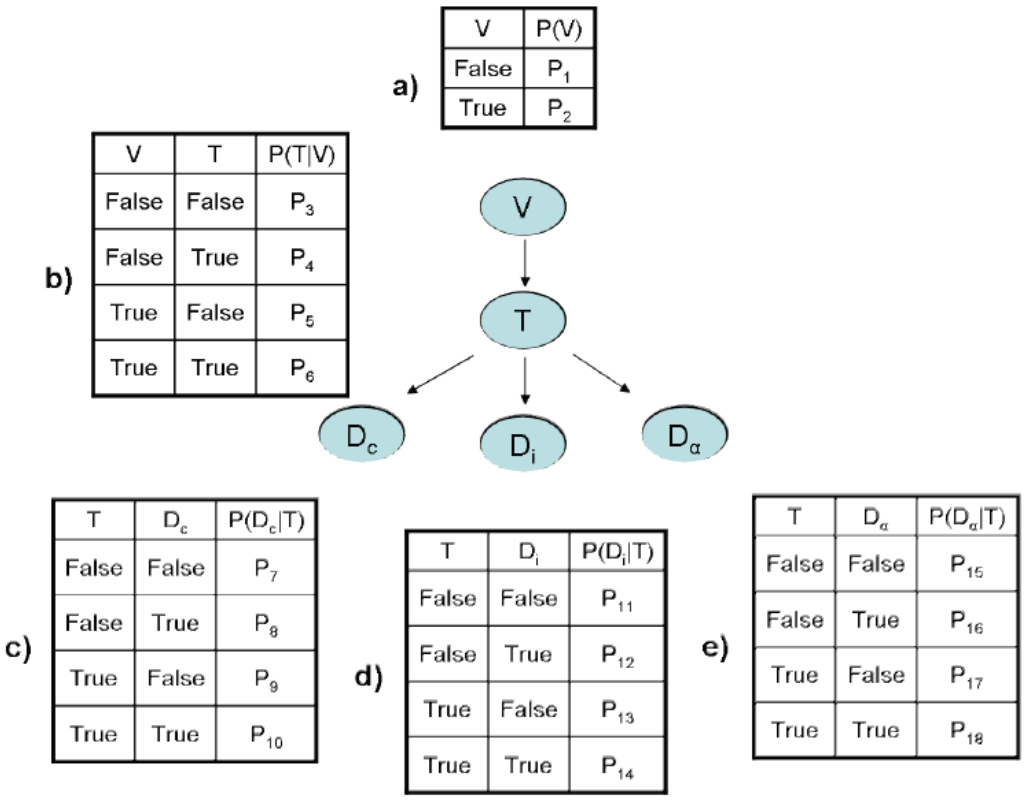

Figure 31. BBN topology and CPTs (a to e)

The proposed BBN topology, presented in Figure 30, illustrates the structure of conditional probabilities for each child node given the evidence of its parent node (Druzdzel, 1996, Xie et al., 2010). For each node, the accompanying CPT is shown and labelled with a letter (a to e). The starting point is CPT (a) which represents V with Boolean values where False expresses a decreased number of occurrences (denoted as V^-) and True an increased number of occurrences (V^+). In CPTs (b-e), False expresses a decreased significance of threats (T^-) or damages in confidentiality (D_c^-), integrity (D_i^-), availability (D_a^-) and True an increased significance of threats (T^+) or damages in confidentiality (D_c^+), integrity (D_i^+), availability (D_a^+).

The probabilities P_1 and P_2 in CPT (a) are complementary and this applies to the remaining conditional probabilities such as $P_3 = (T^-|V^-)$ and $P_4 = (T^+|V^-)$, $P_5 = (T^-|V^+)$ and $P_6 = (T^+|V^+)$, $P_7 = (D_c^-|T^-)$ and $P_8 = (D_c^+|T^-)$, $P_9 = (D_c^-|T^+)$ and $P_{10} = (D_c^+|T^+)$, $P_{11} = (D_i^-|T^-)$ and $P_{12} = (D_i^+|T^-)$, $P_{13} = (D_i^-|T^+)$ and $P_{14} = (D_i^+|T^+)$, $P_{15} = (D_a^-|T^-)$ and $P_{16} = (D_a^+|T^-)$, $P_{17} = (D_a^-|T^+)$ and $P_{18} = (D_a^+|T^+)$. To calculate CPT (a) and initiate risk prediction, we need to estimate the probabilities for increased and decreased number of vulnerability occurrences in the following semester. For this reason, the estimation of the CDF is necessary as illustrated in subsection 5.2.3.

In order to calculate the remaining probabilities from CPTs (b-e) in Figure 31, we use Von Mises' theory of probability and the Common Vulnerability Scoring System (CVSS) v2 base metric group (Mell et al., 2007) as it is concluded in Chapter 2, Table 10.

The base score shows the severity of a particular vulnerability as it is presented in formulas (1) – (4) in Chapter 2, page 66. However, with the computation of base score for the aggregated number of vulnerabilities of a platform, utilizing the history from the NVD allows us to provide an indication of how vulnerable a platform is, such as the Apache web server. In order to compute an aggregated base score (ABS), we propose the computation of formulas (8) - (13). Specifically, each formula is computed as the sum of the frequencies of metric values multiplied with the respective rating value and divided by the total number of recorded vulnerabilities (i.e. NV).

$$IC = ((fpa_c * 0.275) + (fco_c * 0.660))/NV \quad (8)$$

$$II = ((fpa_i * 0.275) + (fco_i * 0.660))/NV \quad (9)$$

$$IA = ((fpa_a * 0.275) + (fco_a * 0.660))/NV \quad (10)$$

$$EV = ((flo * 0.395) + (fad * 0.646) + fne)/NV \quad (11)$$

$$EC = ((flw * 0.71) + (fme * 0.61) + (fhi * 0.35)/NV \quad (12)$$

$$EU = ((fmu * 0.45) + (fme * 0.56) + (fhi * 0.704)/NV \quad (13)$$

Von Mises' theory of probability relies on a sequence of uniform events (in our case, semesters with recorded vulnerabilities) whose frequency of occurrence converges

toward a limit. These events differ in certain characteristics (in our case, Exploitability and Impact). This frequentist approach relates to infinite sequences of observations, known as collectives. Each collective has to fulfil two conditions. First, the convergence condition (i.e. the limits of the relative frequencies of events with particular attributes within the collective exists). Second, the randomness condition (i.e. the limits remain unchanged with respect to the choice of any subsequence of the collective) (Gnedenko, 2005). Von Mises' theory is expressed as follows (Keuzenkamp, 2000):

$$P_i = \lim_{\lambda \to \infty} \frac{\lambda_i}{\lambda} \tag{14}$$

where P_i is the limiting frequency of an event with characteristic i, λ_i is the number of occurrences of the event with characteristic i, λ is the number of all events. Note that the accuracy of P_i is continuously refined as λ increases.

For CPT (b), we need to compute whether threat significance increases (i.e. T^+) or decreases (i.e. T^-) compared to the preceding semester. To achieve this, we consider that the significance of a threat depends on qualitative characteristics of vulnerabilities and therefore we compute for each semester compared to the preceding semester the threat significance based on formula (2). Hence, P_4 ($P_3 = 1 - P_4$) denotes the conditional probability of threat significance increasing given the number of vulnerability occurrences is decreasing and this is expressed as follows:

$$P_4 = P(T^+ \mid V^-) = P(T^+ \bigcap V^-) / P(V^-) \tag{15}$$

The probability $P(T^+ \bigcap V^-)$ is computed using formula (14) and considering that the number of all events (denoted as λ) is the semesters with recorded vulnerabilities plus the semester the vulnerability prediction is conducted. The number of occurrences of an event with characteristic i (λ_i) is the number of semesters where the threat significance increased and at the same time the number of vulnerability occurrences decreased.

The probability $P(V^-)$ is computed using formula (14) and considering that λ is the same semesters as in the numerator and λ_i is the number of the semesters where the number of vulnerability occurrences decreased. The same rationale applies for the computation of the remaining conditional probabilities in CPTs (b-e). Note that for CPTs (c-e) we compute whether the damage significance increases (i.e. D_c^+, D_i^+, D_a^+) or decreases (i.e. D_c^-, D_i^-, D_a^-) by utilizing formulas (8), (9) and (10) for CPTs (c), (d) and (e) respectively.

According to CLUSIF (2009), there are two ways to express risks, either with a static or dynamic way. On one hand, the static way considers risk as the probability of a threat exploiting vulnerability and the consequences of its occurrence. This static

approach does not incorporate time as a variable and it is impossible to describe risk as an aftermath of sequences of events, causes and consequences. On the other hand, the dynamic way incorporates time as a variable and it is based on a situation or scenario analysis. Towards this perspective, ISO Guide 73 (ISO, 2009) describes risk as the combination of dangerous phenomena (circumstances), triggering events and consequences.

Moreover, in the spirit of Strecker et al., (2010) risk is considered an uncertainty, in that the probability of occurrence as well as the impact is uncertain and can at best be assigned estimated values. In this direction, risk is regarded as an uncertainty associated with an event that can be quantified on the basis of empirical observations or causal knowledge (Gigerenzer, 2002). The author argues that frequencies and probabilities are ways to express risks.

This notion is also supported by Houmb et al., (2010), in which CVSS estimates of frequency and impact are used to quantify security risk. In our case, having defined the dependence structure of BBN and associated CPTs, we are interested to estimate zero-day risk based on the event that induces increased damage significance because this particular event increases the risk for the following semester compared to the preceding semester. Hence, the computation of zero-day risk in security properties is proposed as follows:

$$Risk_c = P(D_c^+) * \text{Impact}_c \tag{16}$$

$$Risk_i = P(D_i^+) * \text{Impact}_i \tag{17}$$

$$Risk_a = P(D_a^+) * \text{Impact}_a \tag{18}$$

where $\text{Impact}_{c,i,a}$ are the consequences for each security property. According to Fenton and Neil (2011), the accompanied impacts are unique for each enterprise and different projects or business divisions will estimate impact differently depending on a localized perspective.

5.3 Implementation example

In the following, we demonstrate the applicability of the proposed risk prediction methodology. In particular, we present a step-by-step implementation example from the e-banking sector due to the high percentage of revenue and the consequent demands for information assurance. Recent studies on the progress and adoption of e-banking recommend a proactive approach to risk management for e-banking (Aduda and Kingoo, 2012; Aggelis, 2005; Angelakopoulos and Mihiotis, 2011; Gikandi, 2010; Shah and Clarke, 2009; Shah and Siddiqui, 2006; Khatri and Budhiraja, 2013; Kondabagil, 2007; Osunmuyiwa, 2013). This notion is heightened by the increasing

rate of phishing techniques and intelligent attack strategies related to this domain (Aburrous et al., 2010).

5.3.1. Platform identification

We used network security tools to identify information infrastructure supporting software platforms, such as Nmap (Wolfgang, 2002) to discover operating systems and Nikto2 (Nikto2, 2010) to discover the running application in terms of web servers. In Table 15, we present an indicative list of the software platforms identified under the CPE specification together with abbreviations which will be used for the next step. Each abbreviation corresponds to a software platform as defined on page 98. For profound reasons, we refer to banks as numbers.

Bank	Operating System		Web Server	
	CPE	Abbr	CPE	Abbr
Bank#1	cpe:/o:microsoft:windows_2008_server	WS	cpe:/a:microsoft:iis:6.0	IIS
Bank#2	cpe:/o:microsoft:windows_2003_server	WS	cpe:/a:microsoft:iis:6.0	IIS
Bank#3	cpe:/o:microsoft:windows_2000	WS	cpe:/a:microsoft:iis:5.0	IIS
Bank#4	cpe:/o:sun:solaris:8	SOL	cpe:/a:apache	APA
Bank#5	cpe:/o:microsoft:windows_2000	WS	cpe:/a:microsoft:iis:6.0	IIS
Bank#6	cpe:/o:linux	LIN	cpe:/a:apache	IBM
Bank#7	cpe:/o:microsoft:windows_2008_server	WS	cpe:/a:microsoft:iis:7.5	IIS
Bank#8	cpe:/o:microsoft:windows_2003_server	WS	cpe:/a:microsoft:iis:7.5	IIS

Table 15. E-banking sector

5.3.2 Vulnerability history

Based on platform classification, we measured the historical rate of vulnerability occurrences. Table 16 shows from the first semester of 2000 until the first semester of 2013, the rate of vulnerability occurrences for the identified software platforms per semester. Numbers may be subject to minor changes due to updates in the NVD.

Year	Semester	APA	IIS	LIN	SOL	WS
2000	1	3	13	58	1	13
	2	13	15	53	0	12
2001	3	5	12	106	8	13
	4	12	11	87	18	28
2002	5	5	3	49	6	9
	6	42	12	125	14	34
2003	7	19	10	60	5	8
	8	17	5	80	10	22
2004	9	9	2	57	7	28
	10	23	6	138	14	32
2005	11	7	4	234	9	47
	12	28	7	309	5	45
2006	13	24	3	260	11	39
	14	27	4	258	21	56
2007	15	42	5	267	16	61
	16	33	3	245	17	33
2008	17	20	7	121	10	40
	18	7	3	171	18	80
2009	19	38	2	156	28	106
	20	26	2	141	20	143
2010	21	19	2	125	31	133
	22	23	9	167	30	88
2011	23	41	1	146	24	117
	24	21	2	68	41	80
2012	25	22	0	128	17	65
	26	51	4	114	8	44
2013	27	26	2	206	8	103
Totals		603	149	3929	397	1479

Table 16. Historical rate of vulnerability occurrences per platform

According to Table 16, software platforms have different rates of vulnerability occurrences. For example, in Linux (LIN), the rate of vulnerability occurrences is steadily increasing as semesters progress with an observed peak in the second semester of 2005 whereas in the case of Microsoft Internet Information Services (ISS), decreases.

The rate of vulnerability occurrences is used in section 4.3 to perform vulnerability prediction for each software platform as means to indicate trends in vulnerability occurrences. In the following Table 17, the frequencies of recorded occurrences are presented for APA, in terms of CVSS base metric group, from the first semester of 2000 until the first semester of 2013 according to the NVD 2.2v (National Vulnerability Database, 2013).

Metric name	Metric value	Frequencies
EC	*lw*	261
	me	300
	hi	42
EU	*mu*	0
	si	56
	no	547
EV	*lo*	57
	ad	0
	ne	546
IC	nn_c	341
	pa_c	220
	co_c	42
II	nn_i	214
	pa_i	347
	co_i	42
IA	nn_a	273
	pa_a	262
	co_a	68

Table 17. CVSS base metric group frequency of occurrences for APA

Based on Table 17, we compute an ABS to indicate what the vulnerability severity status of APA was. Hence, based on formulas (11) - (13), formula (2) is computed as follows:

$$Exploitability = 20*0.942*0.635*0.690 = 8.254 \quad (19)$$

Based on formulas (8) - (10), formula (3) is computed as follows:

$$Impact = 10.41*(1-(1-0.146)*(1-0.204)*(1-0.193)) = 4.705 \quad (20)$$

Based on formula (20), formula (4) is computed as follows:

$$f(Impact) = 1.176 \tag{21}$$

Hence, formula (1) is computed as follows:

$$BaseScore = round\_to\_1\_decimal(((0.6*4.705)+ (0.4*8.254)–1.5)*(1.176)) = 5.4 \tag{22}$$

Interpreting the result from formula (22), APA scores an ABS of 5.4 out of 10 from 2000 until the first semester of 2013. According to the Payment Card Industry (PCI, 2010), CVSS scores between 4.0 and 6.9 are ranked as medium severity scores and risk administrators should take prioritization actions starting from the most critical severity vulnerabilities.

5.3.3. Vulnerability prediction

We applied a distribution fitting procedure to formulate hypotheses about the reference probability distributions for the variable of interest (vulnerability occurrences). For this case, we used Matlab 8.1 to perform distribution fitting analysis (Mathworks, 2013). In the following Figures (32) – (36), we show the visual proximity among empirical probability distributions and Matlab-recommended reference distributions in terms of the CDF for each identified software platform. As described in the legend of each figure, the empirical probability distributions are demonstrated using the abbreviations of the software platforms followed by the word data, followed by the reference probability distributions that can differ from platform to platform.

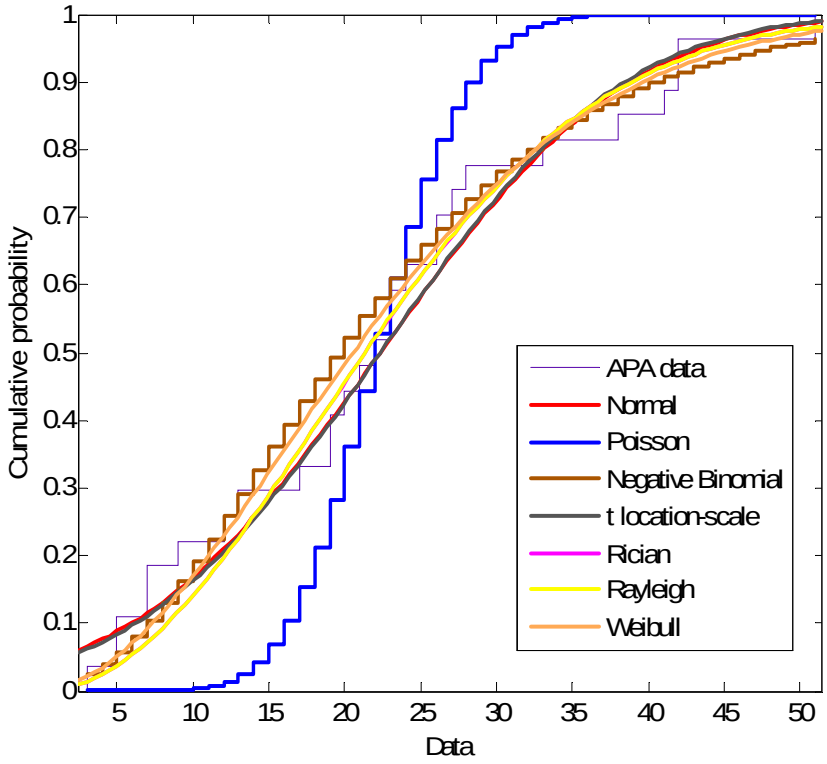

Figure 32. APA

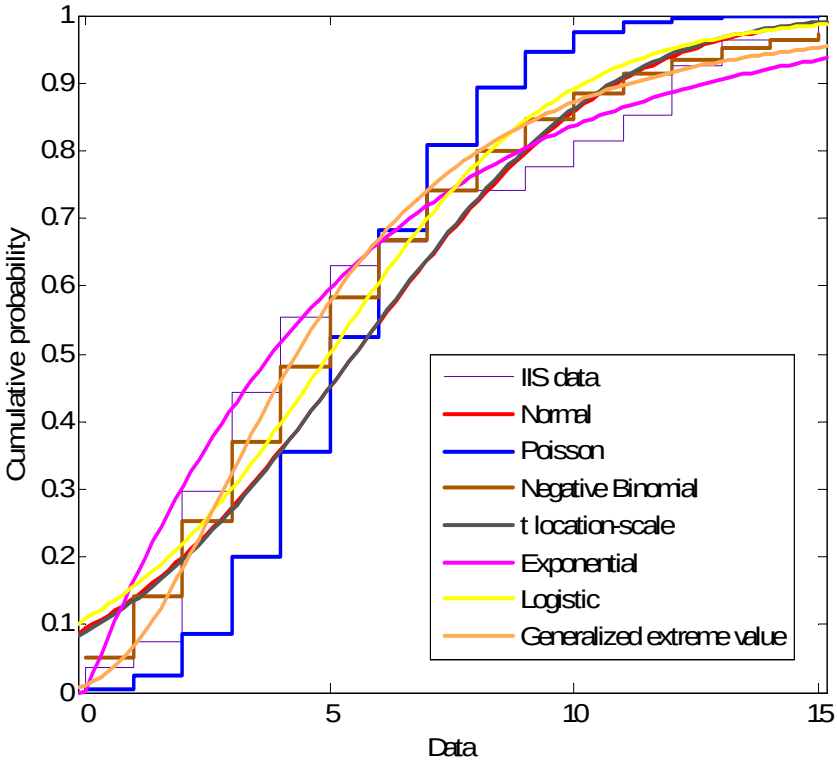

Figure 33. IIS

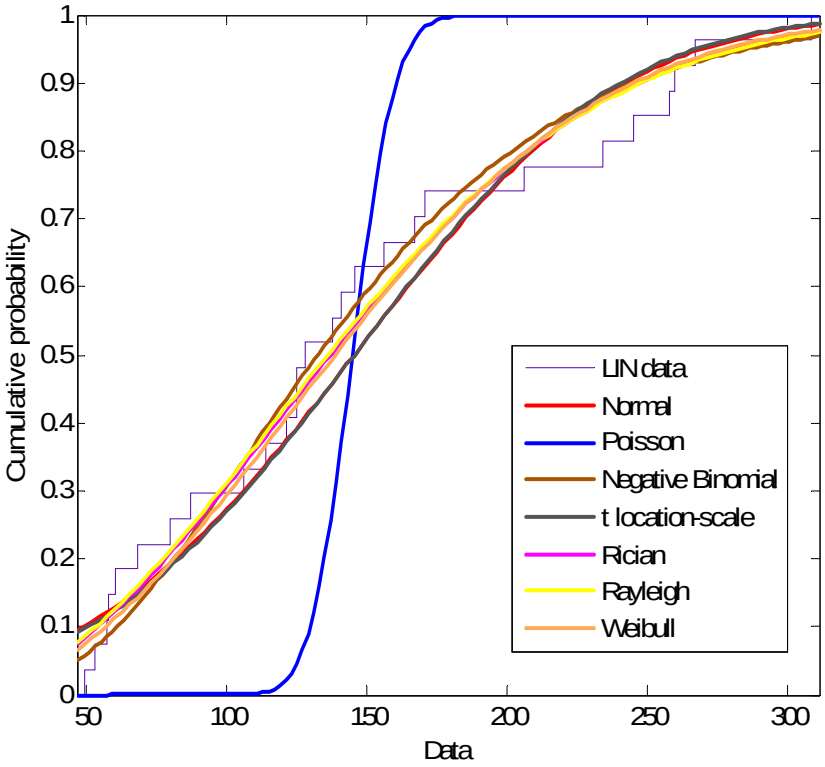

Figure 34. LIN

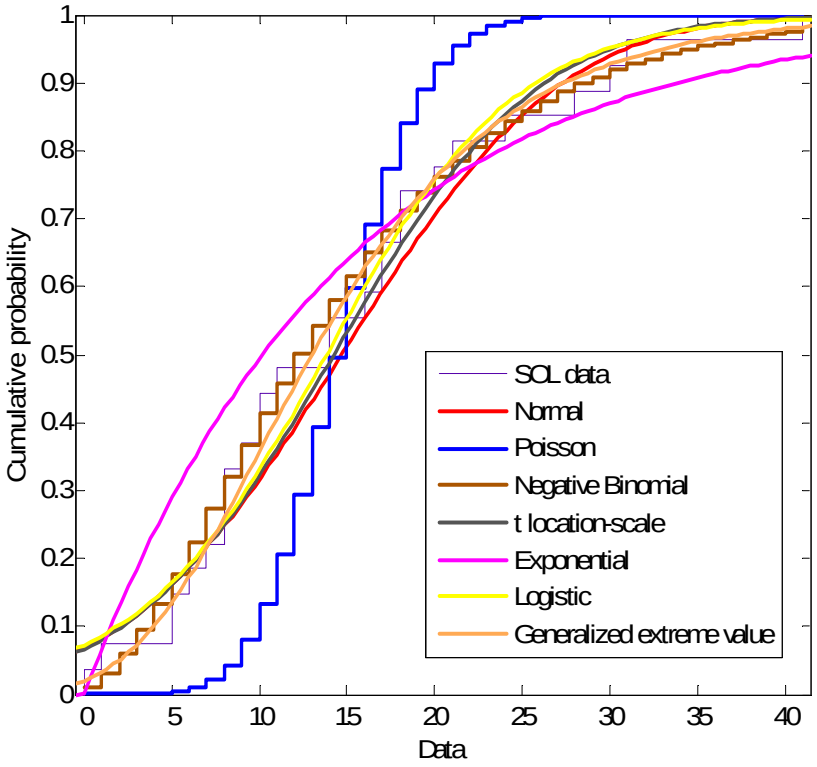

Figure 35. SOL

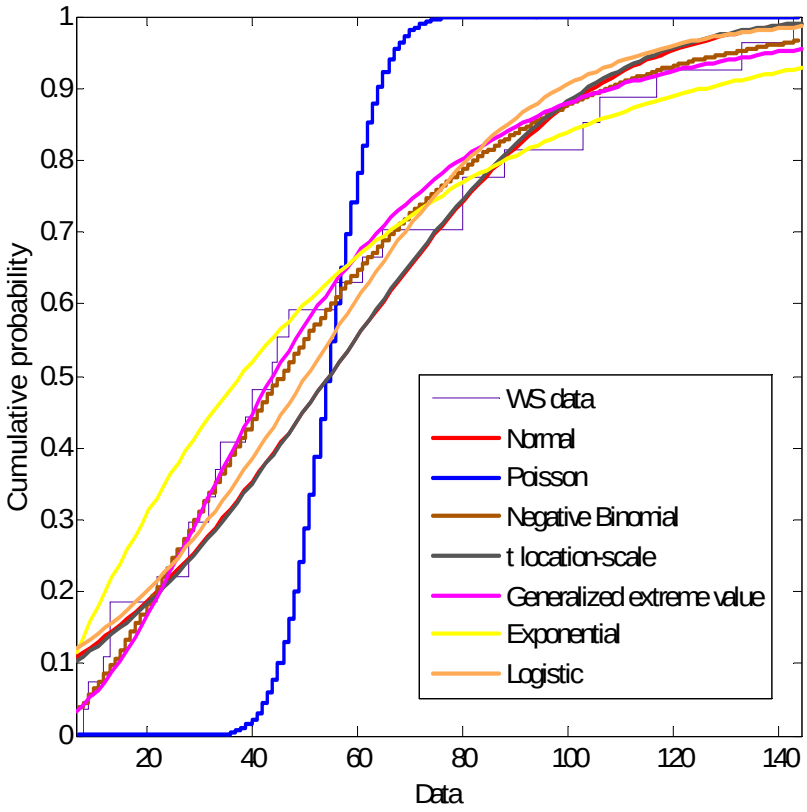

Figure 36. WS

Because a visual approximation is not enough to conclude which is the best fit reference probability distribution for the empirical data, we used the one-sample K-S test to verify distribution fitting results. The K-S test parameters are (Papoulis and Pillai, 2002):

- *h* which takes values either 0 or 1, where $h = 0$ means that the empirical probability distribution matches with the reference probability distribution (hypothesis is true), and $h = 1$ means does not match and thus the hypothesis is rejected,

- *p* or *p-value* denotes the probability of getting a *k* test statistic at least as high assuming $h = 0$,

- *k* test statistic is the statistic value on which the result on whether to accept or reject a hypothesis is based. In particular, it is the measure of the maximum difference between the reference and empirical distribution curves (Papoulis and Pillai, 2002).

In this respect, the criterion for comparison between the reference and empirical distributions is the lowest k value. For increased accuracy of results, the alpha value

for all the tests conducted was set at 0.01 instead of the Matlab's default value 0.05. The alpha value represents the probability that the test fails under the null hypothesis (Corder and Foreman, 2009; Marsaglia et al., 2003; Massey, 1951).

The following Table 15 shows which one of the 7 reference CDFs tested is the best fit for each software platform. The following abbreviations are used: ExP for Exponential, GeV for Generalized extreme value, LoG for Logistic, NeB for Negative Binomial, NoR for Normal, PoI for Poisson, RaY for Rayleigh, RiC for Rician, tl-s for t location-scale, WeI for Weibull.

APA							
	NoR	PoI	NeB	**tl-s**	RiC	RaY	WeI
h	0	0	1	0	0	0	0
p	0.8858	0.0292	3.09E-25	0.9051	0.8435	0.841	0.8
k	0.1068	0.2724	1	0.1037	0.1129	0.1133	0.1186
IIS							
	NoR	PoI	NeB	tl-s	ExP	**LoG**	GeV
h	0	0	1	0	0	0	1
p	0.2137	0.0654	1.07E-21	0.1807	0.0976	0.4545	1.11E-15
k	0.1973	0.2448	0.9259	0.2047	0.2299	0.1591	0.7842
LIN							
	NoR	PoI	NeB	tl-s	RiC	**RaY**	WeI
h	0	1	1	0	0	0	0
p	0.7288	2.5950E-005	3.8062E-0.5	0.7294	0.9507	0.9537	0.9363
k	0.1271	0.4415	1	0.127	0.0945	0.0938	0.0978
SOL							
	NoR	PoI	NeB	tl-s	ExP	**LoG**	GeV
h	0	0	1	0	0	0	1
p	0.7176	0.0081	1.44E-22	0.2929	0.1446	0.8349	1.37E-15
k	0.1284	0.3109	0.9447	0.1824	0.2142	0.1141	0.7818
WS							
	NoR	PoI	NeB	tl-s	GeV	ExP	**LoG**
h	0	1	1	0	1	0	0
p	0.3622	1.0228E-005	1.28E-003	0.3521	4.47E-14	0.3204	0.6845
k	0.1717	0.4587	0.522	0.1731	0.7417	0.178	0.1322

Table 18. K-S test results based on distribution fitting

The bold text cells indicate the best fit reference distributions for each platform. For example, according to Table 15, the empirical data for the Apache web server (APA) follow (closest fit) the t-location-scale (tl-s) distribution whereas the empirical data for the Microsoft Internet Information Services web server (IIS) follow the Logistic (LoG) distribution. Knowing which reference distribution has the best fit, we move on to the next step to calculate the probability of vulnerability occurrences in the near future.

5.3.4. Risk prediction

To initiate risk prediction, first, we calculate the probability of having an increased or decreased number of vulnerability occurrences in the following semester. For the purpose of this paper, we select the Apache web server (APA) to perform risk prediction. According to Table 15, the t location-scale CDF is the best fit and is based on the following formula (Mathworks, 2013):

$$F(x) = \frac{\Gamma\left(\frac{\delta+1}{2}\right)}{\sigma\sqrt{\delta\pi}\,\Gamma\left(\frac{\delta}{2}\right)}\left[\frac{\delta+\left(\frac{x-\mu}{\sigma}\right)^2}{\delta}\right]^{-\left(\frac{\delta+1}{2}\right)} \tag{23}$$

The t-location scale parameters are:

- x: the number of vulnerabilities
- Γ, which stands for the gamma function
- μ, which stands for the location parameter of the t location-scale distribution, stored as a scalar value
- σ, which stands for the scale parameter of the t location-scale distribution, stored as a positive scalar value
- δ, which stands for the degrees of freedom of the t location-scale distribution, stored as a positive scalar value

Given the distribution fitting results, μ = 22.33, σ = 12.5636 and v = 7350840. The t-location-scale distribution is useful for modeling data distributions with heavier tails (more prone to outliers) than the normal distribution. In this case, because ν approaches infinity, the t-location-scale distribution approaches the normal distribution (Hastie, 2009; Mathworks, 2013). This notion can be supported if one observes the minimum difference between t-location-scale and normal distribution as shown in Table 18.

Based on formulas (7) and (23) and given that the number of vulnerabilities of the first semester of 2013 for the APA correspond to x = 26 according to Table 16, we compute P_1 = 0.62 and P_2 = 1 – P_1 = 0.38. The results imply that the number of

vulnerabilities which will occur in the second semester of 2013 to be less than the first semester. Based on formulas (14) and (15), for the probability $P(T^+ \bigcap V^-)$, the number of all events (λ) is 28 semesters and the number of occurrences of an event with characteristic i (λ_i) is 7 semesters. For the probability $P(V^-)$, λ is the same as the numerator and the λ_i is 14 semesters, hence $P_4 = 0.5$.

The result imply that in the second semester of 2013, the event of having, at the same time, increased threat significance and decreased number of vulnerability occurrences is equal of it occurring or not. The same rationale is valid for the computation of the remaining probabilities as grouped in the following Table 19.

CPT(a)	CPT(b)	CPT(c)	CPT(d)	CPT(e)
$P_1 = 0.62$	$P_3 = 0.5$	$P_7 = 0.43$	$P_{11} = 0.57$	$P_{15} = 0.5$
$P_2 = 0.38$	$P_4 = 0.5$	$P_8 = 0.57$	$P_{12} = 0.43$	$P_{16} = 0.5$
	$P_5 = 0.46$	$P_9 = 0.62$	$P_{13} = 0.46$	$P_{17} = 0.54$
	$P_6 = 0.54$	$P_{10} = 0.38$	$P_{14} = 0.54$	$P_{18} = 0.46$

Table 19. CPTs results

Based on formula (14) and given formulas (8), (9) and (10), $P(D_c^+) = 0.5$, $P(D_i^+) = 0.46$ and $P(D_a^+) = 0.46$. The results imply that if the Impact does not vary in a great extent for security properties, then, confidentiality presents the most serious zero-day risk for the second semester of 2013.

5.4 Related work

Related work on vulnerability prediction has introduced an approach to predict trends or patterns in the number of vulnerabilities using historic data and a fuzzy logic application. The study highlights the problems of using vulnerability scanning technologies, either reactive or proactive, such as intrusion detection systems (IDS) or vulnerability scanners (VS) respectively, as inadequate and outdated (Venter and Eloff, 2004). Authors developed a vulnerability harmonization component as means to transform scan data into a set of harmonized vulnerability classes. This approach depends on scanning results to perform vulnerability prediction which may have a great amount of false positives. Moreover, contrary to our approach, there is no attempt to map vulnerability prediction with vulnerability severity or threat profile.

Research on vulnerability prediction focused on time and effort-based vulnerability discovery models (Woo et al., 2011). Authors compared the predictive capability of the models based on historical data retrieved from NVD for two HTTP servers, Apache and IIS. To determine future vulnerability patterns, this study is based on two types of server classification, the vulnerability cause and severity. Using regression analysis to obtain the parameter values for the chi-square test, the study produces

statistical results based on the classified vulnerabilities. Considerations of this study appear to be the representative sample which does not allow for generalization or comparison of results and the statistical accuracy of results. The chi-square test depends on an adequate sample size for the approximations to be valid whereas the kolmogorov-smirnov test is an exact test and does not depend on the underlying distribution sample. Moreover, the alpha level value was set to 5%, contrary to the 1% in our study for more sound statistical results.

Alhazmi et al., (2007) examined vulnerability prediction on a number of software platforms. Authors introduced a new metric, the vulnerability density, to estimate the number of vulnerabilities in Windows and Red Hat Linux platforms. The vulnerability discovery rate is assessed with a logistic and a linear model and is verified with the chi-square test to unveil future vulnerability occurrence patterns. In our work, we extend both the representative sample and the distribution models for a more in-depth examination on software platform vulnerability patterns.

Most recent research on vulnerability prediction focused on two web application vulnerabilities, SQL injection (SQLI) and cross site scripting (XSS) in PHP applications (Shar and Tan, 2013). Authors using machine learning techniques, proposed a vulnerability prediction framework which consists of a set of static code attributes. The attributes enable prediction of program statements, vulnerable to SQLI or XSS. The framework is evaluated with PhpMinerI, a tool for data collection, and results indicate promising figures in recall and low false alarm rates. Limitations may be considered that this study examines specialized vulnerabilities and this a_ects the generalization of results. Moreover, the vulnerability prediction is linked with program statements guiding auditors to report vulnerability as such, compared to our approach where we link vulnerability prediction with specific software-based platforms. Vulnerability prediction has also been measured with software component metrics such as complexity, coupling and cohesion (CCC). Authors experimented on the Mozilla Firefox platform different techniques such as decision tree, random forest, logistic regression and naive bayes as means to compare the predictive capabilities (Chowdhury and Zulkernine, 2011). With respect to the statistical significance of this study, it could become more fascinating to see how other platforms perform statistical projections and how CCC metrics behave in comparison to CVSS or other metrics towards vulnerability prediction. Another issue is the granularity of analysis which in this case is a file-level vulnerability prediction instead to our platform-level approach. In addition, research on vulnerability prediction conducted an empirical study on the predictive capabilities of the National Vulnerability Database (Zhang et al., 2011).

Authors used a data mining process based on construction schemes that contain time and version features to capture the characteristics of past vulnerabilities. As evaluation metrics they used parameter tuning based on correlation coefficient. The study concludes that the NVD have generally a poor prediction capability due to missing information, data errors, zero version differences and late vulnerability release times. However, authors claim that NVD data related to software applications may lead to reliable vulnerability prediction. In our research and based on the evaluation method regarding vulnerability prediction in section 6, we conclude that NVD possess adequate capability of vulnerability prediction for all identified platforms.

Another study, examined empirically whether fault prediction metrics such as complexity, code churn, and fault history can be used for vulnerability prediction purposes (Shin and Williams, 2013). Based on the Mozilla Firefox platform, authors conclude that both fault and vulnerability prediction models provide almost identical ability in vulnerability prediction across a wide range of classification thresholds. Authors tested a logistic regression technique, bayesian network, J48, and random forest to predict faulty and vulnerable files and concluded that all techniques have almost identical predictive capability. In this approach, the granularity of analysis is at the file-level and due to the sole platform examination the generalization of results is debatable.

Close to our approach in terms of risk prediction is the research work of Grunske and Joyce (2008), who presented a quantitative risk-based approach to predict security in component-based systems based on modular attack trees. The approach is demonstrated with an example that evaluates the confidentiality property of a distributed document management system. Authors use regression analysis to calibrate measures such as attacker motivation and ranking along with attack risk and cost. Considerations may be the lack of vulnerability measurement together with the attack profile as means to prioritize treatment actions. Moreover, it would be interesting to investigate how other security properties behave in comparison to confidentiality. Authors argue that there is a strong need for future research in risk prediction towards software-based assets. In the following Table 20, a comparative study of related work is presented.

References	Methods	Highlights	Considerations
Venter and Ellof, 2004	Fuzzy expected interval	Vulnerability harmonization components	Depends on scanning results
Woo et al., 2011	Regression analysis Chi-square test	Vulnerability patterns based on time/effort based models	Representative sample Models require a certain size of data to work
Alhazmi et al., 2007	Vulnerability density Chi-square test	Vulnerability discovery models through a linear and logistic model	Limited distribution models tested
Shar and Tan, 2013	Machine learning techniques	Proposed static attributes PhPMinerI data collection tool	Specialized vulnerabilities reported as program statements
Chowdhury and Zulkernine, 2011	Decision tree, Random forest, Logistic regression, Naïve bayes	Utilization of software component metrics (CCC)	Representative sample File-level analysis
Zhang et al., 2011	Data mining techniques	Prediction models for zero-day vulnerabilities	Limited use of CVSS metrics
Shin and Williams, 2013	Logistic regression, Bayesian network, J48, Random forest	Utilization of fault prediction metrics	Single platform examination File-level analysis
Grunske and Joyce, 2008	Regression analysis	Quantitative risk-based approach based on modular attack trees	Lack of mapping vulnerability with attack profile

Table 20. Related work comparative study

5.5 Evaluation

In this section, we develop and demonstrate a method to test the accuracy of vulnerability prediction, which is the core of the risk prediction methodology, based on the class interval technique (Salkind and Rasmussen, 2007). As depicted in Table 16, the number of vulnerability occurrences for every platform is recorded on a per semester basis. In a preliminary phase, for every platform we define a classification of the semesters (i.e. column titled "Semester class" in Tables 21-25) based on class intervals that are expressed with lower and upper limits of the number of recorded vulnerability occurrences (i.e. column titled "Vulnerability intervals"). However, the last vulnerability interval is open (e.g. in Table 21, the vulnerability interval of semester class C is "greater than 30") in order to capture the semesters with unexpected high number of vulnerability occurrences.

The number of semesters with recorded vulnerability occurrences that fall in a particular vulnerability interval is summed in column titled "Frequency" (e.g. in Table 21, 1 to 14 vulnerability occurrences have been recorded in eight semesters). Moreover, in an extra column titled "Probability intervals" we define the lower and upper limits of probability (in our case we use the P_2) to estimate the class of the following semester. For each semester class, probability limits are defined on the basis of frequency, such as the higher the frequency, the wider the probability limits (e.g. in Table 21, in Class B, the probability limits are wider compared to Class A or C).

Moreover, the main rationale for defining the probability limits is that, according to Table 16, when a semester presents a high number of vulnerability occurrences (e.g. greater than 30 for APA) that corresponds to a high level of semester class (e.g. C for APA) then a prediction for decreased vulnerability occurrences in the following semester, which is depicted in low values of P_2 ($P_2 < 0.5$), is to be expected while the semester class retains in a high level (e.g. C or B for APA) and vice versa. Hence, high valued probability intervals (e.g. 0.7 – 0.99 in Table 21) correspond to low valued vulnerability intervals (e.g. 1-14 in Table 21).

Having defined the semester classes for each platform, as depicted in Tables 21-25, we proceed with applying the proposed evaluation method, which consists of two-phases: a) the P_2 test and b) the class prediction. First, for each semester (e.g. the second semester of 2001) and for every platform (e.g. APA), the P_2 is computed based on equations (7) and (23). Second, the class of the following semester (e.g. the first semester of 2002) is predicted according to the probability interval the P_2 value falls in.

The results of applying the proposed method of evaluation are grouped in Table 26, which is an enriched version of Table 16 and has the following abbreviations: Y for year, S for serial number of semester, V for number of vulnerability occurrences, P for probability P_2, C for pair of classes. In the column C two values are recorded: the

left one is the class wherein the current semester (e.g. the 4th semester) falls in according to the vulnerability intervals and the right one is the class wherein the following semester (e.g. the 5th semester) falls in according to the probability intervals.

Semester class	Vulnerability intervals	Frequency	Probability intervals
A	1-14	8	0.70-0.99
B	15-29	13	0.30-0.69
C	>30	5	0-0.29

Table 21. Semester class for APA

Semester class	Vulnerability intervals	Frequency	Probability intervals
A	1-4	14	0.50-0.99
B	>5	12	0-0.49

Table 22. Semester class for IIS

Semester class	Vulnerability intervals	Frequency	Probability intervals
A	1-80	6	0.70-0.99
B	81-160	12	0.30-0.69
C	161-240	5	0.15-0.29
D	>241	5	0-0.14

Table 23. Semester class for LIN

Semester class	Vulnerability intervals	Frequency	Probability intervals
A	1-11	11	0.50-0.99
B	12-22	9	0.20-0.49
C	>23	6	0-0.19

Table 24. Semester class for SOL

Semester class	Vulnerability intervals	Frequency	Probability intervals
A	1-20	9	0.70-0.99
B	21-40	8	0.40-0.69
C	41-70	6	0.20-0.39
D	>71	5	0-0.19

Table 25. Semester class for WS

		APA			IIS			LIN			SOL			WS		
Y	S	V	P	C	V	P	C	V	P	C	V	P	C	V	P	C
2000	1	3	0.93	A/A	13	0.03	B/B	58	0.88	A/A	1	0.91	A/-	13	0.84	A/A
	2	13	0.77	A/A	15	0.01	B/B	53	0.90	A/B	0	0.92	A/A	12	0.85	A/A
2001	3	5	0.91	A/A	12	0.04	B/B	106	0.65	B/B	8	0.74	A/B	13	0.84	A/B
	4	12	0.79	A/A	11	0.07	B/A	87	0.75	A/A	18	0.31	B/A	28	0.73	A/A
2002	5	5	0.91	A/C	3	0.69	A/B	49	0.91	A/B	6	0.80	A/B	9	0.86	A/B
	6	42	0.05	C/B	12	0.04	A/A	125	0.55	B/A	14	0.49	B/A	34	0.67	B/A
2003	7	19	0.60	B/B	10	0.10	A/A	60	0.87	A/A	5	0.83	A/A	8	0.87	A/B
	8	17	0.66	B/A	5	0.49	B/A	80	0.78	A/A	10	0.66	A/A	22	0.78	A/B
2004	9	9	0.85	A/B	2	0.77	A/B	57	0.88	A/B	7	0.77	A/B	28	0.73	A/B
	10	23	0.47	B/A	6	0.39	B/A	138	0.48	B/C	14	0.49	B/A	32	0.69	B/C
2005	11	7	0.88	A/B	4	0.60	A/B	234	0.12	D/D	9	0.70	A/A	47	0.53	B/C
	12	28	0.32	B/B	7	0.30	B/A	309	0.02	D/D	5	0.83	A/A	45	0.56	B/B
2006	13	24	0.44	B/B	3	0.69	A/A	260	0.07	D/D	11	0.62	A/B	39	0.62	B/C
	14	27	0.35	B/C	4	0.60	A/B	258	0.08	D/D	21	0.21	B/B	56	0.90	A/C
2007	15	42	0.05	C/C	5	0.49	B/A	267	0.06	D/D	16	0.40	B/B	61	0.90	A/B
	16	33	0.19	C/B	3	0.69	A/B	245	0.10	D/B	17	0.35	B/A	33	0.68	B/B
2008	17	20	0.57	B/A	7	0.30	B/A	121	0.57	B/C	10	0.66	A/B	40	0.61	B/D
	18	7	0.88	A/C	3	0.69	A/A	171	0.33	B/B	18	0.31	B/C	80	0.20	C/D
2009	19	38	0.10	C/B	2	0.77	A/A	156	0.40	B/B	28	0.06	C/B	106	0.07	D/D
	20	26	0.38	B/B	2	0.77	A/A	141	0.47	B/B	20	0.24	B/C	143	0.01	D/D
2010	21	19	0	B/B	2	0.77	A/B	125	0.55	B/C	31	0.04	C/C	133	0.02	D/D
	22	23	0	B/C	9	0.15	B/A	167	0.35	B/B	30	0.04	C/C	88	0.15	D/D
2011	23	41	0.01	C/B	1	0.84	A/A	146	0.44	B/A	24	0.13	C/C	117	0.04	D/D
	24	21	0.93	B/B	2	0.77	A/-	68	0.84	A/B	41	0.01	C/B	80	0.20	C/C
2012	25	22	0.77	B/C	0	0.89	A/A	128	0.54	B/B	17	0.35	B/A	65	0.34	C/C
	26	51	0.91	C/B	4	0.60	A/A	114	0.61	B/C	8	0.74	A/A	44	0.57	B/D
2013	27	26	0.79	B/C	2	0.97	A/-	206	0.20	C/B	8	0.74	A/C	103	0.08	D/C
	28	62	0.91	C/C	0	0.89	A/A	116	0.60	B/C	25	0.11	C/B	62	0.37	C/B
2014	29	59	0.05	C/C	1	0.84	A/A	172	0.33	B/C	14	0.49	B/B	23	0.78	C/B
	30	50	0.60	C/-	1	0.84	A/-	189	0.26	C/-	20	0.24	B/-	31	0.71	A/B

Table 26. Application of the two-phase evaluation method per platform

The results from Table 26 are summarized in the following Table 27. Particularly, the accuracy of the vulnerability prediction results has been evaluated in terms of a) P_2, b) the exact class and c) the approximation class. The exact class examines whether the proposed prediction falls within the semester class proposed, such as A/A, whereas the approximation class examines whether the proposed prediction falls within an approximation semester class, such as A/B.

Software platform	P_2	Exact class	Approximation class	Total class score
APA	69%	13/29	14/29	27/29
IIS	72.4%	14/29	13/29	27/29
LIN	65.5%	15/29	13/29	28/29
SOL	69%	12/29	15/29	27/29
WS	69%	12/29	14/29	26/29

Table 27. Accuracy of results

According to Table 24, small variations between success rates imply that reliable measurements about forthcoming vulnerability occurrences can be achieved. In addition, since the threat and damage significance rely on qualitative characteristics of vulnerabilities, their evaluation is highly depended on the accuracy of the vulnerability prediction results.

5.6 Discussion

The proposed methodology aims at predicting zero-day risk through platform vulnerability analysis based on standardized and automated security specifications. Standalone vulnerability prediction approaches are deemed incomplete due to the inability to predict risk in a proactive manner. In this respect, based on existing vulnerability data accompanied with SCAP specifications, we build on the notion of vulnerability prediction to propose a risk prediction methodology based on SCAP specifications, as means to be part of a response and preparedness strategy to IIP.

The risk prediction methodology proposed has two roles: a) represents the cause and effect relationship among risk elements thereby allowing for tracking of results and b) acts as a real-time diagnostic and prediction tool showing the criticality of zero-day risk. This practice provides the competitive advantage because even the most planned maintenance operations can cause disruptions.

Limitations of our work can derive from interpreting the research results. Based on a BBN topology and accompanied CPTs, we provide warnings about which event is more probabilistically imminent to be realized, however, we do not relate this risk in terms of monetary value. This depends on factors such as the total cost of an asset, including maintenance, replacement and acquisition costs, the criticality of a process and stakeholders' requirements. Moreover, we do not relate risk prediction with adequacy of existing controls or selection of new. This should ideally take into consideration the overall cost of controls and business goals.

Another issue is the generalization of results that derive from the representative sample we used for vulnerability and risk prediction purposes. Particularly, we

focused on the electronic banking sector and the list of examined platforms is indicative and by no means exhaustive. However, our approach can be applied to other information infrastructures as well.

We believe that keeping CVSS metric and rating values and using frequencies to indicate whether the threat and damage significance increases or decreases, offers a clear advantage as it builds on the foundations of existing research (Houmb et al., 2010), maintains compliance with the CVSS and can also adapt to CVSS future versions. Moreover, our approach extends the work from Xie et al., (2010), who suggest using CVSS metrics, such as the EC metric, to compute the CPT parameters for threat structures that involve exploiting software vulnerabilities due to the uncertainty inherent in threat scenarios, such as unknown attack behavior or no guarantee of success upon vulnerability exploitation. In this perspective, current research (Younis et al., 2014) supports the reasoning of using the exploitability sub-score for threat structures however, considerations of using CVSS metrics, such as subjectivity of values, inspired authors to propose a new approach that assesses vulnerability exploitability based on two software properties, namely attack surface entry points and reachability analysis.

Another consideration is the use of software platforms to perform vulnerability and risk prediction. When a software platform carries a high number of vulnerability occurrences at a particular time period, implies that a specific software version may require more attention. Hence, if a software version is deemed critical, it is safer to focus attention on the criticality the version carries rather than the platform itself. It follows that one may consider using software versions to conduct vulnerability analysis for increased granularity of results, instead of the proposed software platform analysis.

Moreover, our approach is an effort to predict patterns in vulnerability occurrences at a platform level and it is out of the scope of this paper to identify the location of the vulnerability in the source code or perform analysis at a file-level. Finally, our approach to vulnerability and risk prediction is based on the NVD and CVSS combination. We believe that a different combination of vulnerability database and scoring system, in particular hybrid, may provide a different result to risk prediction.

5.7 Chapter summary

The principal aim of this research is to contribute to the proactive information security studies by assisting auditors manage zero-day risks. Conducting vulnerability and risk predictions that are developed on the basis of automated specifications means that vulnerability and risk problem areas, in the form of zero-day, can be attended before they occur. Future research projects are possible in the quest to compare and test the predictive capabilities of different vulnerability databases, such as the Open Source Vulnerability Database and Rapid7, or different vulnerability identifiers, such as USN and ELSA, as means to address which vulnerability database and identifier is most reliable for prediction purposes. Towards a holistic approach to vulnerability and risk

predictions, we believe that factors, such as compliance requirements and human participation, should also be examined as an integration of efforts useful to manage the overall uncertainty in an information infrastructure. It might be said that predicting the future is about exploring it to improve decision making today.

"We know what we are, but know not what we may be"
William Shakespeare

Chapter 6

6. Conclusions

In this dissertation, we have investigated on a holistic approach to enterprise management through the utilization of the GRC concept and provided a thorough analysis of its components including related work and approaches as presented in Chapter 2. Chapter 3 is concerned with automated security metrics and for this reason, specifications from the Security Content Automation Protocol and similar were presented. In addition, vulnerability scoring methods are described to assist in vulnerability severity measurement. In chapter 4, we elaborated on software trust and particularly we presented software development models that can improve on software maturity which is the road to achieving software trust. Moreover, software quality standards are presented and a case study that shows when particular software versions are mature enough so they can be trusted is presented. This chapter concludes with a literature review on software trust. In chapter 5, we introduced and utilized a novel risk prediction methodology for electronic banking information platforms through vulnerability analysis based on SCAP specifications. In this last chapter, we summarize the contribution of this dissertation and recommend future research areas.

6.1 Summary of the contributions

The contributions of this dissertation are multi-fold and presented as follows.

Holistic approach to enterprise management. Considering the complex landscape modern enterprises are required to thrive, we present the GRC concept and decompose it into its components in order to accumulate knowledge towards benefits and misfits for each particular component. Essentials, literature reviews and applicable methods are presented for the governance and risk management.

Security metrics. In need to protect important enterprise assets, measure the performance of a GRC program and monitor operational and security activities including compliance checking, we presented automated security metrics in terms of SCAP and similar specifications. Utilizing the proposed series of security metrics, the enterprise is liable to reap multiple benefits including isomorphic data analysis, automatic security content management and increased interoperability with other

enterprises. In the same context, we also presented vulnerability scoring methods to present the reader with the types and the methods that are applicable for vulnerability severity measurement.

Software trust. Pursuing software that is mature enough so they can be trusted, we elaborate on software trust. Particularly, we approach software maturity from the perspective of software development models and we present software quality standards that aim to improve refinement of software security practices. In addition, we conduct a case study on software trust by analyzing the behaviour of versions that constitute products of a software platform through vulnerability analysis. We introduce maturity points and intervals and propose when each version is considered trustworthy. We conclude this topic with a literature review on software trust.

Risk prediction. It is without a doubt that unforeseen risk hinders unpleasant surprises. In this respect and taking into account that precaution is better than cure, we develop a novel information infrastructure risk prediction methodology through platform vulnerability analysis and based on SCAP specification as a proactive approach to zero-day risks. The methodology consists of four distinct steps. First, we identify information infrastructure supporting platforms and then we measure the historical rate of vulnerability occurrences. Next, we use a distribution fitting procedure and a goodness of fit test to examine and verify vulnerability patterns respectively. Then, we model risk prediction based on conditional probability tables and estimate risk on security properties. The overall aim is to support auditors in controlling imminent risks by control optimization and resources allocation (Chatzipoulidis et al., 2015b).

6.2 Future work

In the following, we describe future work based on the research conducted in this dissertation.

Comparison of vulnerability databases and identifiers. In search for accurate predictive solutions in the context of information infrastructure protection, as presented in Chatzipoulidis et al., (2015b), an interesting proposition is to compare the capabilities of vulnerability databases, such as Rapid7 and OSVD, as well as vulnerability identifiers, such as USN and ELSA, in order to examine which one offers the more reliable predictive solution.

Business analysis of security controls. A successful enterprise is the one that is sustainable in the long term and optimizes existing resources. In this respect, a complementary research to risk prediction, as presented in chapter 5, is to link zero-day risk with selection of security controls on the basis of performance, cost and alternatives. In this way, the auditor could justify existing security controls and develop a strategy on the allocation of security controls based on zero-day risks.

Asset interdependencies. Protecting business and information assets is a priority for enterprises. Finding interdependencies among those assets and calculating how damage on an asset can affect other assets would bring multiple benefits in enterprise management. To achieve this, one needs to model those interdependencies and provide a formal method of measuring how damage is distributed from one asset to another.

6.3 Closing remarks

The research appeared in this dissertation has been presented in varying forms of journals and conferences papers (see Appendix). Particularly, the GRC concept is presented in (Chatzipoulidis and Mavridis, 2009; Chatzipoulidis and Mavridis, 2010a; Chatzipoulidis et al., 2010c). Adoption of electronic banking services and user behaviour is presented in (Chatzipoulidis and Mavridis, 2010b). Research on automated security metrics, including vulnerability scoring methods, appears in (Mavridis et al., 2012; Chatzipoulidis et al., 2014; Chatzipoulidis et al., 2015a). The utilization of automated security metrics and the risk prediction methodology are introduced in (Chatzipoulidis et al., 2015b).

Appendix

International journals

Information infrastructure risk prediction through platform vulnerability analysis

(co-authors: Dimitris Michalopoulos, Ioannis Mavridis)
International Journal of Systems and Software, Elsevier, 2015.
Impact Factor (2014): 1.245
Abstract: *The protection of information infrastructures is important for the function of other infrastructure sectors. As vital parts for the information infrastructure operation, software-based platforms, face a series of vulnerabilities and threats. This paper aims to provide a complementary approach to existing vulnerability prediction solutions and launch the measurement of zero-day risk by introducing a risk prediction methodology for an information infrastructure. The proposed methodology consists of four steps and utilizes the outcomes of a proper analysis of security measurements provided by specifications from the Security Content Automation Protocol. First, we identify software platform assets that support an information infrastructure and second we measure the historical rate of vulnerability occurrences. Third, we use a distribution fitting procedure to estimate the statistical correlation between empirical and reference probability distributions and verify the statistical significance of the distribution fitting results with the Kolmogorov-Smirnov test. Fourth, we develop conditional probability tables that constitute a Bayesian Belief Network topology as means to enable risk prediction and estimation on security properties. The practicality of the risk prediction methodology is demonstrated with an implementation example from the electronic banking sector. The contribution of the proposed methodology is to provide auditors with a proactive approach about zero-day risks.*

Book chapters

Managing Enterprise IT Risks through Automated Security Metrics

(co-authors: Dimitris Michalopoulos, Ioannis Mavridis)
Handbook of research chapter in "Automated enterprise systems for maximizing business performance", IGI Global.
Abstract: *Information systems of modern enterprises are quite complex entities. This fact has influenced the overall information technology (IT) risk profile of the enterprise and it has become all the more critical now to have sound information systems that can maximize business performance of an enterprise. At this point, the practical challenge for enterprises is how to manage enterprise IT risks for persistent protection of business and security goals. This chapter covers different aspects of managing enterprise IT risks, providing solutions in terms of risk management methods, automated security metrics and vulnerability scoring methods. The purpose is to introduce an in-depth study on*

enterprise IT risks and add value to enterprise sustainability through an extensive analysis of methods and automated security specifications.

Refereed papers in proceedings of international conferences and workshops

i. **Managing IT risk in the agriculture sector through automated security metrics**
(co-authors: Dimitrios Michalopoulos, Ioannis Mavridis)
In Proceedings of the International Workshop on Enterprise Information Systems and their Applications 2014, May, Tirana, Albania.
Abstract: *In a world of asymmetric development, the agriculture sector is as complex and dependent on other sectors' operability. This paper focuses on managing the information technology (IT) risk residing in the agriculture sector from distributed IT infrastructures. Inspired by the fact that the agriculture sector has not yet established a comprehensive process to deal with information systems' interoperability in terms of integrating data from distributed systems in a homogeneous way, we propose a series of specifications that automatically allow for proper identification, analysis, reporting on critical assets, systems, networks and other functions, such as checking policy compliance. Specifically, the research question is as follows: how can we integrate data into an automated process from distributed information systems without requiring an extensive investment in controls or human resources? The answer lies to standardized security metrics that will increase the productivity of an agriculture infrastructure, reduce uncertainty and imprecise results due to same syntax and semantics and at the same time provide guidelines on policy intervention, in terms of compliance and governance principles. In this respect, Security Content Automation Protocol (SCAP) and similar specifications are proposed as means to enable consistent data processing, comparison of results and automatically updates on security content.*

ii. **Utilization of security content for risk measurement**
(co-authors: Ioannis Mavridis, Christos Petridis)
Article in ISACA Newsletter 2012, August (In Greek)
Abstract: *The multidimensional nature of risk requires the use of appropriate tools for continuous and effective measurement. In this direction, the use of automated content security offered by the Security Content Automation Protocol (SCAP) and similar specifications is an ideal solution to reduce management costs and increase control of information systems' safety.*

iii. **Developing strategic perspectives for enterprise risk management towards information assurance**
In Proceedings of the 9th European Conference on Information Warfare and Security (ECIW'10), Thessaloniki, Greece, July 2010.
(co-authors: Ioannis Mavridis and Theodoros Kargidis)
Abstract: *Information is an important key business asset, which can exist in many forms, it involves various risks and it is essential that is suitably protected. Therefore, it requires the involvement of proper management ensuring that information assets are sufficiently secured and controlled. Truth is that the risk management discipline has received increasing attention in recent years due to increased regulations, ongoing changes and greater economic volatility that all affect the business*

environment. The purpose of a proper risk management action is to ensure transparency at all levels of the organization by taking the appropriate measures to reduce costs and manage financial, organizational and personal risk all at once, satisfying business objectives. However, due to misleading fallacies around its concept and the complexity that derive from governance, risk and compliance (GRC) activities, risk management falls short of assuring information assets. In this paper the results of our work on studying government, compliance and human factors in information security risk management are presented. The scope is to develop strategic perspectives around risk management implementation related to the concept of information security, helping minimize risks and cost. Sustaining security value over long term necessitates the realization of the information security lifecycle and the recognition of an imperative factor, the human involvement. Security spending remains a main concern despite the current economic crisis showing challenges that need to be confronted. Such challenges include maintaining a strong IT workforce, addressing growing foreign and domestic competition, developing critical infrastructure protection, balancing automated and manual controls and controlling intellectual property rights. The road ahead is the recognition of an enterprise risk management (ERM) strategy able to maintain security assurance and challenge ongoing changes that impact on the effectiveness of risk management. In addition, it is high time to consider a wider risk management approach, that of the societal risk management. For optimized results, the organization should foster a culture based on communication and feedback, recognizing training and security awareness a top priority. Creating a holistic picture of an enterprise as part of risk management and compliance efforts, it will provide a comprehensive platform for capturing and integrating multiple perspectives on processes, thus controlling information flow. Information assurance depends on the level of collaboration across internal and external parties and the correlation of disperse information. To avoid unpleasant circumstances, the risk management principle should engage into a dual approach of operability, that is maintaining performance and periodically re-evaluate itself to tackle with upcoming trends and risks.

iv. **A study on user behaviour and acceptance of electronic banking services**
In Proceedings of the Special Session on "Performance analysis of Computer Networks (PaCoNet)" organized in conjunction with the 14th Panhellenic Conference on Informatics (PCI, 2010), September, Tripoli, Greece.
(co-author: Ioannis Mavridis)
Abstract: *This paper presents a study which investigates user behaviour towards electronic banking (e-banking) and particularly in internet banking based on behavioural theoretical models and scales such as the theory of planned behaviour (TPB), the diffusion of innovations theory, the technology acceptance model (TAM) and Kirton's adaptor-innovator scale (KAI). In this study, behavioural and personality patterns lead to certain hypothesis regarding adoption towards internet banking. In addition, we categorize the most important factors affecting e-banking and propose the use of dependencies among different factors within the e-banking infrastructure in order to assess potential impacts and risks.*

v. **An ICT Security Management Framework**
In Proceedings of the International Conference on Security and Cryptography (SECRYPT 2010), July, Athens, Greece.
(co-author: Ioannis Mavridis)

Abstract: *Recently, organizations started to realize that managing information security is more than a software solution; it is a strategic discipline. This realization has emerged a major challenge in the business and technology field, the integration of all governance, risk, and compliance (GRC) activities to operate in synergy and balance in configuration with the business and security objectives. The goal of this paper is to develop a comprehensive ICT security management framework as a unified platform against the evolving GRC complexity. Considering the endemic nature of risk, the risk approach requires periodical rethinking in order to keep pace with security changes and prevent undesirable incidents while preserving the stakeholders' interests continuously. Such an approach depends on the risk management maturity level, and the portfolio of monitoring controls.*

vi. **Evolving challenges in information security compliance**
In Proceedings of the 4th Mediterranean Conference on Information Systems (MCIS 2009), September, Athens, Greece.
(co-author: Ioannis Mavridis)
Abstract: *With the proliferation of computer-driven organizations and internet-based business information systems, the need for security has increased significantly. In addition, information security compliance is becoming a controversial issue among IT professionals. This paper aims to address the concerns arising from compatibility of security standards, compliance cost, certification approval and human involvement that affect compliance management. A unified approach to information security compliance is suggested for organizations seeking to build strong relationships across business and IT departments, improving in that way a company's security value.*

References

Abdollahi, A., Afzali, M. 2014. A Single Sign-on based Integrated Model for E-banking Services through Cloud Computing. International Journal 3(1), 34-38.

Abdolmohammadi, M. J., Boss, S. R. 2010. Factors associated with IT audits by the internal audit function, International Journal of Accounting Information Systems 11(3), 140-151, doi:10.1016/j.accinf.2010.07.004.

Aburrous, M., Hossain, M. A., Dahal, K., Thabtah, F. 2010. Experimental Case Studies for Investigating E-Banking Phishing Techniques and Attack Strategies, Journal of Cognitive Computing 2(3), 242-253, doi:10.1007/s12559-010-9042-7.

Aduda, J., Kingoo, N. 2012. The Relationship between Electronic Banking and Financial Performance among Commercial Banks in Kenya, Journal of Finance and Investment Analysis 1(3), 99-118, ISSN:2241-0988.

Aggelis, V. C. 2005. The bible of e-banking, Athens: New Technologies Publications (in Greek), ISBN:9608105811.

Al-Ani, B., Marczak, S., Redmiles, D., Prikladnicki, R. 2014. Facilitating contagion trust through tools in Global Systems Engineering teams, Information and Software Technology 56(3), 309-320, doi:10.1016/j.infsof.2013.11.001.

Alhazmi, O. H., Malaiya, Y. K., Ray, I. 2007. Measuring, analyzing and predicting security vulnerabilities in software systems, Computers & Security 26(3), 219-228, doi:10.1016/j.cose.2006.10

Amoroso, E., Nguyen, T., Weiss, J., Watson, J., Lapiska, P., Starr, T. 1991. Toward an approach to measuring software trust, In Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, 198-218, doi: 10.1109/RISP.1991.130788.

Angelakopoulos, G., Mihiotis, A. 2011. E-banking: challenges and opportunities in the Greek banking sector, Electronic Commerce Research 11(3), 297-319, doi:10.1007/s10660-011-9076-2.

Arbaugh, W.A., Fithen, W. L., McHugh, J. 2000. Windows of Vulnerability: A Case Study Analysis, IEEE Computer.

Arena, M., Arnaboldi, M., & Azzone, G. 2010. The organizational dynamics of Enterprise Risk Management, Accounting, Organizations and Society 35(7), 659-675.

Arnold, V., Benford, T., Canada, J., Sutton, S. G. 2011. The role of strategic enterprise risk management and organizational flexibility in easing new regulatory compliance, International Journal of Accounting Information Systems 12(3), 171-188.

AS/NZS ISO 8402:1994. Quality management and quality assurance—Vocabulary.

Asnar, Y., Massacci, F. 2011. A Method for Security Governance, Risk, and Compliance (GRC): A Goal-Process Approach, Foundations of Security Analysis and Design VI , Lecture Notes in Computer Science 6858, 152-184, doi:10.1007/978-3-642-23082-0_6.

Assessment Summary Results (ASR), Retrieved May 2015 from http://measurablesecurity.mitre.org/incubator/asr/.

Asset Identification (AI), Retrieved May 2015 from http://csrc.nist.gov/publications/nistir/ir7693/NISTIR-7693.pdf.

Asset Reporting Format (ARF), Retrieved May 2015 from http://csrc.nist.gov/publications/nistir/ir7694/NISTIR-7694.pdf.

Asterix, 2014. SAMM Self Assessment tool, Retrieved May 2015 from https://labs.asteriskinfosec.com.au/tag/opensamm/.

Babar, M. A., Verner, J. M., Nguyen, P. T. 2007. Establishing and maintaining trust in software outsourcing relationships: An empirical investigation, Journal of Systems and Software 80(9), 1438-1449, doi:10.1016/j.jss.2006.10.038.

Bakari, J. K. 2007. A holistic approach for managing ICT security in non-commercial organizations. A case study in a developing country, PhD thesis, Stockholm University, Department of Computer and Systems Science.

Barnum, S. 2012. Cyber Observable eXpression (CybOX) Foundations, Retrieved May 2015 from https://cybox.mitre.org/documents/Cyber%20Observable%20eXpression%20%28CybOX%29%20Foundations%20-%20%28SwA%20Forum%20Spring%202012%29%20-%20Sean%20Barnum.pdf.

Barnum, S. 2014. Standardizing Cyber Threat Intelligence Information with the Structured Threat Information eXpression (STIX), Version 1.1, Revision 1, Retrieved May 2014 from https://stix.mitre.org/about/documents/STIX_Whitepaper_v1.1.pdf.

Benini M., Sicari S. 2008. Risk assessment in practice: A real case study, Journal of Computer Communications 31, 3691–3699.

Bilge, L., Dumitras, T. 2012. Before We Knew It, An Empirical Study of Zero-Day Attacks In The Real World, ACM Conference on Computer and Communications Security, USA, 833-844.

Bjorck, J. F. 2005, Disconvering Information security Management, PhD thesis, Report Series No. 05-010. Department of Computer and Systems Science, Stockholm University.

Bilge, L., Dumitras, T. 2012. Before We Knew It, An Empirical Study of Zero-Day Attacks In The Real World, ACM Conference on Computer and Communications Security, USA, 833-844.

Bonfè, M., Fantuzzi, C., Secchi, C. 2013. Design patterns for model-based automation software design and implementation, Control Engineering Practice 21(11), 1608-1619, doi:10.1016/j.conengprac.2012.03.017.

Booth, H., Halbardie, A. 2011. Trust Model for Security Automation Data 1.0 (TMSAD) NIST Interagency Report 7802, Retrieved May 2015 from http://csrc.nist.gov/publications/nistir/ir7802/NISTIR-7802.pdf.

Borrett, M. 2013. Compliance: keeping security interest alive, Computer Fraud & Security 2013(2), 5-6, doi:10.1016/S1361-3723(13)70017-4.

Brændeland, G., Refsdal, A., Stolen, K. 2010. Modular analysis and modelling of risk scenarios with dependencies, Journal of Systems and Software 83, 1995-2013, doi:10.1016/j.jss.2010.05.069.

British CCTA 2003. CRAMM (CCTA Risk Analysis and Management Method), Insight Consulting.

Bugiel, S., Davi, L. V. Schulz, S. 2011. Scalable trust establishment with software reputation, In Proceedings of the the sixth ACM workshop on Scalable trusted computing, New York, USA, 15-24, doi:10.1145/2046582.2046587.

Bunbury, P. 2009. Moving from compliance-based security to a risk-based security model, Computer Fraud & Security 9, 14-17, doi:10.1016/S1361-3723(09)70115-0.

Buttner, A. 2009. CPE Specification 2.2, Retrieved May 2015 from http://cpe.mitre.org/_les/cpe-specification_2.2.pdf.

Caron, F., Vanthienen, J., Baesens, B. 2013. A comprehensive investigation of the applicability of process mining techniques for enterprise risk management, Computers in Industry 64(4), 464-475.

Casey, V. 2010. Developing Trust In Virtual Software Development Teams, Journal of Theoretical and Applied Electronic Commerce Research 5(2), 41-58, doi:10.4067/S0718-18762010000200004.

Cavano, J. P. 1984. Software reliability measurement: Prediction, estimation, and assessment, Journal of Systems and Software 4(4), 269-275, doi:10.1016/0164-1212(84)90026-8.

Chartered Institute of Management Accountants, 2006. Code of Ethics for Professional Accountants, Retrieved May 2015 from http://www.cimaglobal.com/Documents/ImportedDocuments/external_audit_guidelines_practical_experience_04.pdf.

Chatzipoulidis A., Michalopoulos D., Mavridis I. 2015b. Information infrastructure risk prediction through platform vulnerability analysis, Journal of Systems and Software, Impact Factor 2013: 1.245, doi:10.1016/j.jss.2015.04.062.

Chatzipoulidis, A., Mavridis, I. 2009. Evolving challenges in information security compliance, In Proceedings of the 4th Mediterranean Conference on Information Systems (MCIS), Athens, Greece.

Chatzipoulidis, A., Michalopoulos, D., Mavridis, I. 2014. Managing IT risk in the agriculture sector through automated security metrics, International Workshop on Enterprise Information Systems and their Applications, May 15, Tirana, Albania.

Chatzipoulidis, A., Michalopoulos, D., Mavridis, I. 2015a. Managing Enterprise IT Risk through Automated Security Metrics, Handbook of research chapter in "Automated enterprise systems for maximizing business performance", IGI Global.

Chatzipoulidis, A., Mavridis I. 2010b. A study on user behaviour and acceptance of electronic banking services, In Proceedings of the Special Session on "Performance analysis of Computer Networks (PaCoNet)" Tripoli, Greece.

Chatzipoulidis, A., Mavridis I., Kargidis T. 2010c. Developing strategic perspectives for enterprise risk management towards information assurance, In Proceedings of the 9th European Conference on Information Warfare and Security (ECIW), Thessaloniki, Greece.

Chatzipoulidis, A., Mavridis, I. 2010a. An ICT security management framework, In the Proceedings of the International Conference on Security and Cryptography (SECRYPT), Poster, Athens, Greece.

Cheffins, B. R. 2011. The History of Corporate Governance. OXFORD HANDBOOK OF CORPORATE GOVERNANCE, Mike Wright, Donald Siegel, Kevin Keasey and Igor Filatotchev, eds., Oxford University Press, 2013; University of Cambridge Faculty of Law Research Paper No. 54/2011; ECGI - Law Working Paper No. 184/2012. doi:10.2139/ssrn.1975404.

Cheikes A. B., Waltermire, D., Scarfone, K. 2011. Common Platform Enumeration: Naming Specification Version 2.3, NIST Interagency Report 7695, Retrieved May 2015 from http://csrc.nist.gov/publications/nistir/ir7695/NISTIR-7695-CPE-Naming.pdf.

Chowdhury, I., Zulkernine M. 2011. Using complexity, coupling, and cohesion metrics as early indicators of vulnerabilities, Journal of Systems Architecture 57(3), 294-313, doi:10.1016/j.sysarc.2010.06.003.

Chrissis, M.B., Konrad, M., Shrum, S. 2006. CMMI: Guidelines for Process Integration and Product Improvement, 2nd ed., Addison-Wesley, Boston, MA, USA.

CLUSIF, 2009. Risk management, concepts and methods. White paper, Retrieved May 2015 from http://www.clusif.asso.fr/fr/production/ouvrages/pdf/CLUSIF-risk-management.pdf.

Committee of Sponsoring Organizations of the Treadway Commission (COSO) 2004. Enterprise Risk Management — Integrated Framework. Retrieved May 2015 from http://www.coso.org/documents/coso_erm_executivesummary.pdf.

Common Attack Pattern Enumeration and Classification (CAPEC), Retrieved May 2015 from http://capec.mitre.org.

Common Configuration Enumeration (CCE), Retrieved May 2015 from http://cce.mitre.org.

Common Configuration Scoring System (CCSS), Retrieved May 2015 from http://csrc.nist.gov/publications/nistir/ir7502/nistir-7502_CCSS.pdf.

Common Misuse Scoring System (CMSS), Retrieved May 2015 from http://csrc.nist.gov/publications/drafts/nistir-7517/Draft-NISTIR-7517.pdf.

Common Platform Enumeration (CPE), Retrieved May 2015 from http://cpe.mitre.org/.

Common Vulnerability and Exposures (CVE), Retrieved May 2015 from http://cve.mitre.org/.

Common Vulnerability Reporting Framework (CVRF), Retrieved May 2015 from http://www.icasi.org/cvrf.

Common Vulnerability Scoring System (CVSS), Retrieved May 2015 from http://www.first.org/cvss/cvss-guide.pdf.

Common Weakness Enumeration (CWE), Retrieved May 2015 from http://cwe.mitre.org/.

Common Weakness Risk Analysis Framework (CWRAF), Retrieved May 2015 from https://cwe.mitre.org/cwraf/.

Common Weakness Scoring System (CWSS), Retrieved May 2015 from http://cwe.mitre.org/cwss/.

Cook, A. T. 2012. Compliance and Security: Trading Partners, Encyclopedia of Supply Chain Management, Chapter 37, 153-161, Taylor and Francis Publ., doi:10.1081/E-ESCM-120048027.

Corder, G. W., Foreman, D. I. 2009. Nonparametric statistics for non-statisticians: a step-by-step approach. John Wiley and Sons Publ. ISBN:978-0-470-4546-9
Cyber Observable eXpression (CYBOX), Retrieved May 2015 from http://cybox.mitre.org/.

Da Veiga, A., Eloff, J. H. P. 2007. An Information Security Governance Framework. Information Systems Management 24(4), 361–372.

De Ru, W. G., Eloff, J. H. P. 1996. Risk analysis modelling with the use of fuzzy logic, Computers and Security 15(3), 239-48.

Diane, K. 2006. A study of design characteristics in evolving software using stability as a criterion, Software Engineering, IEEE Transactions on Software Engineering 32(5), 315,329, doi:10.1109/TSE.2006.42.

Dieter F., Huegle, T., Dortschy, M. 2013. A Model of Information Security Governance for E-Business, Electronic Business: Concepts, Methodologies, Tools, and Applications. IGI Global, 2143-2154, doi:10.4018/978-1-60566-056-1.ch133.

Dingsøyr, T., Nerur, S., Balijepally, V., Moe, N. B. 2012. A decade of agile methodologies: Towards explaining agile software development, Journal of Systems and Software 85(6), 1213-1221, doi:10.1016/j.jss.2012.02.033.

Druzdzel, J. M. 1996. Qualitative verbal explanations in Bayesian belief networks. Artificial Intelligence and Simulation of Behaviour Quarterly, Special issue on Bayesian networks 94, 43-54.

Er, M. J., Zhou, Y. 2008. Automatic generation of fuzzy inference systems via unsupervised learning, Neural Networks 21(10), 1556–1566.

European Telecommunications Standards Institute 2006. Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Methods and protocols; Part 1: Method and proforma for Threat, Risk, Vulnerability Analysis (ETSI TS 102 165-1 V4.2.1), Retrieved May 2015 from http://portal.etsi.org/mbs/Referenced%20Documents/ts_10216501v040201p.pdf.

Extensible Configuration Checklist Description Format (XCCDF), Retrieved May 2015 from http://nvd.nist.gov/scap/xccdf/docs/xccdf-spec-1.0.pdf.

eXtensible Name and Address Language (xNAL), Retrieved May 2015 from http://www.immagic.com/eLibrary/TECH/OASIS/XAL_V2.PDF.

Farrell, R. 2010. Securing the Cloud — Governance, Risk, and Compliance Issues Reign Supreme, Information Security Journal: A Global Perspective, 19:6, 310-319, doi:10.1080/19393555.2010.514655.

Fenton, N., Neil, M. 2011. The Use of Bayes and Causal Modelling in Decision Making, Uncertainty and Risk. Agena Risk White Paper, Cambridge, UK, Retrieved May 2015 from http://www.agenarisk.com/resources/white_papers/fenton_neil_white_paper2011.pdf.

Fitzgerald, T. 2011. Compliance Assurance, Encyclopedia of Information Assurance, Chapter 62, 524-531, Taylor and Francis Publ., doi:10.1081/E-EIA-120046824.

Fuenzalida, D., Mongrut, S., Arteaga, J. R., Erausquin, A. 2013. Good corporate governance: Does it pay in Peru?, Journal of Business Research 66(10), 1759-1770, doi:10.1016/j.jbusres.2013.01.008.

Gelbstein, E., Kellermann, T. 2012. ICT and Security Governance: Doing the Right Things the Right Way (and Well Enough). In Law, Policy, and Technology: Cyberterrorism, Information Warfare, and Internet Immobilization. IGI Global. 74-91, doi:10.4018/978-1-61520-831-9.ch006.

Gemma, A., Minero, M. 2013. IT Security Governance Legal Issues, IT Security Governance Innovations: Theory and Research. IGI Global, 47-73, doi:10.4018/978-1-4666-2083-4.ch003.

Gigerenzer, G. 2002. Calculated risks: How to know when numbers deceive you, Simon & Schuster, USA, ISBN:0-7432-5423-6.

Gikandi, J. W., Bloor, C. 2010. Adoption and effectiveness of electronic banking in Kenya, Electronic Commerce Research and Applications 9, 277-282, doi:10.1016/j.elerap.2009.12.003.

Giuffrida, R., Dittrich, Y. 2013. Empirical studies on the use of social software in global software development – A systematic mapping study, Information and Software Technology 55(7), 1143-1164, doi:10.1016/j.infsof.2013.01.004.

Gnedenko, B. V. 2005. The theory of probability and the elements of statistics, American Mathematical Society, Chelsea Publ. Comp., USA, 57-60.

Google, 2007. Severity Guidelines for Security Issues, Retrieved May 2015 from http://dev.chromium.org/developers/severity-guidelines.

Gordon, L. A., Loeb, M, P., Tseng, C.-Y. 2009. Enterprise risk management and firm performance: A contingency perspective, Journal of Accounting and Public Police 28(4), 301-327.

Gorschek, T., Tempero, E., Angelis, L. 2014. On the use of software design models in software development practice: an empirical investigation, Journal of Systems and Software, doi:10.1016/j.jss.2014.03.082.

Grace, M. F., Leverty, J. T., Phillips, R. D., Shimpi, P. 2014. The Value of Investing in Enterprise Risk Management. Journal of Risk and Insurance, doi:10.1111/jori.12022.

Gragido, W., Pirc, J. 2011. The Silent Killer: How Regulatory Compliance has Worsened the State of Information Security, In Cybercrime and Espionage, 35-47, doi:10.1016/B978-1-59749-613-1.00003-0.

Graydon, P. J., Kelly, T. P. 2013. Using argumentation to evaluate software assurance standards, Information and Software Technology 55, 1551-1562, doi:10.1016/j.infsof.2013.02.008.

Grembergen, W. V. 2004. Strategies for Information Technology Governance, IDEA Group Publishing.

Grunske, L., Joyce, D. 2008. Quantitative risk-based security prediction for component based systems with explicitly modeled attack profiles, Journal of Systems and Software 81(8), 1327-1345, doi:10.1016/j.jss.2007.11.716.

Halbardier, A., Waltermire, D., Johnson, M. 2011. Specification for the Asset Reporting Format 1.1, NIST Interagency Report 7694, Retrieved May 2015 from http://csrc.nist.gov/publications/nistir/ir7694/NISTIR-7694.pdf.

Hampton, J. J. 2015. Fundamentals of Enterprise risk management: How Top Companies assess risk, manage exposure, and seize opportunity, 2nd ed., USA, AMACOM Publication.

Harris, M., Furnell, S. 2012. Routes to security compliance: be good or be shamed?, Computer Fraud & Security 2012 (12), 12-20, doi:10.1016/S1361-3723(12)70122-7.

Hastie, T., Tibshirani, R., Friedman, J. 2009. The Elements of Statistical Data Mining, Inference, and Prediction, Springer Series of Statistics, 2nd Ed., 153-158, doi:10.1007/978-0-387-84858-7.

Hertzum, M. 2002. The importance of trust in software engineers' assessment and choice of information sources, Information and Organization 12(1), 1-18, doi:10.1016/S1471-7727(01)00007-0.

Houmb, S. H., Franqueira, V. N. L., Engum, E. A. 2010. Quantifying security risk level from CVSS estimates of frequency and impact, Journal of Systems and Software 83(9), 1622-1634, doi:10.1016/j.jss.2009.08.023.

Hoyt, E. R., Liebenberg, A. P. 2011. The Value of Enterprise Risk Management, Journal of Risk and Insurance 78(4), 795–822.

Huang, C.-C., Lin, F.-Y., Lin, F., Y.-S., Sun, Y. S. 2013. A novel approach to evaluate software vulnerability prioritization, Journal of Systems and Software 86(11), 2822-2840, doi:10.1016/j.jss.2013.06.040.

Humphreys, E. 2008. Information security management standards: Compliance, governance and risk management, Information Security Technical Report 13(4), November, 247-255, ISSN 1363-4127, http://dx.doi.org/10.1016/j.istr.2008.10.010.

IEC 61131-3:2013. The Fast Guide to IEC 61131-3 Open Control Standard & Software, Retrieved May 2015 from http://www.rtaautomation.com/iec61131-3/.

IEC 61508:2008. Functional safety of electrical/electronic/programmable electronic safety-related systems — Part 3: Software requirements, International Electrotechnical Commission (IEC), Retrieved May 2015 from http://webstore.iec.ch/Webstore/webstore.nsf/Artnum_PK/ 43984.

IFAC, 2004. Enterprise governance: getting the balance right, International Federation of Accountants, Professional Accountants in Business Committee, Retrieved May 2015, from www.ifac.org/Members/DownLoads/EnterpriseGovernance.pdf.

Ifinedo, P. 2012. Understanding information systems security policy compliance: An integration of the theory of planned behavior and the protection motivation theory, Computers & Security 31(1), 83-95, doi:10.1016/j.cose.2011.10.007.

Ifinedo, P. 2014. Information systems security policy compliance: An empirical study of the effects of socialization, influence, and cognition, Information & Management 51(1), 69–79, doi:10.1016/j.im.2013.10.001.

IIA, 2000, Internal Auditing: Adding Value across the Board, Corporate Brochure II.

IIA, 2010. International standards for the professional practice of internal auditing (standards), Retrieved May 2015 from https://na.theiia.org/standards-guidance/Public%20Documents/IPPF%202013%20English.pdf.

IIA, 2009. The role of internal auditing in resourcing the internal audit activity, Position paper, Retrieved May 2015 from https://na.theiia.org/standards-guidance/Public%20Documents/PP%20The%20Role%20of%20Internal%20Auditing%20in%20Resourcing%20the%20Internal%20Audit%20Activity.pdf.

ISAMM, 2002. Information Security Assessment and Monitoring Method, Retrieved December 2014 from http://rm-inv.enisa.europa.eu/methods_tools/m_isamm.html.

ISO 31000:2009 – Risk management.

ISO 2009. Risk Management Vocabulary, ISO Guide 73:2009.

ISO/IEC 15504-5:2012, Information technology -- Process assessment -- Part 5: An exemplar software life cycle process assessment model.

ISO/IEC 25010:2011. Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models. Geneva: International Organization for Standardization.

ISO/IEC 25012:2008, Software engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Data quality model.

ISO/IEC 27002:2005. Information technology -- Security techniques -- Code of practice for information security management.

ISO/IEC 27005:2008 - Information Technology -- Security techniques -- Information security risk management.

ISO/IEC 38500:2008 Corporate governance of information technology.

ISO/IEC 9126:1991. Software product evaluation – Quality characteristics and guidelines for use.

ISO/IEC 9126-1:2001. Software engineering-product quality – Part 1: Quality model.

ISO/IEC/IEEE 12207:2008 Standard for Systems and Software Engineering - Software Life Cycle Processes. doi:10.1109/IEEESTD.2008.4475826.

IT Governance Institute, 2006. Information Security Governance, Guidance for Boards of Directors and Executive Management, 2nd edn. Rolling Meadows, IL.

IT Governance Institute, 2007. COBIT 4.1 Excerpt: Executive Summary – Framework, Retrieved May 2015 from http://www.isaca.org/KnowledgeCenter/cobit/Documents/COBIT4.pdf.

Johnson, M. E., Goetz, E. 2007. Embedding Information Security into the Organization, IEEE Computer Society, Retrieved May 2015 from http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&arnumber=4218547&isnumber=4218538.

Jung, H.-W. Kim, S.-G., Chung, C.-S. 2004. Measuring software product quality: a survey of ISO/IEC 9126, IEEE Software, 88-92, Retrieved May 2015 from http://dis.unal.edu.co/~icasta/ggs/Documentos/Normas/9126.pdf.

Karagiorgos, T, Drogalas, G., Gotzamanis, E., Tampakoudis, I., 2010, Internal Auditing as an effective tool for corporate governance, Journal of Business Management 2(1), 15-23.

Keuzenkamp, H. A. 2000. Probability, Econometrics and Truth: The methodology of econometrics, Cambridge University Press, 35-38.

Khatri, A., Budhiraja, I. 2013. E-banking: Cryptography with unique identity, International Journal of Research in IT & Management 3(1), 78-85, ISSN:2231-4334

Kirillov, I., Chase, P., Beck, D., Martin, R. 2010. Malware attribute enumeration and characterization, MITRE Corporation, Received May 2015 from http://maec.mitre.org/about/docs/Introduction to MAEC white paper.pdf.

Knechel, W. R. 2007. The business risk audit: Origins, obstacles and opportunities, Accounting, Organizations and Society 32(4-5), 383-408, doi:10.1016/j.aos.2006.09.005.

Koch, S. 2005. Free/open Source Software Development, IGP, USA, ISBN:1-59140-369-3.

Kolkowska, E., Dhillon, G. 2013. Organizational power and information security rule compliance, Computers & Security 33, 3-11, doi:10.1016/j.cose.2012.07.001.

Kondabagil, J. 2007. Risk Management in Electronic Banking. Concepts and Best Practices, John Wiley and Sons, Asia, Singapore, ISBN:978-0-470-82243-2.

Koons, J., Minoli, D. 2010. Information Technology Risk Management in Enterprise Environments, A Review of Industry Practices and a Practical Guide to Risk Management Teams, John Wiley & Sons, ISBN:978-0-471-76254-6.

Korhonen, J., Hiekkanen, K., Mykkänen, J. 2013. Information Security Governance, Strategic and Practical Approaches for Information Security Governance: Technologies and Applied Solutions. IGI Global, 53-66, doi:10.4018/978-1-4666-0197-0.ch004.

Kowalski, S. 1994. IT insecurity: a multi-disciplinary inquiry, PhD thesis, Department of Computer and Systems Sciences, University of Stockholm.

Kritzinger, E., Von Solms, S. H. 2006. E-learning: incorporating information security governance, Issues in Informing Science and Information Technology 3, 319-325.

Kuznets, S. 1955. Economic growth and income inequality. American Economic Review 49, 1-28.

Lai, Y., Zhang, W., Yang, Z. 2014. Research on Software Trust Analysis Based on Behavior, IEICE TRANSACTIONS on Information and Systems, Vol.E97-D, No.3, 488-496, Online ISSN: 1745-1361.

Lam, J. 2014. What is ERM?, in Enterprise Risk Management: From Incentives to Controls, 2nd ed., John Wiley & Sons, Inc., Hoboken, NJ, USA.

Landoll, D. J. 2006. The Security Risk Assessment Handbook: A Complete Guide for Performing Security Risk Assessments, 2nd ed., Boca Raton: Auerbach.

Le Grand C. H. 2013. EDPACS: The EDP Audit, Control, and Security Newsletter, 47(4), 1-10, doi:10.1080/07366981.2013.775792.

Lee, C. 1990. Fuzzy logic in control systems: fuzzy logic controllers—parts I and II, IEEE Transactions on Systems, Man, and Cybernetics 20, 419–435.

LeMay, E., Scarfone, K., Mell, P. 2012. The Common Misuse Scoring System (CMSS): Metrics for Software Feature Misuse Vulnerabilities, Retrieved May 2015 from http://nvlpubs.nist.gov/nistpubs/ir/2012/NIST.IR.7864.pdf.

Li, J., Li, M., Wu, D., Song, H. 2012. An integrated risk measurement and optimization model for trustworthy software process management, Information Sciences 191, 47-60, doi:10.1016/j.ins.2011.09.040.

Liu, Q., Zhang, Y. 2011. VRSS: A new system for rating and scoring vulnerabilities. Computer Communications 34(3), 264-273, doi:10.1016/j.comcom.2010.04.006.

Liu, Q., Zhang, Y., Kong, Y., Wu, Q. 2012. Improving VRSS-based vulnerability prioritization using analytic hierarchy process, Journal of Systems and Software 85(8), 1699-1708, doi:10.1016/j.jss.2012.03.057.

Losavio, F., Chirinos, L., Matteo, A., Levy, N., Ramdane-Cherif, A. 2004. ISO quality standards for measuring architectures. Journal of Systems and Software 72, 209–223.

Magdaleno, A. M., Werner, C. M. L., Araujo, R. M. 2012. Reconciling software development models: A quasi-systematic review, Journal of Systems and Software 85(2), 351-369, doi:10.1016/j.jss.2011.08.028.

MAGERIT, 2006. Methodology for Information Systems Risk Analysis and Management – version 2, Retrieved May 2015 from https://www.ccn-cert.cni.es/publico/herramientas/pilar44/en/magerit/index.html.

Malware Attribute Enumeration and Characterization (MAEC), Retrieved May 2015 from http://maec.mitre.org/.

Mann, D. 2008. An Introduction to the Common Configuration Enumeration, Retrieved May 2015 from https://cce.mitre.org/documents/Introduction_to_CCE_White_Paper_July_2008.pdf.

Mannaert, H., Verelst, J., Ven, K. 2011. The transformation of requirements into software primitives: Studying evolvability based on systems theoretic stability, Science of Computer Programming 76(12), 1210-1222, doi:10.1016/j.scico.2010.11.009.

Mannaert, H., Verelst, J., Ven, K. 2012. Towards evolvable software architectures based on systems theoretic stability. Software: Practice and Experience 42, 89–116, doi: 10.1002/spe.1051.

Marsaglia, G., Tsang, W., Wang, J. 2003. Evaluating Kolmogorov's Distribution, Journal of Statistical Software 8(18), 1-4, ISSN:1548-7660.

Martin, A. R., 2013. Introduction to CWRAF 0.8.3 Version, Retrieved May 2015 from https://cwe.mitre.org/cwraf/introduction.html#relationships.

Martin, B., Coley, S. C. 2014. CWSS version: 1.0.1, Retrieved May 2015 from https://cwe.mitre.org/cwss/cwss_v1.0.1.html.

Martín, Y.-S., Yelmo, J. C. 2014. Guidance for the Development of Accessibility Evaluation Tools Following the Unified Software Development Process, Procedia Computer Science 27, 302-311, doi:10.1016/j.procs.2014.02.033.

Martinez, A. F., Troncoso, A., Riquelme, J. C. 2009. Improving Time Series Forecasting By Discovering Frequent Episodes in Sequences, Advances in Intelligent Data Analysis VIII Lecture Notes in Computer Science 5772, Springer-Verlag, 357-368, doi:10.1007/978-3-642-03915-7_31.

Masera, M., Fovino, I. N. 2007. A Service-Oriented Approach for Assessing Infrastructure Security, Critical Infrastructure Protection, IFIP International Federation for Information Processing 253, Springer US, 367-379, doi:10.1007/978-0-387-75462-8_26.

Mason, C., Kirkbride, J., Bryde, D. 2007. From stakeholders to institutions: the changing face of social enterprise governance theory, Management Decision 45(2), 284-301, Emerald, doi:10.1108/00251740710727296.

Massey, F. J. 1951. The Kolmogorov-Smirnov Test for Goodness of Fit, Journal of the American Statistical Association 46(253), 68-78, doi:10.1080/01621459.1951.10500769.

Mathrani, S., Mathrani, A. 2013. Utilizing enterprise systems for managing enterprise risks, Computers in Industry 64(4), 476-483, doi:10.1016/j.compind.2013.02.002.

Mathworks, 2013. Matlab, English version 8.1, Release 2013a, Natick, Massachusetts, ISBN: 978-0-9825838-8-3.

Matsuo, Y. 2003. Prediction, Forecasting, and Chance Discovery, Advanced Information Processing, Springer-Verlag, 30-43, doi:10.1007/978-3-662-06230-2_3.

Mavridis, I., Chatzipoulidis, A., Petridis, C. 2012. Exploiting security content to measure risk, ISACA Newsletter 4(2).

Mell, P., Grance, T. 2002. Use of the Common Vulnerabilities and Exposures (CVE) Vulnerability Naming Scheme, NIST SP 800-51, Retrieved May 2015 from http://www.hsdl.org/?view&did=440983.

Mell, P., Scarfone, K., Romanosky, S. 2007. A Complete Guide to the Common Vulnerability Scoring System Version 2, Retrieved May 2015 from http://www.first.org/cvss/cvss-guide.pdf.

Mermod, A. Y. 2011. Customer's perspectives and risk issues on e-banking in Turkey; Should We Still be Online. Journal of Internet Banking and Commerce, 16(1), 2011-04.

Microsoft, 2012. Security Bulletin Severity Rating System, Retrieved May 2015 from http://technet.microsoft.com/en-us/security/gg309177.aspx.

MITRE, 2006. An Introduction to the OVAL Language, Version 5.0, Retrieved May 2015 from http://oval.mitre.org/documents/docs-06/an_introduction_to_the_oval_language.pdf.

Moulton, R., Coles, R. S. 2003. Applying Information Security Governance. Computers & Security 22(7), 580–584

Mozilla, 2005. Mozilla Foundation Security Advisories, Retrieved May 2015 from http://www.mozilla.org/security/announce/.

Mu, Y. 2003. E-banking: Status, trends, challenges and policy implications. Trends, Challenges and Policy Implications.

Munisi, G., Randøy T. 2013. Corporate governance and company performance across Sub-Saharan African countries, Journal of Economics and Business 70, 92-110, doi:10.1016/j.jeconbus.2013.08.003.

National Cyber Security Summit Task Force, 2004. Corporate Governance Report, Retrieved May 2015 from https://www.dhs.gov/sites/default/files/publications/csd-informationsecuritygovernance-acalltoaction-2004.pdf.

National Institute of Technology and Standards (NIST), 2008. Special Publication 800-55 Revision 1, Performance Measurement Guide for Information Security, Retrieved May 2015 from http://csrc.nist.gov/publications/nistpubs/800-55 -Rev1/ SP800-55-rev1.pdf.

National Vulnerability Database (NVD), 2013. Version 2.2, Retrieved May 2015 from http://nvd.nist.gov/.

Nicho, M. 2013. An Information Governance Model for Information Security Management. In D. Mellado, L. Enrique Sánchez, E. Fernández-Medina, & M. Piattini (Eds.) IT Security Governance Innovations: Theory and Research, 155-189, Hershey, PA: Information Science Reference. doi:10.4018/978-1-4666-2083-4.ch00.

Nickell, E. B., Roberts, R. W. 2013. Organizational legitimacy, conflict, and hypocrisy: An alternative view of the role of internal auditing, Critical Perspectives on Accounting, http://dx.doi.org/10.1016/j.cpa.2013.10.005.

Nikolic, B., Ruzic-Dimitrijevic, L. 2009. Risk assessment of information technology Systems, Issues in Informing Science & Information Technology 6.

Nikto2, 2010. Retrieved May 2015 from http://cirt.net/nikto2.

NIST, 2011. Special Publication 800-39, Managing Information Security Risk, Organization, Mission, and Information System View, Retrieved May 2015 from http://csrc.nist.gov/publications/nistpubs/800-39/SP800-39-final.pdf.

Norman, C. S., Rose, A. M., Rose, J. M. 2010. Internal audit reporting lines, fraud risk decomposition, and assessments of fraud risk, Accounting, Organizations and Society 35(5), 546-557, http://dx.doi.org/10.1016/j.aos.2009.12.003.

OASIS, 2001. Extensible Name and Address Language (xNAL) Standard Description Document for W3C DTD/Schema, Version 2.0.

OCTAVE, 2003. Operationally Critical Threat, Asset, and Vulnerability Evaluation, Retrieved May 2015 from http://www.cert.org/octave/approach_intro.pdf.

Open Group, 2009. TOGAG version 9.1, Retrieved May 2015 from https://www.opengroup.org/togaf/.

Open Vulnerability Assessment Language (OVAL), Retrieved May 2015 from http://oval.mitre.org/language/version5.10/.

Osunmuyiwa, O. 2013. Online Banking and the Risks Involved, Research Journal of Information Technology 5(2), 50-54, ISSN:2041-3106.

OWASP (2009). Software Assurance Maturity Model version 1.0 (SAMM).

Papoulis, A., Pillai, U. 2002. Probability, random variables and stochastic processes, McGraw Hill, 4th Ed., ISBN:0073660116.

Pasquinucci, A. 2007. Security, risk analysis and governance: a practical approach, Computer Fraud & Security 2007 (7), 12-14, doi:10.1016/S1361-3723(07)70091-X.

Paulk, M. C., Curtis, B., Chrissis, M. B., Weber, C. V. 1993. Capability Maturity Model for Software, Version 1.1.

Payment Card Industry (PCI), 2010. Approved Scanning Vendors, Program Guide, Reference 1.0, PCI DSS 1.2, Retrieved May 2015 from https://www.pcisecuritystandards.org/pdfs/asv_program_guide_v1.0.pdf.

Peltier, T. R. 2000. Facilated Risk Analysis Process (FRAP), Auerbach Publications.

Pink Elephant, 2008. IT service management tools: compatibility considerations, Retrieved May 2015 from https://www.pinkelephant.com/NR/rdonlyres/3C232863-4423-430E-B5C6-8358A2D217B9/4340/PinkVERIFYServiceWhitepaperV333.pdf.

Pressman, R. S. 2010. Software Engineering, A practitioner's approach, 7th ed., McGraw Hill, ISBN 978–0–07–337597–7.

PTA Technologies 2005. Practial Threat Analysis. Retrieved May 2015 from http://www.ptatechnologies.com/.

Qualys, 1999. Severities KnowledgeBase, Retrieved May 2015 from http://www.qualys.com/research/knowledge/severity/.

Quon, T. K., Zeghal, D., Maingot, M. 2012. Enterprise Risk Management and Firm Performance, Procedia - Social and Behavioral Sciences 62, 263-267.

Racz, N., Weippl, E., and Seufert, A. 2010. A Frame of Reference for Research of Integrated Governance , Risk & Compliance, In Proceedings of the 11th IFIP TC 6/TC 11 International Conference, 107-116.

Rai, A., Song, H., Troutt, M. 1998. Software quality assurance: an analytical survey and research prioritization. Journal of Systems and Software 40, 67-83.

Rao, H. R., Gupta, M., Upadhyaya S. J. 2007. Managing Information Assurance in Financial Services, IGI Publishing.

Rastogi, R., Von Solms, R. 2006. Information Security Governance a Re-definition. IFIP, vol. 193. Springer, Boston.

Redhat, 2005. Classification of Security Issues, Retrieved May 2015 from http://www.redhat.com/f/pdf/rhel4/SecurityClassification.pdf.

Rinaldi, S., Peerenboom, J., Kelly, T. 2001. Identifying, Undestanding and Analyzing Critical Infrastructure Interdependencies, IEEE Control Systems Magazine 21(6), 11-25.

Saleh, M. S., Alfantook, A. 2011. A new comprehensive framework for enterprise information security risk management, Applied Computing and Informatics 9(2), 107-118.

Salkind, N. J., Rasmussen, K. 2007. Encyclopedia of Measurement and Statistics, Sage Publications, doi:10.4135/9781412952644

Sarens, G., Beelde, I. D., Everaert, P. 2009. Internal audit: A comfort provider to the audit committee, The British Accounting Review 41(2), 90-106, doi:10.1016/j.bar.2009.02.002.

Sarigiannidis, L., Chatzoglou, P. D. 2014. Quality vs risk: An investigation of their relationship in software development projects, International Journal of Project Management 32(6), 1073-1082, doi:10.1016/j.ijproman.2013.11.001.

Scacchi, W., Feller, J., Fitzgerald, B., Hissam, S., Lakhani, K. 2006. Understanding Free/Open Source Software Development Processes, Software Process Improvement Practice 11, 95 – 105, doi:10.1002/spip.255.

Scarfone, K., Mell, P. 2010. The Common Configuration Scoring System (CCSS): Metrics for Software Security Configuration Vulnerabilities, NIST Interagency Report 750, Retrieved May 2015 from http://csrc.nist.gov/publications/nistir/ir7502/nistir-7502_CCSS.pdf.

Schiffman, M. 2011. The Common Vulnerability Reporting Framework, An Internet Consortium for Advancement of Security on the Internet (ICASI), Whitepaper, Version 1.0, Retrieved May 2015 from http://www.icasi.org/docs/cvrf-whitepaper.pdf.

Schiffman, M., CIAG, C. 2005. CVSS: A Common Vulnerability Scoring System. Retrieved May 2015 from http://www.first.org/cvss/v1/guide.

Schiffman, M. 2007. A Complete Guide to the Common Vulnerability Scoring System (CVSS). Retrieved May 2015 from http://www.first.org/cvss/cvss-guide.html.

Schryen, G., Volkamer, M., Ries, S., Habib, S. M. 2011. A formal approach towards measuring trust in distributed systems., in William C. Chu; W. Eric Wong; Mathew J. Palakal & Chih-Cheng Hung, ed., 'SAC', ACM, 1739-1745.

Secunia, 2002. Advisories, Retrieved May 2015 from http://secunia.com/advisories/historic/.

Shah, M. H., Siddiqui, F. A. 2006. Organisational critical success factors in adoption of e-banking at the Woolwich bank, International Journal of Information Management 26, 442–456, doi:10.1016/j.ijinfomgt.2006.08.003.

Shah, M., Clarke, S. 2009. E-Banking Management: Issues, Solutions and Strategies, IGI Global, New York, ISBN:978-1-60566-252-7.

Shar, L. K., Tan, H. B. K. 2013. Predicting SQL injection and cross site scripting vulnerabilities through mining input sanitization patterns, Information and Software Technology 55(10), 1767-1780, doi:10.1016/j.infsof.2013.04.002.

Sherwood, J., Clark, A., Lynas, D. 2009. Enterprise security architecture whitepaper, SABSA Limited.

Shin, Y., Williams, L. 2013. Can traditional fault prediction models be used for vulnerability prediction? Empirical Software Engineering 18(1), Springer US, 25-59, doi:10.1007/s10664-011-9190-8.

Siakas, K. V., Siakas, E. 2008. The need for trust relationships to enable successful virtual team collaboration in software outsourcing, International Journal of Technology, Policy and Management 8(1), 59-75, doi:10.1504/IJTPM.2008.016181.

Software Engineering Institute, 2010. CMMI for Development, Version 1.3, November, Technical report Retrieved May 2015 from http://www.sei.cmu.edu/reports/10tr033.pdf.

Software Identification Tags (SWIDs), Retrieved May 2015 from http://tagvault.org/swid-tags/.

Spanos, G., Sioziou, A., Angelis, L. 2013. WIVSS: a new methodology for scoring information systems vulnerabilities. In Proceedings of the 17th Panhellenic Conference on Informatics ACM, NY, USA, 83-90, doi:10.1145/2491845.2491871.

Spira, F. L., Page, M. 2003. Risk management: The reinvention of internal control and the changing role of internal audit, Accounting, Auditing & Accountability Journal 16(4), 640 – 661.

Steinbart, P. J., Raschke, R. L., Gal, G., Dilla, W. N. 2012. The relationship between internal audit and information security: An exploratory investigation, International Journal of Accounting Information Systems 13(3), 228-243, doi:10.1016/j.accinf.2012.06.007.

Stoel, D., Havelka, D., Merhout, J. W. 2012. An analysis of attributes that impact information technology audit quality: A study of IT and financial audit practitioners, International Journal of Accounting Information Systems 13(1), 60-79, doi:10.1016/j.accinf.2011.11.001.

Stoll, M., Breu, R. 2012. Information Security Governance and Standard Based Management Systems. In M. Gupta, J. Walp, & R. Sharman (Eds.) Strategic and Practical Approaches for Information Security Governance: Technologies and Applied Solutions (pp. 261-282). Hershey, PA: Information Science Reference. doi:10.4018/978-1-4666-0197-0.ch015.

Strecker, S., Heise, D., Frank, U. 2011. RiskM: A multi-perspective modelling method for IT risk assessment, Information Systems Frontiers 13(4), Springer-US, 595-611, doi:10.1007/s10796-010-9235-3.

Structured Threat Information, eXpression (STIX), Retrieved May 2015 from http://stix.mitre.org/

Symantec, 2000. Symantec Security Response Glossary, Retrieved May 2015 from http://www.symantec.com/security response/severityassessment.jsp.

Tagani, T., Sugeno, M., 1985. Fuzzy identification of systems and its applications to modeling and control, IEEE Transactions on Systems, Man, and Cybernetics, SMC-15, 116–132.

Tan, T. C. C., Ruighaver, A. B., Ahmad, A. 2010. Information Security Governance: When Compliance Becomes More Important than Security, In Proceedings of the 25th IFIP TC 11 International Information Security Conference, 55–67.

Tarantino, A. 2008. Governance, Risk, and Compliance Handbook: Technology, Finance, Environmental, and International Guidance and Best Practices, John Wiley and Sons.

Tariq, Y. B., Abbas, Z. 2013. Compliance and multidimensional firm performance: Evaluating the efficacy of rule-based code of corporate governance, Economic Modelling 35, 565-575, doi:10.1016/j.econmod.2013.08.015.

Taylor, L. 2014. Practical enterprise risk management: How to optimize business strategies through managed risk taking, Kogan Page Limited Publ. Technology 55(10), 1767-1780, doi:10.1016/j.infsof.2013.04.002.

Tekathen, M., Dechow, N. 2013. Enterprise risk management and continuous re-alignment in the pursuit of accountability: A German case, Management Accounting Research 24(2), 100-121.

The Open Checklist Interactive Language (OCIL), Retrieved May 2015 from http://nvd.nist.gov/ocil/OCIL_language.pdf.

Theoharidou, M., Kotzanikolaou, P., Gritzalis, D. 2010. A multi-layer criticality assessment methodology based on interdependencies, Computers & Security 29(6), 643-658, doi:10.1016/j.cose.2010.02.003.

Tiwari, S., Singh S. K. 2013. Information Security Governance Using Biometrics. In IT Security Governance Innovations: Theory and Research. IGI Global, 191-224, doi:10.4018/978-1-4666-2083-4.ch008.

Tompkins, F. G., Rice R. S. 1986. Integrating Security Activities into the Software Development Life Cycle and the Software Quality Assurance Process, Journal of Computers and Security 5, 218-242.

Trust Model for Security Automation Data (TMSAD), Retrieved May 2015 from http://csrc.nist.gov/publications/nistir/ir7802/NISTIR-7802.pdf.

Usman, A. K., Shah, M. H. 2013. Critical success factors for preventing e-banking fraud. Journal of Internet Banking and Commerce, 18(2).

Vance, A., Siponen, M., Pahnila, S. 2012. Motivating IS security compliance: Insights from Habit and Protection Motivation Theory, Information & Management 49 (3-4), 190-198, doi:10.1016/j.im.2012.04.002.

Venter, H. S., Eloff, J. H. P. 2004. Vulnerability forecasting-a conceptual model, Computers & Security 23(6), 489–497. doi:10.1016/j.cose.2004.06.005.

Veríssimo, P., Neves, N. F., Correia, M. 2006. The CRUTIAL reference critical information infrastructure architecture: A blueprint, Critical Information Infrastructures Security, Lecture Notes in Computer Science 4347, 1-14.

Von Solms, B. 2005. Information Security governance: COBIT or ISO 17799 or both?, Computers & Security 24(2), 99-104, doi:10.1016/j.cose.2005.02.002.

Von Solms R., R., von Solms, S. H. 2006. Information Security Governance: A model based on the Direct–Control Cycle, Computers & Security 25(6), 408-412, doi:10.1016/j.cose.2006.07.005.

Von Solms, S. B., Louwrens, C. B. 2006. The Relationship Between Digital Forensics, Corporate Governance, IT Governance, and IS Governance. In P. Kanellis, E. Kiountouzis, N. Kolokotronis, & D. Martakos (Eds.) Digital Crime and Forensic Science in Cyberspace, 243-266, Hershey, PA: Idea Group Publishing. doi:10.4018/978-1-59140-872-7.ch011.

Von Solms, S. H. 2005. Information Security Governance – Compliance management vs operational management, Computers & Security 24(6), 443-447, doi:10.1016/j.cose.2005.07.003.

Vroom, C., von Solms, R. 2004. Towards information security behavioural compliance, Computers & Security 23(3), 191-198, doi:10.1016/j.cose.2004.01.012.

VUPEN, 2005. Vulnerability Research and Intelligence, Retrieved May 2015 from http://www.vupen.com/.

Vyatkin, V., Zoitl, A. 2013. Advanced software engineering in industrial automation, Control Engineering Practice 21(11), 1606-1607, doi:10.1016/j.conengprac.2012.11.003.

Waltermire, D., Quinn, S., Scarfone, K., Halbardier, A. 2011. The Technical Specification for the Security Content Automation Protocol (SCAP): SCAP Version 1.2, Retrieved May 2015 from http://csrc.nist.gov/publications/nistpubs/800-126-rev2/SP800-126r2.pdf.

Waltermire, D., Schmidt, C., Scarfone, K. Ziring, N. 2011. Specification for the Extensible Configuration Checklist Description Format (XCCDF) Version 1.2, NIST Interagency Report 7275, Revision 4, Retrieved May 2015 from http://csrc.nist.gov/publications/nistir/ir7275-rev4/NISTIR-7275r4.pdf.

Wang, Y., Li, M. 2011. The Role of Internal Audit in Engineering Project Risk Management, Procedia Engineering 24, 689-694, doi:10.1016/j.proeng.2011.11.2719.

Wang, Y., Yang, Y. 2012. PVL: A Novel Metric for Single Vulnerability Rating and Its Application in IMS, Journal of Computational Information Systems 8(2), 579-590.

Weber, R. H. 2013. Internet of things – Governance quo vadis?, Computer Law & Security Review 29 (4), 341-347, doi:10.1016/j.clsr.2013.05.010.

Wheeler, E. 2011. Security Risk Management: Building an Information Security Risk Management Program from the Ground Up, 1st ed., Waltham: Syngress.

Wolfgang, M. 2002. Host Discovery with Nmap, Retrieved May 2015 from http://nmap.org/docs/discovery.pdf.

Woo, S.-W., Joh, H., Alhazmi, O. H., Malaiya, Y. K. 2011. Modeling vulnerability discovery process in Apache and IIS HTTP servers, Computers & Security 30(1), 50-62, doi:10.1016/j.cose.2010.10.007.

Wunder, J., Baker, J. 2010. Assessment Summary Results, MITRE, Retrieved May 2015 from http://measurablesecurity.mitre.org/incubator/asr/docs/Assessment_Summary_Results_Whitepaper.pdf.

Wunder, J., Halbardier, A., Waltermire, D., 2011. Specification for Asset Identification 1.1, NIST Interagency Report 7693, Retrieved May 2015 from http://csrc.nist.gov/publications/nistir/ir7693/NISTIR-7693.pdf.

X-Force, 1999. Frequently asked questions, Retrieved May 2015 from http://www-935.ibm.com/services/us/iss/xforce/faqs.html

Xie, P., Jason, H. L., Xinming, O., Peng, L., Renato, L. 2010. Using Bayesian Networks for Cyber Security Analysis, In Proceedings of the International Conference on Dependable Systems and Networks (DSN), IEEE/IFIP, 211-220, doi:10.1109/DSN.2010.5544924.

Yan, Z., MacLaverty, R. 2006. Autonomic Trust Management in a Component Based Software System, Autonomic and Trusted Computing, Lecture Notes in Computer Science Volume 4158, Springer, 279-292, doi:10.1007/11839569_27.

Yaokumah, W. 2013. Evaluating the Effectiveness of Information Security Governance Practices in Developing Nations: A Case of Ghana. IJITBAG 4(1) 27-43, IGI Global, doi:10.4018/jitbag.2013010103.

Yngström, L. 1996. A Systemic-Holistic Approach to Academic Programmes in IT Security, PhD Thesis, Department of Computer and Systems Sciences, University of Stockholm and the Royal Institute of Technology, Stockholm, ISBN: 91-7153-521-7.

Younis, A. A., Malaiya, Y. K., Ray, I. 2014. Using Attack Surface Entry Points and Reachability Analysis to Assess the Risk of Software Vulnerability Exploitability, IEEE 15th International Symposium on High-Assurance Systems Engineering (HASE), 1 - 8, doi:10.1109/HASE.2014.10.

Yuana, Y. Hanab, Q. 2011. A Software Behavior Trustworthiness Measurement Method based on Data Mining, International Journal of Computational Intelligence Systems 4(5), 817-825, doi:10.1080/18756891.2011.9727833.

Yuen, K. K. F., Lau, H. C. W. 2011. A fuzzy group analytical hierarchy process approach for software quality assurance management: Fuzzy logarithmic least squares method, Expert Systems with Applications 38(8), 10292-10302, doi:10.1016/j.eswa.2011.02.057.

Zhang, S., Caragea, D., Ou, X. 2011. An Empirical Study on Using the National Vulnerability Database to Predict Software Vulnerabilities, Database and Expert Systems Applications, Lecture Notes in Computer Science 6860, Springer-Verlag, 217-231, doi:10.1007/978-3-642-23088-2_15.

Zhu, K, Sun, H. 2012. The reform of accounting standards and audit pricing, China Journal of Accounting Research 5(2), 187-198, doi:10.1016/j.cjar.2012.05.002.

Ziring, N. 2005. Specification for the Extensible Configuration Checklist Description Format (XCCDF), NISTIR 7188, Retrieved May 2015 from https://nvd.nist.gov/scap/xccdf/docs/xccdf-spec-1.0.pdf.

Zuccato, A. 2005. Holistic Information Security Management Framework—Applied for Electronic Commerce, PhD thesis, Karlstad University Studies.