ROYAL INSTITUTE
OF TECHNOLOGY

A framework and theory for cyber security assessments

Teodor Sommestad

2012

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Industrial Information and Control Systems
KTH, Royal Institute of Technology
Stockholm, Sweden

TRITA-EE 2012:043
ISSN 1653-5146
ISRN KTH/ICS/R—12/03—SE
ISBN 978-91-7501-511-8

Stockholm 2012, Universitetsservice US AB

Abstract

Information technology (IT) is critical and valuable to our society. An important type of IT system is Supervisor Control And Data Acquisition (SCADA) systems. These systems are used to control and monitor physical industrial processes like electrical power supply, water supply and railroad transport. Since our society is heavily dependent on these industrial processes we are also dependent on the behavior of our SCADA systems. SCADA systems have become (and continue to be) integrated with other IT systems they are thereby becoming increasingly vulnerable to cyber threats. Decision makers need to assess the security that a SCADA system's architecture offers in order to make informed decisions concerning its appropriateness. However, data collection costs often restrict how much information that can be collected about the SCADA system's architecture and it is difficult for a decision maker to know how important different variables are or what their value mean for the SCADA system's security.

The contribution of this thesis is a modeling framework and a theory to support cyber security vulnerability assessments. It has a particular focus on SCADA systems. The thesis is a composite of six papers. Paper A describes a template stating how probabilistic relational models can be used to connect architecture models with cyber security theory. Papers B through E contribute with theory on operational security. More precisely, they contribute with theory on: discovery of software vulnerabilities (paper B), remote arbitrary code exploits (paper C), intrusion detection (paper D) and denial-of-service attacks (paper E). Paper F describes how the contribution of paper A is combined with the contributions of papers B through E and other operationalized cyber security theory. The result is a decision support tool called the Cyber Security Modeling Language (CySeMoL). This tool produces a vulnerability assessment for a system based on an architecture model of it.

Keywords: cyber security, security assessment, vulnerability assessment, architecture modeling, enterprise architecture.

Sammanfattning

Informationsteknik (IT) är kritiskt och värdefullt för vårt samhälle. En viktig typ av IT-system är de styrsystem som ofta kallas SCADA-system (från engelskans ”Supervisor Control And Data Acquisition”). Dessa system styr och övervakar fysiska industriella processer så som kraftförsörjning, vattenförsörjning och järnvägstransport. Eftersom vårt samhälle är beroende av dessa industriella processer så är vi också beroende av våra SCADA-systems beteende. SCADA-system har blivit (och fortsätter bli) integrerade med andra IT system och blir därmed mer sårbara för cyberhot. Beslutsfattare behöver utvärdera säkerheten som en systemarkitektur erbjuder för att kunna fatta informerade beslut rörande dess lämplighet. Men datainsamlingskostnader begränsar ofta hur mycket information som kan samlas in om ett SCADA-systems arkitektur och det är svårt för en beslutsfattare att veta hur viktiga olika variabler är eller vad deras värden betyder för SCADA-systemets säkerhet.

Bidraget i denna avhandling är ett modelleringsramverk och en teori för att stödja cybersäkerhetsutvärderingar. Det har ett särskilt focus på SCADA-system. Avhandlingen är av sammanläggningstyp och består av sex artiklar. Artikel A beskriver en mall för hur probabilistiska relationsmodeller kan användas för att koppla samman cybersäkerhetsteori med arkitekturmodeller. Artikel B till E bidrar med teori inom operationell säkerhet. Mer exakt, de bidrar med teori angående: upptäckt av mjukvarusårbarheter (artikel B), fjärrexekvering av godtycklig kod (artikel C), intrångsdetektering (artikel D) och attacker mot tillgänglighet (artikel E). Artikel F beskriver hur bidraget i artikel A kombineras med bidragen i artikel B till E och annan operationell cybersäkerhetsteori. Resultatet är ett beslutsstödsverktyg kallat Cyber Security Modeling Language (CySeMoL). Beslutsstödsverktyget producerar sårbarhetsutvärdering för ett system baserat på en arkitekturmodell av det.

Nyckelord: cybersäkerhet, säkerhetsvärdering, sårbarhetsvärdering, arkitekturmodellering.

Preface

When my research on this topic began in early 2007 the American cyber security regulation NERC CIP was a buzzword, and electrical power utilities in my surroundings began to become aware of the cyber security issues related to their SCADA systems. During my first years of working with cyber security of SCADA systems I often ended up in discussions concerning the relevance of the topic with those who owned the problem, i.e., asset owners and SCADA system suppliers. These discussions were on aspects such as: if there were a threat at all, why cyber attacks would be used instead of dynamite and what a cyber attack against a SCADA system possibly could accomplish. Now, at the finalization of this thesis, the computer worm (or "cyber weapon") Stuxnet still gets headlines in prominent magazines and papers, even though it was discovered more than two years ago. During the past two years, my discussions with problem-owners have focused on finding and describing solutions, and not on debating whether there is a problem worth considering. I sincerely hope that this thesis, along with the other outputs produced during my PhD studies (e.g., the tool supporting applications of these theories), will help to make our SCADA systems more secure.

As is customary in the Swedish system, this thesis is divided into two parts. The first part summarizes and gives an overview of the second part. In the second part the actual contributions are presented. The actual contributions are six of the papers produced during my doctoral studies. These six papers all contribute to the problem of assessing the cyber security of a system. The first paper presents a template which can be used to express security theory so that it can be directly applied on a system model. Papers two through five present theories on the topic and paper six presents a software tool that combines the formalism and the theory in order to support cyber security vulnerability assessments.

It is difficult to produce an exhaustive list of all those who have helped, contributed, and supported me during this journey. In addition to my colleagues at the department and paper co-

authors (especially Hannes Holm), I would like to thank associate professor Mathias Ekstedt, professor Pontus Johnson, and professor Lars Nordström for their guidance. I would also like to give a special thanks to Judith Westerlund and my wife Caroline for their support and encouragement. Finally, I would like to thank all the security experts who have contributed to my research projects.

Teodor Sommestad

List of included papers

Paper A: T. Sommestad, M. Ekstedt, and P. Johnson, "A probabilistic relational model for security risk analysis," *Computers & Security*, vol. 29, no. 6, pp. 659-679, 2010.

Paper B: T. Sommestad, H. Holm, and M. Ekstedt, "Effort estimates for vulnerability discovery projects," in *Proceedings of the 45th Hawaii International Conference on System Sciences*, pp. 5564-5573, 2012.

Paper C: T. Sommestad, H. Holm, and M. Ekstedt, "Estimates of success rates of remote arbitrary code execution attacks," *Information Management & Computer Security*, vol. 20, no. 2, pp. 107-122, 2012.

Paper D: T. Sommestad, H. Holm, M. Ekstedt, and N. Honeth, "Quantifying the effectiveness of intrusion detection systems in operation through domain experts," Submitted.

Paper E: T. Sommestad, H. Holm, and M. Ekstedt, "Estimates of success rates of Denial-of-Service attacks," in *Proceedings of IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 21-28, 2011.

Paper F: T. Sommestad, M. Ekstedt, and H. Holm, "The Cyber Security Modeling Language: A Tool for Vulnerability Assessments of Enterprise System Architectures," IEEE Systems Journal, Accepted for publication.

Contribution in the included papers: In all papers (A-F) Teodor Sommestad has been the leading researcher and primary author. In paper A Mathias Ekstedt and Pontus Johnson assisted with problem definition and authoring. In papers B-E Hannes Holm had an active role in the research and did approximately a third of the authoring. Mathias Ekstedt contributed to papers B-E in terms of authoring and advice on method selection and survey design. Nicholas Honeth contributed to the authoring and survey design in paper D. Mathias Ekstedt contributed to paper F in terms of the overall idea and authoring. Hannes Holm contributed to paper F with a case study.

Publications not included in the thesis

Publication I: P. Johnson, E. Johansson, T. Sommestad, and J. Ullberg, "A tool for enterprise architecture analysis," in *Proceedings of Enterprise Distributed Object Computing Conference*, 2007, pp. 142–142.

Publication II: M. Ekstedt, P. Johnson, M. Gammelgård, T. Sommestad, and P. Gustafsson, "Setting the Business Goals," in *Enterprise Architcture: models and analyses for information systems decision making*, Sweden: Studentlitteratur AB, 2007.

Publication III: P. Johnson, M. Ekstedt, R. Lagerström, and T. Sommestad, "Introduction," in *Enterprise Architcture: models and analyses for information systems decision making*, Sweden: Studentlitteratur AB, 2007.

Publication IV: U. Franke, T. Sommestad, M. Ekstedt, and P. Johnson, "Defense Graphs and Enterprise Architecture for Information Assurance Analysis," in *Proceedings of the 26th Army Science Conference*, 2008.

Publication V: T. Sommestad, M. Ekstedt, and P. Johnson, "Combining Defense Graphs and Enterprise Architecture Models for Security Analysis," in *Proceedings of 2008 12th International IEEE Enterprise Distributed Object Computing Conference*, 2008.

Publication VI: E. Johansson, T. Sommestad, and M. Ekstedt, "Security Isssues For SCADA Systems within Power Distribution," in *Proceedings of Nordic Distribution and Asset Management Conference (NORDAC)*, 2008.

Publication VII: Y. Xiaofeng, T. Sommestad, C. Fung, and P. C. K. Hung, "Emergency Response Framework for Aviation XML Services on MANET," in *Proceedings of The IEEE International Conference on Web Services (ICWS)*, 2008.

Publication VIII: U. Franke, J. Ullberg, T. Sommestad, R. Lagerström, and P. Johnson, "Decision support oriented

Enterprise Architecture metamodel management using classification trees," in *2009 13th Enterprise Distributed Object Computing Conference Workshops*, 2009.

Publication IX: E. Johansson, T. Sommestad, and M. Ekstedt, "Issues of Cyber Security In Scada-Systems-on the Importance of Awareness," in *Proceedings of the 20th International Conference on Electricity Distribution (CIRED)*, 2009.

Publication X: P. Närman, T. Sommestad, S. Sandgren, and M. Ekstedt, "A framework for assessing the cost of IT investments," in *PICMET 2009 Proceedings*, 2009.

Publication XI: T. Sommestad, M. Ekstedt, and L. Nordstrom, "Modeling Security of Power Communication Systems Using Defense Graphs and Influence Diagrams," *IEEE Transactions on Power Delivery*, vol. 24, no. 4, pp. 1801-1808, 2009.

Publication XII: S. Buckl, U. Franke, O. Holschke, F. Matthes, C.M. Schweda, T. Sommestad, and J. Ullberg, "A Pattern-based Approach to Quantitative Enterprise Architecture Analysis," in *Proceedings of 15th Americas Conference on Information Systems (AMCIS)*, 2009.

Publication XIII: M. Ekstedt, U. Franke, P. Johnson, R. Lagerström, T. Sommestad, J. Ullberg, and M. Buschle, "A Tool for Enterprise Architecture Analysis of Maintainability," in *Proceedings of the 2009 European Conference on Software Maintenance and Reengineering*, 2009.

Publication XIV: W. R. Flores, T. Sommestad, P. Johnson, and M. Simonsson, "Indicators predicting similarities in maturity between processes: An empirical Analysis with 35 European organizations," in *Proceedings of 1st Annual Pre-ICIS Workshop on Accounting Information Systems*, 2009.

Publication XV: T. Sommestad, M. Ekstedt, and P. Johnson, "Cyber Security Risks Assessment with Bayesian Defense Graphs and Architectural Models," in *Proceedings of Hawaii International Conference on System Sciences*, 2009.

Publication XVI: M. Ekstedt and T. Sommestad, "Enterprise Architecture Models for Cyber Security Analysis," in *Proceedings of IEEE PES Power Systems Conference & Exhibition (PSCE)*, 2009.

Publication XVII: M. Buschle, J. Ullberg, U. Franke, R. Lagerström, and T. Sommestad, "A Tool for Enterprise Architecture Analysis using the PRM formalism," in *CAiSE2010 Forum PostProceedings*, 2010.

Publication XVIII: M. Buschle, J. Ullberg, U. Franke, R. Lagerström, and T. Sommestad, "A Tool for Enterprise Architecture Analysis using the PRM formalism," in *Proceedings of CAiSE Forum 2010*, 2010.

Publication XIX: T. Sommestad, G. Björkman, M. Ekstedt, and L. Nordström, "Information system architectures in electrical distribution utilities," in *Proceedings of NORDAC*, 2010.

Publication XX: G. Björkman, T. Sommestad, M. Ekstedt, H. Hadeli, Z. Kun, and M. Chenine, *SCADA system architectures*. Stockholm, Sweden: Report of The VIKING project, 2010.

Publication XXI: F. Löf, J. Stomberg, T. Sommestad, M. Ekstedt, J. Hallberg, and J. Bengtsson, "An Approach to Network Security Assessment based on Probabilistic Relational Models," in *First Workshop on Secure Control Systems (SCS-1)*, 2010.

Publication XXII: T. Sommestad, M. Ekstedt, and L. Nordström, "A case study applying the Cyber Security Modeling Language," in *Proceeding of CIGRE (International Council on Large Electric Systems)*, 2010.

Publication XXIII: T. Sommestad, G. Ericsson, and J. Nordlander, "SCADA System Cyber Security – A Comparison of Standards," in *Proceedings of IEEE PES General Meeting*, 2010.

Publication XXIV: T. Sommestad and J. Lillieskold, "Development of an effort estimation model – a case study on delivery projects at a leading IT provider within the electric utility industry," *International Journal of Services Technology and Management*, vol. 13, no. 1/2, p. 152, 2010.

Publication XXV: H. Holm, T. Sommestad, J. Almroth, M. Persson, "A quantitative evaluation of vulnerability scanning", *Information Management & Computer Security*, vol. 19, no. 4, pp. 231-247, 2011.

Publication XXVI: T. Sommestad, *Exploiting network configuration mistakes: practitioners self-assessed success rate*. Stockholm, Sweden: TRITA-EE 2011:069, 2011.

Publication XXVII: T. Sommestad, H. Holm, and M. Ekstedt, *Threats and vulnerabilities, final report*. Stockholm, Sweden: Report of The VIKING project, 2011.

Publication XXVIII: H. Holm, T. Sommestad, and M. Ekstedt *Vulnerability assessment of SCADA systems*. Stockholm, Sweden: Report of The VIKING project, 2011

Publication XXIX: T. Sommestad, *Password authentication attacks: a survey of attacks and when they will succeed*. Stockholm, Sweden: TRITA-EE 2011:067, 2011.

Publication XXX: H. Holm, T. Sommestad, U. Franke, M. Ekstedt, "Expert Assessment on the Probability of Successful Remote Code Execution Attacks," in *Proceedings of WOSIS 2011 - Proceedings of the 8th International Workshop on Security in Information Systems, In conjunction with ICEIS 2011*, Beijing, China, 49-58, 2011

Publication XXXI: T. Sommestad and J. Hallberg, "Cyber security exercises as a platform for cyber security experiments," in *TAMSEC*, 2011, p. 33.

Publication XXXII: W. Flores, T. Sommestad, and H. Holm, "Assessing Future Value of Investments in Security-Related IT Governance Control Objectives – Surveying IT Professionals," *The Electronic Journal of Information Systems Evaluation*, vol. 14, no. 2, pp. 216-227, 2011.

Publication XXXIII: H. Holm, T. Sommestad, U. Franke, and M. Ekstedt, "Success rate of remote code execution attacks – expert assessments and observations," *Journal of Universal Computer Science*, vol. 18, no. 6, pp. 732-749, 2012.

Publication XXXIV: T. Sommestad and A. Hunstad, "Intrusion detection and the role of the system administrator," in *Proceedings of International Symposium on Human Aspects of Information Security & Assurance*, 2012.

Publication XXXV: T. Sommestad and J. Hallberg, "Cyber security exercises and competitions as a platform for cyber security experiments," in *Proceedings of the 17th Nordic Conference on Secure IT Systems*, 2012.

Table of contents

Part one: Summary

1 Introduction

This introduction describes the thesis' outline, the background of the research and the objectives of the research.

1.1 Outline of the thesis

This thesis is divided into two parts. Part one (this part) presents an overview and part two presents a summary to the actual contribution.

The remainder of section 1 in this part of the thesis gives a description of the background and objectives of the research. Section 2 describes related works and relates this to the contribution of this thesis. Section 3 summarizes the contribution of this thesis by presenting properties of the theory presented in it. Section 4 describes the research design.

The second part of the thesis contains six papers labeled papers A through F. Two of these papers have been published in the proceedings of international conferences, three have been accepted or published in international journals, and one is currently under review for publication at an international journal. The papers contain the same content as when they were published/accepted/submitted, only their typesetting has been changed.

1.2 Background

Information technology (IT) is critical and valuable to our society. IT systems support business processes by storing, processing, and communicating critical and sensitive business data. In addition, IT systems are often used to control and monitor physical industrial processes. For example, our electrical power supply, water supply and railroads are controlled by IT systems. These "controlling" systems have many names. In this thesis they are referred to as SCADA (Supervisory Control And Data Acquisition) systems, or occasionally, as industrial control systems. They are complex real-time systems that include components like databases, application servers, web interfaces, human machine interfaces, dedicated communication equipment, process control logic, and numerous sensors and actuators that measure and control the state of the industrial process. In many industrial processes (e.g., electrical power transmission) these components are also distributed over a large geographical area. SCADA systems can be seen as the nervous system of industrial processes [1] and since our society is heavily dependent on the industrial processes that SCADA systems manage, we are also dependent on the behavior of our SCADA systems.

Over the last two decades our SCADA systems and their environments have changed. They used to be built on proprietary and specialized protocols and platforms [2]. Today, however, SCADA systems operate on top of common and widely used operating systems (e.g., Windows XP) and use protocols that are standardized and publicly available (e.g., IEC 60870-5-104). These changes have altered the threat environment for SCADA systems.

The move to more well-known and open solutions lowers the threshold for attackers who seek to exploit vulnerabilities in these SCADA systems. Vulnerabilities are regularly found in the software components used in SCADA systems (e.g., the operating systems) and instructions that can be used to exploit these vulnerabilities are often made available in the public domain. The increased openness also lowers the thresholds for attacks targeting special-purpose SCADA components, e.g., programmable logic controllers (PLCs). Today there is an interest in the vulnerabilities they have and there is information available in the public domain about their design and internal components. In fact, it is even possible to buy a subscription to exploit code specifically targeting SCADA systems' components (see for example [3]). In other words, a successful cyber attack against a SCADA system today does not require the SCADA-expertise that was required prior to the move to more open, standardized and common components.

In parallel with the move to more common and widely known solutions, SCADA systems have moved from being isolated and standalone to be interwoven in the larger IT environment of enterprises. Process data collected by SCADA systems, production plans, and facility drawings are often exchanged over enterprises' computer networks [4]. It is also common to allow users to remotely connect to operator interfaces, for instance, so that process-operators can connect remotely when they are on standby duty and so that suppliers are able to perform maintenance remotely [4].

The increased integration with more administrative enterprise systems has also contributed to a changed threat environment. Administrative systems are, with few exceptions, connected (directly or indirectly) to the internet. Hence, the possibility for administrative systems to exchange data with SCADA systems is also a possibility for attackers or malware to come in contact with these systems and exploit their vulnerabilities, without physical proximity.

The lowered threshold to find and use SCADA-related vulnerabilities and tighter integration with enterprise systems are two cyber security problems that add to the volume of cyber security issues related to architecture and configuration of the actual SCADA systems [5–7]. Historically, SCADA systems were built to be reliable and available, but not to be secure against attacks with a malicious intent.

SCADA systems are thus critical assets, have exploitable vulnerabilities, and are interwoven into the enterprise architectures. Decision makers who wish to manage their cyber security need to be able to assess the vulnerabilities associated with different solution architectures. However, assessing the cyber security of an enterprise environment is difficult. The budget allocated for cyber security assessments is usually limited. This prohibits assessments from covering and investigating all factors that could be of importance. The set of variables that should be investigated, and how important they are, is also hazy and partly unknown. For instance, guidelines such as [8–11] do not prioritize their cyber security recommendations. Such prioritizations are also difficult to do in a generic guideline since the importance of many variables are contingent on the systems architecture and environment and guidelines are limited to one or few typical architectures. Variables are also dependent on each other. An attack against a SCADA system may be performed in a number of ways and can involve a series of steps where different vulnerabilities are exploited. Thus, some combinations of vulnerabilities can make an attack easy, but a slightly different combination may make attacks extremely difficult. Thus, informed decisions require an analysis of the vulnerabilities associated with different architectural scenarios, and at the same time, an analysis of how these vulnerabilities relate to each other.

These problems are not unique for SCADA systems. Many administrative IT systems also have complex environments; administrative IT systems often need to be analyzed on a high level of abstraction; the importance of different variables is hazy also for administrative IT systems. Like the administrative environment, the SCADA environment consists of software, hardware, humans, and management processes. And as described above, there is a substantial overlap between the components which are used in both environments today. However, there is a difference in what needs to be protected in these environments. Security is often thought of as a triage of confidentiality, integrity and availability. For SCADA systems, integrity and availability of functionality are crucial, but confidentiality of business data is not [9]. Because of this, cyber security assessments of SCADA systems have a different focus than for many other systems. The importance of availability and integrity has also other implications. For instance, because of the consequence of a potential malfunction, it is recommended that SCADA systems should not be updated before extensive testing, and network based vulnerability scanners should be used with care in SCADA environments [9].

1.3 Objectives

The overall aim of this research is to develop support for those conducting cyber security assessments. More precisely, the objective is to: *Develop a tool that makes cyber security theory easy to use for decision makers.* To reach this objective the two sub-objectives were identified:

(1) *Define a formalism that makes it possible to apply a cyber security theory on system architecture specifications* and
(2) *Compile and develop cyber security theory that is relevant for decision makers in the SCADA domain.*

The purpose of this research is thus to help decision makers to assess the cyber security of IT systems with different architectures. Help is needed to assesses both existing systems "as-is" and potential future "to-be" systems. Focus is on supporting decision makers in the SCADA domain. As presented above (cf. section 1.2) such support must tackle practical issues. First, cyber security assessment cannot be overly costly to perform, viz. all details concerning the SCADA system's architecture and configuration cannot be investigated. Second, the theory on what makes a system secure is, is not always clear (especially when details about the system are missing) and in approximations are necessary. Both these practical issues make assessments uncertain and to support a decision maker, trade-offs are needed with respect to accuracy. The aim is to produce a reasonable tradeoff between accuracy and the cost of collecting system specific data while communicating the uncertainty of the result.

2 Related works

The contribution of this thesis follows ideas of the management approach called enterprise architecture. Enterprise architecture is an approach for holistic management of information systems where diagrammatic descriptions of systems and their environment are central. A number of established enterprise architecture frameworks exist, including: The Open Group Architecture Framework [12], the Ministry of Defence Architecture Framework [13] and the Department of Defense Architecture Framework [14]. The research presented in this thesis follows the ideas presented in [15], [16] concerning enterprise architecture modeling and decision making. The overall idea is that the concepts represented in (enterprise) architecture models should be there because they, according to theory, are needed to answer questions of interest to the decision maker that uses the architecture for some specific purpose.

This thesis focuses on questions related to cyber security and how to answer those questions with the support of architectural models of systems. While established some enterprise architecture frameworks do address security explicitly, the analysis support they offer is sparse. For instance, in the process suggested by The Open Group Architecture Framework [12] includes steps where one should "Identify potential/likely avenues of attack" and "Determine what can go wrong?", however, it is up to the user of the method (the architect) to do so. Similarly, the support offered by the Ministry of Defence Architecture Framework is to document the result of a security assessment, not to support the analysis required to do it. As described in [17]: "the aim of this guidance for representing security considerations is to enable sufficient information to be recorded for interested parties".

The thesis describes a framework for connecting system architecture models to cyber security assessment (paper A), theory to aid such assessments (papers B-E) and the combination of these into a model that can be described as an expert system (paper F). The three sections below are intended to provide an overview of related work in the directions of the included papers. More elaborate descriptions can be found in the corresponding papers.

Section 2.1 describes methods and models for cyber security assessments. These methods and models require operationalized cyber security theory or system-specific cyber security data (e.g., mean-time to compromise data) to be able to operate. Work on operationalized theory is described in section 2.2. Section 2.3 describes methods and tools that use operationalized cyber security theory to help decision makers assess cyber security.

2.1 Metrication frameworks and methods

A number of ideas can be found on how cyber security should be assessed. Some ideas concern how security measurements should be defined and operationalized. Examples include the ISO/IEC standard 27000-4 [18] and NIST's security metric guide [19]. These publications describe how an organization should develop and maintain a measurement program, but do not define the actual measurements that should be made or what different measurement values mean in terms of security. In addition to these there are general qualitative models that describe variables (or concepts) in the security domain and how these concepts relate to each other. CORAS contains a metamodel over the security field to support assessments made using the CORAS method [20], Common Criteria has a conceptual model over variables (or concepts) a security assessment needs to consider [21] and several similar qualitative models are available. For instance, [22–29] are generic alternatives and [30], [31] are alternatives with a particular focus on SCADA systems that control energy systems. These methods, security metamodels, conceptual models and technical reference models can support cyber security assessments and be used to define operational cyber security metrics. However, they require a substantial mental effort from their user – the user must identify what to measure and how important this is for the IT system's cyber security.

To ease this burden, articles published in scientific forums on security measurement often describe methods to combine security-variables into one metric. Broadly speaking, they define which cyber security variables that should be operationalized and how they should be combined. Examples include: attack trees [32], threat trees[33], defense trees [34], attack and protection trees [35], Boolean Logic Driven Markov Processes [36], the CORAS method [20], XMASS [37], ISRAM [38], NIST's risk assessment framework [39], the economic framework given in [40] and Secure Tropos [41]. Some metrication methods have also been proposed specifically for SCADA systems (e.g., [42–44]).

These metrication methods describe how their variables should be combined to produce a meaningful result. They can thus help to combine cyber security values of single systems to a value for a system-of-systems (e.g., the expected monetary loss next year due to attacks). However, they all require that cyber security theory is supplied by the user. In some cases both qualitative and quantitative theory is needed. For instance, the actual trees together with their attack success probabilities are needed for defense trees [34] and the attacker's process model together with time-to-compromise data is required for Boolean Logic Driven Markov Processes [36]. In some metrication methods the qualitative theory is complete and the user is only required to supply the system architecture and quantitative theory. One example is the model of Breu et al. [45] which requires threat realization probabilities, but describes which threat realization probabilities that are needed and how they should be combined for the modeled enterprise system. Another example is XMASS [37], which among other things requires that the modeler can acquire or specify "security profiles" for entities. With these security profiles a user can calculate an ordinal "security value" (between 0 and 100) for the components in the system.

Paper A describes a framework that can be used to tie security theory to architecture metamodels. Just as the model of Breu et al. [45] and XMASS [37] it can be used to infer the security properties that needs to be quantified from the system architecture. Like XMASS the framework described in paper A makes it possible to store security theory so that security can be assessed without employing security expertise to quantify security properties. Unlike XMASS the framework in paper A stores theory expressed in with concepts directly corresponding to states and events in the real world (e.g., attacks' success given use of certain countermeasures), and the framework produces output that are expressed in tangible units (e.g., expected monetary losses).

2.2 Operationalized cyber security theory

The metrication methods described in section 2.1 needs to be complemented with quantitative cyber security theory to be of practical use. This theory can be supplied together with the metrication method or supplied by the user of the method. The accuracy of the result when the method is applied will of course be contingent on the accuracy of the theory with which it is used. Many prominent research results have been produced on operational cyber security. Some are also specifically addressing the cyber security of SCADA systems (e.g., the demonstrations, assessments and tests described in [46–50]). Unfortunately only a small portion of these could be used in analyses of the types dealt with in this thesis. This section aims at giving an overview of available theory that has been used as a basis for this research and to point to gaps which are filled by papers B-E. More elaborate descriptions of studies related to the contributions in papers B-E can be found in the papers included in part two of this thesis.

Some areas of cyber security have an intrinsic quantitative element which makes metrication and estimation of the required effort to accomplish an attack straightforward [51]. In particular, established methods are available for assessing the strength of cryptographic methods and authentication methods (e.g., password authentication) under well specified conditions [51]. In other fields, empirical investigations have approximated the probability that the attacker would succeed with different attacks on the level of abstraction manageable in an enterprise security assessment (considering the cost of collecting data). For example, studies on social engineering attacks have produced success frequencies under different conditions [52–55]. Other studies have assessed the frequency of configuration mistakes in enterprises' systems and how difficult such mistakes are to exploit [56], [57]. Results described in these papers make up a subset of the theory used in the model of paper F.

With respect to software vulnerabilities there is empirical data available concerning public disclosed software vulnerabilities in databases like [58], [59]. In these, and in databases like [60], it is also possible to identify the vulnerabilities for which exploit code is publicly available. Models have been developed to predict how many cyber security vulnerabilities that will be publicly disclosed for a product [61–64]. For instance, the number of vulnerabilities found in a software product has been found to correlate to the number of user-months the product has accumulated and the time it has been on the market [62]. The effectiveness of different procedures for deploying security patches has also been assessed [65]. When it comes to development of new exploits it is reasonable to assume that this is a straightforward task for a professional penetration tester when patch information is available for the vulnerability. For instance, it is demonstrated in [66] that exploit development can be automated for selected classes of vulnerabilities under those circumstances. However, to predict how difficult it would be for an attacker to find a zero-day vulnerability (i.e., a vulnerability discovered by someone, but which is still unknown to the public and the system owner) in a software product and develop an exploit for it is more difficult. In [67] it is estimated how many zero day vulnerabilities there have been at different points in time during recent years. However, since data on the effort invested in the discovery projects identifying these vulnerabilities (or those projects that failed to identify a software vulnerability) is unavailable [61], it is difficult to deduce the required effort for finding a new vulnerability from the archival records available. Paper B contributes to this with effort estimates for discovery projects undertaken given different conditions.

Several studies have investigated the exploitation of software vulnerabilities, in particular the type of exploitation where a remote attacker obtains control of the vulnerable system. In [68–82] attacks and defenses are described. While these publications describe countermeasures and attacks they mitigate, no study has been found that states how common different conditions and attack forms are, i.e., how often an intelligent attacker will or can employ each of the attack forms studied. Because of this, these studies could not be applied directly to this work. Paper C contributes to this with success rates under different conditions.

Intrusion detection systems monitor systems and aim at identifying attacks made against them. A number of empirical studies have been performed on the probability of attacks being detected and false alarms being produced by these systems (e.g. [83], [84]) and on the impact of different parameters' impact (e.g. [85–87]). However, testing intrusion detection systems in a way that makes the result generalizable to real systems is difficult [88–91]. Studies on intrusion detection systems are also technical and focus on the property of the system alone. In practice, however, it is a tool used by an administrator who monitors its output [92–95] and judges if the alarms are worth reacting upon. A first attempt to assess detection rates when administrators are monitoring the output of the intrusion detection system is described in [96]. While the result of [96] clearly shows the importance of considering system administrators, it is too narrow to offer generic data on intrusion detection systems' effectiveness. Paper D contributes to this with broad and general estimates on how an administrator using an intrusion detection system will perform given different conditions.

Work has also been performed on the denial of service attacks. Examples of experiments, observations and simulations on denial of service attacks and related countermeasures can be found in [97–103]. However, since these studies are made under different assumptions it is difficult to generalize from their results and translate them into a real-world context. Broader reviews in the denial-of-service field [104–108] are also of a qualitative nature. Paper E makes a quantitative contribution in this field and describes approximate success rates under various conditions.

2.3 Operationalized cyber security assessment methods

A number of research efforts prior to the one presented in this thesis operationalize a security assessment method so that decision makers only need to describe their systems in order to obtain the assessment of their enterprise architecture. In other words, there are other assessment methods where the user only needs to input information about the system architecture (and not operationalized security theory). Instead of requiring theory from the user, these assessment methods assign values for security properties (such as time-to-compromise or attack success probability) for the system architecture based on a generic theory.

Research efforts along these lines have in recent years focused on methods that use attack graphs. These methods aim at resolving which attacks can be made against a system architecture. Since potential attacks are the source of cyber security risk, these methods match decision making processes concerning cyber security. The approach were threats and attacks are modeled could be compared to methods that check compliance to a set of standardized security requirements for SCADA systems (e.g., [109], [110]) instead of indicating the vulnerabilities that different solutions have.

Methods based on attack graphs are based on a model over the system architecture and a database of exploits or security vulnerabilities [111], [112]. With this data, an algorithm calculates privileges and network states that can be reached by an attacker starting from a certain position [111]. Since the early variants of attack graphs (like [113], [114]) several tools have been developed with different solutions to the problem. Differences can be seen both in terms of the data they require as input and the output they produce when they are applied. The most mature tools described in the literature are: NetSPA [115], [116], the TVA-tool [117–119] and MulVAL [120].

The operationalized security assessment method presented in this thesis is called CySeMoL (Cyber Security Modeling Language) and is described in paper F. Its conceptual model is similar to that of attack graphs, and like attack graphs it instantiates ways that an attacker can compromise the modeled system. The abstraction level of CySeMoL's analysis is higher than the abstraction level used in attack graph methods like NetSPA, TVA-tool and MulVAL. In particular, CySeMoL does not model individual instance of software vulnerabilities or individual exploits. On the other hand CySeMoL includes more types of entities in the analysis. For example, CySeMoL includes human users and management processes in the analysis.

CySeMoL proposes solutions to some issues with implemented attack graph methods. In particular:

- Unlike NetSPA, CySeMoL does not assume that all vulnerabilities are exploitable on all machines, regardless of configuration.
- Unlike MulVAL, CySeMoL gives arguments for the validity of quantitative data on how difficult it is to exploit a vulnerability.
- Unlike MulVAL and NetSPA, CySeMoL does not rely on the output of vulnerability scanners (which miss many vulnerabilities [121]) to be practically usable.
- Unlike TVA tool, CySeMoL does not require that the user of the model enters exactly which exploits the attacker can use.
- Unlike MulVAL and TVAtool, CySeMoL can assess attacks against client software.
- Unlike these three tools, CySeMoL covers more attack types than exploitation of software vulnerabilities.

The relationship to other operationalized security assessments methods are also described in paper F.

3 Result and contribution

The primary result of this research is a probabilistic relational model containing cyber security theory. This probabilistic relational model and the theory contained in it are henceforth referred to collectively as CySeMoL (Cyber Security Modeling Language). CySeMoL describes how attack steps and countermeasures relate to each other and how they can be used to assess the cyber security of an IT system architecture.

To use CySeMoL, the user supplies an object model complying with CySeMoL's metamodel, states the initial privilege of the attacker and states which attack step the attacker will try to reach (i.e., where the attack will end). With this input CySeMoL can suggest paths the attacker would take and estimate the probability of the attacker succeeding, given that he/she has tried. CySeMoL is thus a theory developed to support cyber security vulnerability assessment. Below, CySeMoL is described using the seven structural components of theories outlined in [122]:

- means of representation
- constructs
- statements of relationships
- scope
- causal explanations
- testable propositions
- prescriptive statements

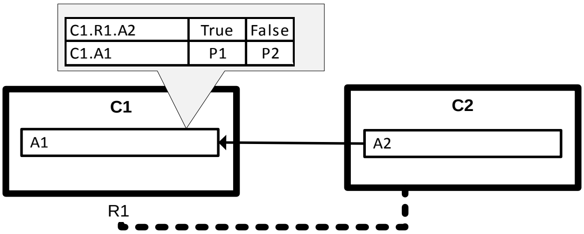
Each of these theory components is described in a separate subsection below.

3.1 Means of representation

A theory needs to be represented physically in some way [122]. The theory in this thesis is represented through a probabilistic relational model. More specifically, it is represented through a probabilistic relational model complying with the template described in paper A.

A probabilistic relational model (PRM) [123] specifies how a Bayesian network [124] should be constructed from an object model (instance model). In other words, it states how a Bayesian network should be created from a model that instantiates a class diagram (metamodel), such as the one of UML (Unified Modeling Language) [125]. A Bayesian network (sometimes called "causal network" [124]) is a graphical representation of probabilistic dependencies between variables [126]. Hence, a PRM can codify how probabilistic dependencies between objects are contingent on the objects' relationships to each other. As succinctly expressed in [123], PRMs ”are to Bayesian networks as relational logic is to propositional logic”.

In a PRM the classes can have attributes and reference slots. The attributes are random variables with discrete states; the reference slots point to other classes to state which relationships the class has with other classes. Attributes in the PRM are associated with a set of parents. The parents of an attribute *A* are attributes in the object model which *A*'s value depends upon. The association to an attribute's parents can be used to express qualitative theory. For instance, in Figure 1, attribute *A1* of class *C1* depends on attribute *A2* of class *C2* if objects of these classes are related to each other with reference slot *R1*. How an attribute depends on its parents is defined using a conditional probability table. The probabilities *P1* and *P2* in table of Figure 1 state how attribute *C1.A1* (attribute *A1* for objects of class *C1*) is determined by the value of *C1.R1.A2* (attribute *A2* of the object that *R1* points to). Thus, the theory embedded in PRM is quantified through conditional probabilities.



C1.R1.A2	True	False
C1.A1	P1	P2

Figure 1. The PRM formalism.

CySeMoL's theory is expressed according to the template depicted in Figure 2. This template is a PRM with abstract classes (i.e., classes that needs to be further refined to be possible to be instantiate in an architecture model). It describes abstract classes that are of relevance to cyber security assessments and describe how the attributes of these classes depend on each other. Among other things, it contains five subclasses to the class *Countermeasure* and details how these influence the cyber security risk. For example, a *PreventiveCountermeasure* influences the probability that an *AttackStep* can be accomplished, while a *ContingencyCountermeasure* influences the loss that would be inflicted on an *Asset* if a *Threat* would be realized.

To summarize, both the qualitative and quantitative parts of the theory are represented through a PRM. An advantage of this means of representation is the possibility of automatically applying the theory on a modeled architecture. A PRM constitutes a formal description for how the value of objects' attributes should be calculated in an object model. Given that a system's architecture is described as an object model, the value of its attributes can be inferred automatically from the theory of the PRM. Such inference can also infer values for attributes which have not been observed, i.e., attributes that do not have a state assigned.

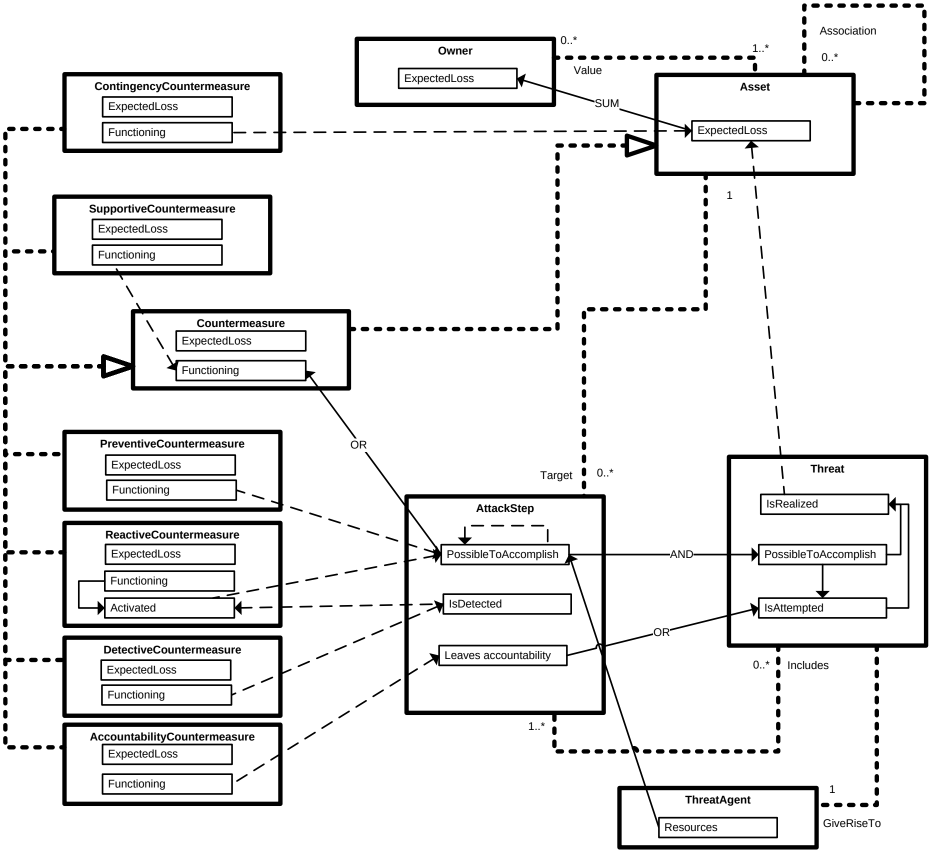


Figure 2. The PRM template used as a framework.

3.2 Constructs

A number of constructs are used in CySeMoL. These constructs are specializations of those in the abstract PRM template (cf. Figure 2). The theory is limited to vulnerability assessments and does not concretize all construct-types in the template. The classes *Asset*, *AttackStep*, *PreventiveCountermeasure*, *DetectivCountermeasure* and *ReactiveCountermeasure* are concretized. One type of *ThreatAgent* is considered, and the *Threat*-class is used but not further concretized.

The theory within CySeMoL is focused on issues concerning SCADA systems. As mentioned in section 1.2, integrity and availability of these systems is the primary concern and confidentiality is not. Also, SCADA systems operate in an environment where certain elements are commonly present and others are not. For instance, bank transactions and mobile phones are not relevant to the typical SCADA system's cyber security. Both the concerns of decision makers and the elements present in SCADA systems' environments have influenced which constructs have been included in CySeMoL.

The metamodel of CySeMoL depicts the constructs of the theory and their relationships to each other. Figure 3 depicts the constructs in terms of classes, attributes and class-relationships (reference slots). Note that this figure is on another level of abstraction than Figure 2, and most attributes in this figure correspond to classes in Figure 2. For example, the attribute *FindHighSeverityVulnerability* in the class *SoftwareInstallation* in Figure 3 is a special type of *AttackStep* (depicted as a class in Figure 2). This is similar to the metamodel layering of UML and the relationship between UML and MOF (Meta Object Facility) [125].

The constructs in CySeMoL have descriptive names. They also have a more elaborate textual definition. For instance, paper C defines and describes a number of the attributes related to arbitrary code exploits. The definitions are intended to be intuitive and accepted in the community. For example, the Common Vulnerability Scoring System's definitions [127] are used in paper C to define properties of attacks and vulnerabilities.

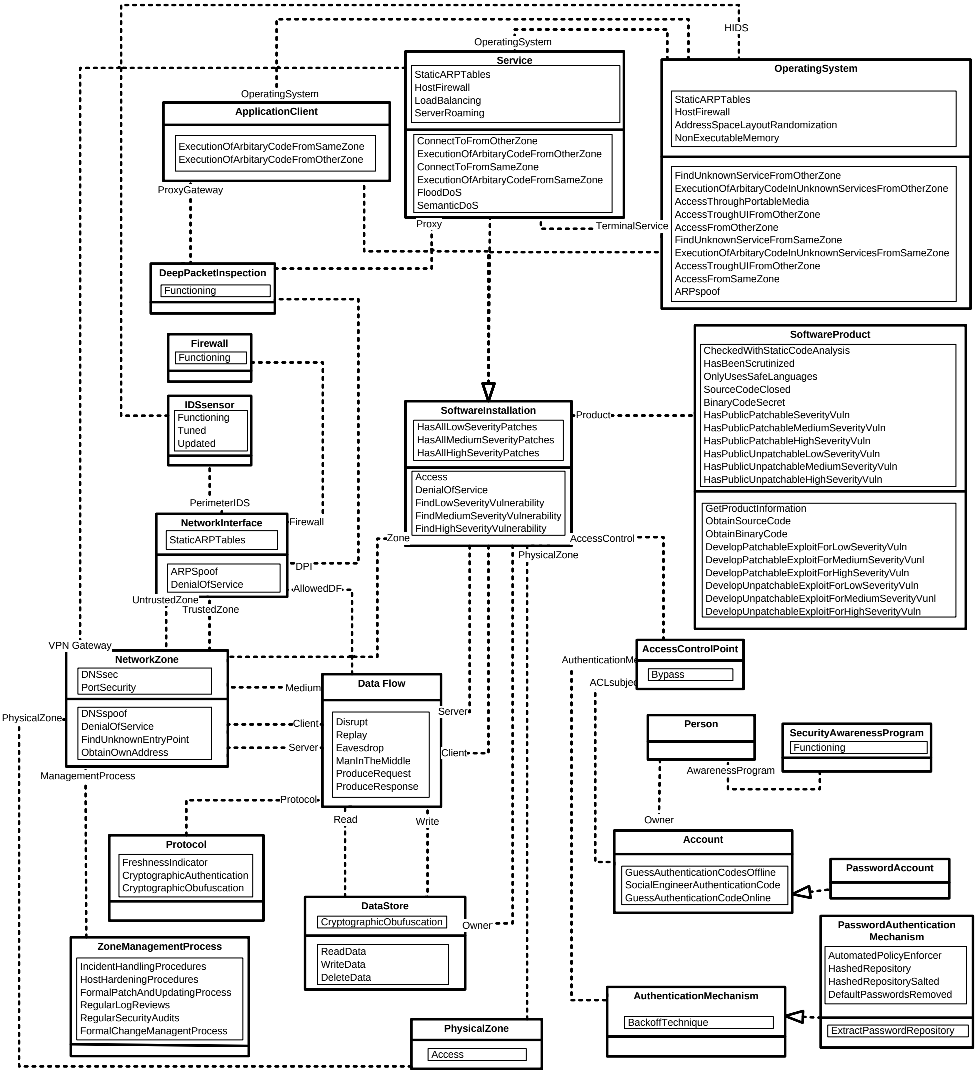


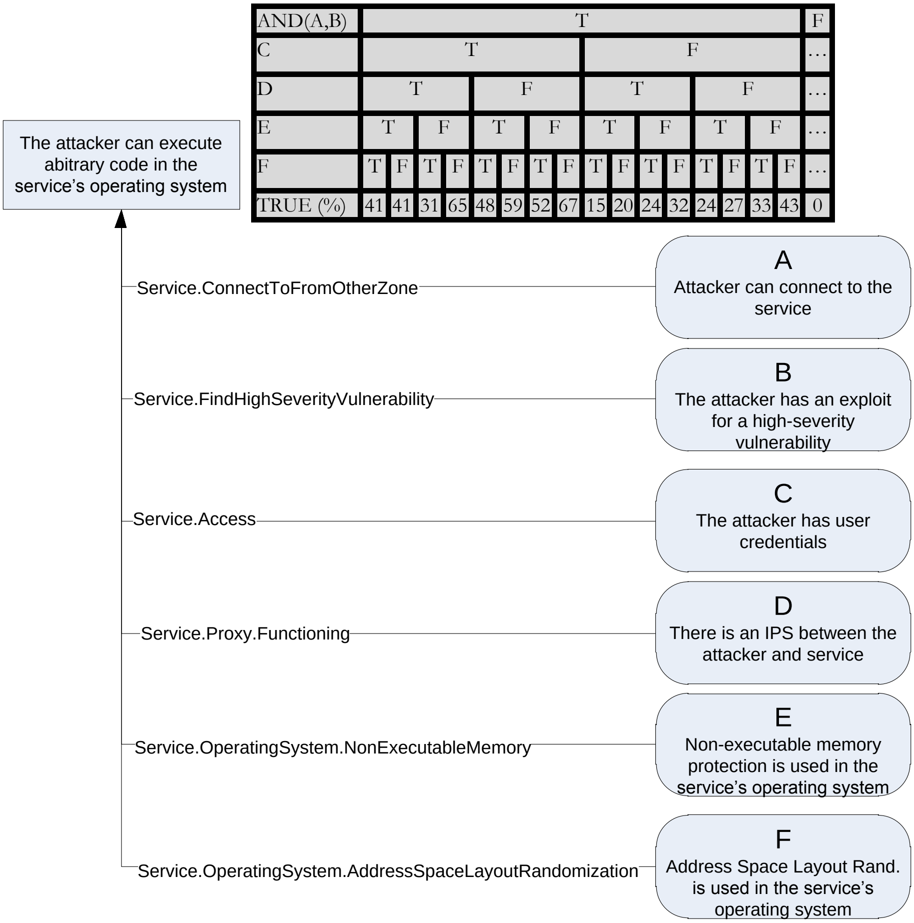
Figure 3. The metamodel of CySeMoL. Countermeasures associated with a class are listed in the class' upper plate. Attack steps associated with a class are listed in the class' lower plate.

3.3 Statements of relationship

CySeMoL describes a large number of relationships. Relationships between classes are expressed as reference slots; relationships between attributes are expressed through slot chains and conditional probability tables. Both types of relationship are directional. The class-relationships (reference slots) are deterministic while many of the attribute-relationships are probabilistic and uncertain.

The attribute-relationships are quantified through conditional probability tables. Just as the constructs are a subset of the constructs in the abstract PRM template, attribute-relationships are a subset of the attribute relationships in the abstract PRM template. This subset is limited to attribute-relationships between subclasses to: *PreventiveCountermeasure* and *AttackStep*, *DetectiveCountermeasure* and *AttackStep*, *ReactiveCountermeasures* and *AttackStep*, *AttackStep* and *AttackStep*. The derived relationships stated in CySeMoL are too many to be described here. Refer to papers B through F for details. An example drawn from paper C is presented in Figure 4. In this example, the influence of six variables is expressed in the conditional probability table. The dependent variable and variables A-C are subclasses to *AttackStep*; variables D-E are subclasses to *PreventiveCountermeasure*. If both parent A and parent B are true, a probabilistic dependency exists. However, if either one of parents A or B is false, the response variable will be false regardless of the state of other variables.

Of all entries in CySeMoL's conditional probability tables, 82 percent are deterministic. In other words, the value is either one or zero under 82 percent of the conditions. Deterministic relationships exist when some set of conditions are required for an attack to be feasible at all (as in the example in Figure 4), or when a variable is used as an aggregate for some other variable to simplify the PRM. The remaining 18 percent of the entries in the conditional probability tables are probabilistic values reflecting uncertainty about the variables state in this scenario. When CySeMoL's theory is applied, it is important to consider this uncertainty. The theory of CySeMoL is specified on a high level of abstraction, and the theory will in many cases only offer a rough approximation.



AND(A,B)	T																F
C	T								F								…
D	T				F				T				F				…
E	T		F		T		F		T		F		T		F		…
F	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F	…
TRUE (%)	41	41	31	65	48	59	52	67	15	20	24	32	24	27	33	43	0

The attacker can execute abitrary code in the service's operating system
Service.ConnectToFromOtherZone
Service.FindHighSeverityVulnerability
Service.Access
Service.Proxy.Functioning
Service.OperatingSystem.NonExecutableMemory
Service.OperatingSystem.AddressSpaceLayoutRandomization
A
Attacker can connect to the service
B
The attacker has an exploit for a high-severity vulnerability
C
The attacker has user credentials
D
There is an IPS between the attacker and service
E
Non-executable memory protection is used in the service's operating system
F
Address Space Layout Rand. is used in the service's operating system

Figure 4. Examples of relationships stated in CySeMoL.

3.4 Scope

As described in section 3.2, CySeMoL focuses on constructs and relationships that concern the cyber security of SCADA system. This focus influences the relationships that have been included in CySeMoL. However, the relationships that have been included in CySeMoL are equally valid for other domains than SCADA. For instance, the relationships depicted in Figure 4 are general and could be applied to any type of IT system. The studies used to define constructs and relationships have not been limited to the SCADA domain. The theory comes from generic security literature and the judgment of security experts from a broad population. The theory is thus possible to generalize to domains other than SCADA systems.

However, CySeMoL's theory is only valid for a specific threat model. The relationships have been expressed for the case when the threat agent is a professional penetration tester with access to publicly available tools and one week to spend on the attack. Clearly, other threats are also present. For instance, a threat agent can be the unskilled "script kiddie", a well-known computer worm or a group of skilled actors such as a military cyber command. The threat agent may also have access to different toolsets and a different amount of time to spend on the attack. CySeMoL's theory only covers cases concerning the professional penetration tester with publicly available tools and one week to spend.

In addition to delimitations regarding the threat agent the validity of the theory is contingent on developments in the threat environment and the cyber security measures employed in enterprises. Cyber security can be seen as an arms race, where attackers and defenders continuously improve and change their practices [128]. Advances on the attacking side will mean that certain attacks become easier to perform while advances on the defending side will mean that they are more difficult to perform. The theory presented in this thesis marginalizes a considerable number of variables with the assumption that they have the value they typically have in enterprises today. When advances are made on the adversarial side with respect to knowledge, skill, or tools, the estimates will underestimate the capability of attackers on the attack steps in questions. The estimates are also contingent on the assumption that marginalized variables related to enterprises' cyber security practices are as they are today. So, if the average values of architecture-related variables outside the scope of the metamodel change significantly, then the estimates will become less accurate. While this means that the utility of the theory will deteriorate over time, maintaining it should possible if there is a will to do so. For instance, if publicly available tools include techniques to efficiently bypass the operating system protection called address space layout randomization, the validity of relationships where this variable is involved needs to be revised. Similarly, if there is a general increase in the security of software producer's products using means other than those included in this theory, other relationships will need to be revised.

3.5 Causal explanations

The theory in CySeMoL is rich in causal relationships and explanations. All the relationships stated in CySeMoL are drawn from hypotheses concerning causality that are described in the literature. In CySeMoL these are quantified and formally represented. As described in section 3.3, some relationships are probabilistic and some are deterministic. The table in Figure 4 gives examples of both. Textual explanations that further explain the causality are also available. For instance, explanations for the relationships in Figure 4 can be found in paper C. Paper C (like the other papers) also contains references to even more elaborate explanations for why they have a causal influence.

3.6 Testable propositions

An important quality of scientific theory is that it is testable. The propositions concern the capability of a professional penetration tester with one week to spend on this task. This threat is believed to be relevant for decision makers, known well-enough to make theory-construction possible, and possible to test formally to an acceptable extent. However, engaging professional penetration testers in weekly undertakings comes at a cost; formal empirical tests of the propositions put forward in CySeMoL in most cases have a considerable cost associated with them. In fact, the costs and practical obstacles associated with observational studies are the reason why domain experts are used to quantify much of the theory.

Performing experimental tests involving sampled professional penetration testers who spend one week each on an attack is certainly costly. Archival data on attack attempts from the threat agents of the type in question would be an option. However, reliable data of this type is not available today. As a consequence, encompassing tests on all parts of the proposed theory is likely to be costly. However, at a reasonable cost, tests can be performed on selected parts of the theory to test these parts' validity, and tests can be performed on a high level of abstraction on the theory as a whole.

On a low level of abstraction CySeMoL proposes conditional probabilities for specific attack steps (see Figure 4 for an example). A full-fledged experimental setup on this level of abstraction would require a sample of systems where attributes included in CySeMoL correspond to the prediction to be tested, and the attributes not included in CySeMoL are distributed in a way that is representative to those systems used in enterprises today. It also requires a representative sample of penetration testers who are willing to spend a week attacking each system according to a predefined path. Observations can then be made on success-frequencies for all entries in a conditional probability table to assess their calibration. A less resource-demanding approach would be to investigate a few strategically selected table-entries (probabilities) which CySeMoL predicts. Since the conditional probabilities in a table often originate from the same source (e.g., a group of security experts), a test on one entry also indicates the calibration of other entries. Tests arranged with less resourceful threat agents can also falsify the theory. For instance, if less resourceful or less skilled threat agents consistently perform better than CySeMoL predicts this suggests that CySeMoL underestimates the success probability.

On a high level of abstraction, CySeMoL proposes attack paths that have an approximated probability of success. An example is shown in Figure 5. Also on this level of abstraction a full-fledged experimental setup would require representative attackers and sampled system configurations that are representative for an enterprise environment. Like the tests on specific probability values, it also requires a representative sample of penetration testers who are willing to attack each system according to a predefined path. However, tests can be performed on strategically selected attack paths, or with less resourceful and/or competent threat agents. For instance, if threat agents consistently fail attack paths that CySeMoL predicts as easy but succeed with attack paths CySeMoL assigns a marginal success-probability, this would point to validity issues with CySeMoL's theory.

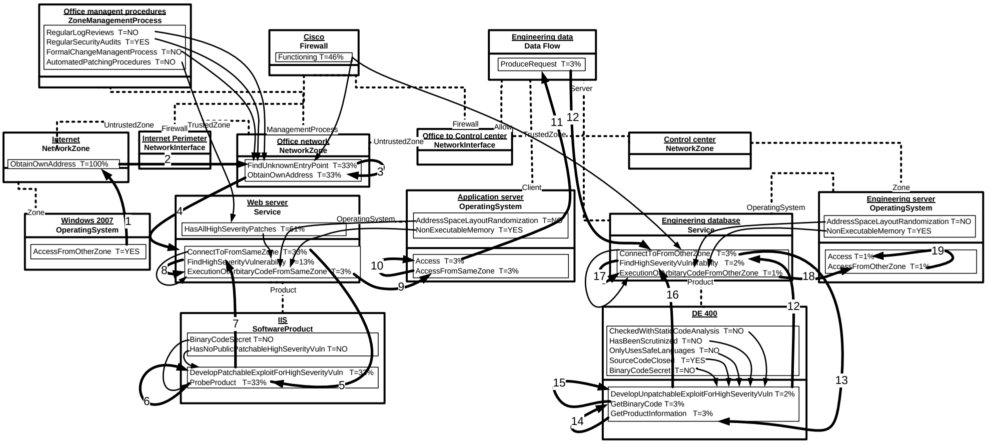


Figure 5. Excerpts from an instance model. A 19-step attack path and probabilities that each step along this path will be reached. The order the path is traversed is shown the enumerated arcs.

Some initial steps have been taken to test and validate the propositions made in CySeMoL through observations. In [129] observations related to remote arbitrary code exploits are made in conjunction with a cyber security exercise, in [96] a formal test of intrusion detection systems' operational effectiveness is made for one scenario and in [130] a formal test is made for one of the propositions CySeMoL makes regarding signature based intrusion detection. These tests corroborates propositions put forward by CySeMoL, however, they only cover a small portion of the theory and only [96] have the threat agent CySeMoL's theory is built around. Yet, they demonstrate the possibility to arrange formal tests of CySeMoL's validity.

A broader test of CySeMoL's convergent validity has been performed by comparing the predictions produced on a high level of abstraction to the predictions made by domain experts concerning a set of system architectures. In the test, the reasonableness of estimates made by CySeMoL was compared to the reasonableness of estimates made by five domain experts and three novices in cyber security. Of the six "experts", CySeMoL ends up in fourth place with respect to mean score, and fifth place with respect to median score. Overall, the test does not show an alarming difference between its ratings and the real experts' ratings. In addition, CySeMoL is rated as more reasonable than all the three novices. This test is further described in paper E.

3.7 Prescriptive statements

The theory of CySeMoL does not prescribe how a decision maker should go about achieving an optimal cyber security solution. The primary reason for this is that the theory does not include a number of variables that are required when the utility of a solution is to be assessed, including:

a) The consequence of attacks and the influence of contingency measures on this consequence, for instance, the cost of an unavailable SCADA server.

b) All threat agents that are relevant for a decision maker, for instance, insiders within SCADA system suppliers or undirected malicious code.

c) The mental model of threat agents and how often they attempt attacks of different types, for instance, how often they are likely to attempt attacks involving social engineering.

d) The business value (or cost) associated with different architectures, for instance, the value of making historical measurements available to IT systems in administrative office networks.

The abstract PRM template suggests how theories on a), b), and c) could be integrated with the theory presented in this thesis. The output of a theory that encompasses all constructs in the abstract PRM template could then be contrasted to the output of methods that assess the business value of an enterprise architecture, i.e., paragraph d). For instance, the method described in [131] could be used.

While important variables are outside the scope of the theory, and CySeMoL cannot be used to produce prescriptive statements directly, the theory can be used to produce prescriptive statements when these variables values have been assessed. The vulnerability estimates produced by CySeMoL can also be used to produce prescriptive statements ceteris paribus. Clearly, a less vulnerable architecture is desirable if all other variables remain unchanged. When perceptive statements are produced it is important to remember that CySeMoL produces rough approximations. It does not produce exact success probabilities.

4 Research design

This section gives an overview of the methodological aspects that have guided the research. The description is process-oriented and each sub-section corresponds to a phase in the research. These phases are (cf. Figure 6): framework and formalism, qualitative theory, quantitative theory and validation. The methods used for data collection and analysis within each of these phases are described.

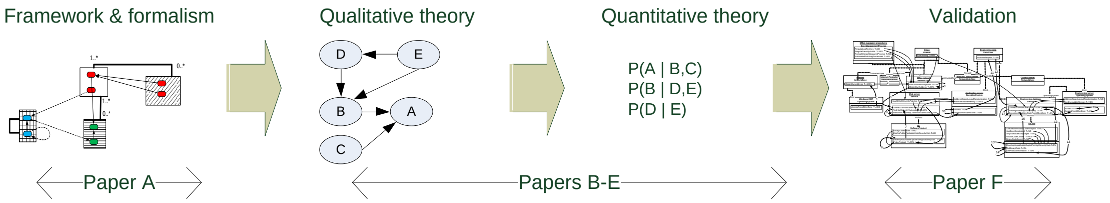


Figure 6. Phases in the research.

4.1 Framework and formalism

The primary purpose of this research is to support decision makers when they need to assess the cyber security of their SCADA systems. While the cyber security issues pertaining to SCADA systems are fairly new, a substantial theoretical body is available with the security field as a whole. This research reviewed existing literature in the field and compared it with the needs of decision makers in the SCADA domain. A number of methods and models have been proposed to address the problem of measuring cyber security, however, none of these were found to fit the needs in their present state (section 2 explained why).

Literature was the primary information source used when the framework used in this research was developed. The result combined qualitative models found in literature with a mathematical formalism and puts these into a framework which allows causal cyber security theory to be coupled with architectural models. As this framework was used as a basis, it has an influence on the approach used in other parts of this research. The framework approaches cyber security assessments as risk assessments and aims at quantifying the monetary risk associated with different architectures, i.e., the probability of unwanted events and the expected consequences of these events. The framework also directs the theory developer to model the attacks that give rise to the risk and the influence of countermeasures that reduce it. The primary sources of inspiration for this framework are Common Criteria's and its conceptual model [21], time-based-security [27], attack-modeling [32], [113], [114] and monetary security risk assessments [40], [132]. The formalism used to couple this framework to architectural models was that of PRMs [123]. The result was the abstract PRM template described in section 3.1 and paper A.

4.2 Qualitative theory

The framework (or PRM template) was used to develop a qualitative theory over cyber security. This qualitative theory details the PRM's: classes, reference slots, attributes and attribute relationships. In other words, it details everything except the conditional probabilities of the PRM.

An extensive literature review and interviews with experts in the cyber security domain were the primary sources for this theory. The objective was to produce a qualitative causal theory to support assessments of cyber security vulnerability. A subset of the framework was used for this purpose. To efficiently tackle practical issues relating to cyber security assessments this theory should offer a good tradeoff between the cost of applying the theory, the cost of quantifying the theory and the theory's accuracy.

First, literature was consulted to identify which attack steps to include. This literature study included review of a large number of textbooks (e.g. [133]), standards and reports (e.g. [9]), overview-articles (e.g. [104]) and security databases (e.g. [134]). After an initial model over attacks and assets had been created, literature on specific attacks was consulted. These sources were used to assess the parents to attack steps, i.e., countermeasures and states (completed attack steps) that literature suggests have an important influence on the probability that an attack step could be accomplished. A large number of sources were used for each type of attack. Examples of sources can be found in section 2.2 and in papers B-F.

The qualitative model was subsequently reviewed by domain experts. These reviews were made both on a high level of abstraction to ensure that the scope constituted a reasonable tradeoff and on a low level of abstraction to prioritize specific countermeasures and operationalize their definitions. Overall, these experts confirmed the prioritizations that had been made based on literature, but suggested some minor changes, e.g., to focus more on attacks on password authentication. For the reviews on a low level of abstraction, the number of reviewers used varied with the attack type. For instance, literature on social engineering was deemed sufficient to prioritize this field, while the details on remote code exploits was decided after a pilot study was made and after consulting three domain experts. Details concerning the expert reviews can be found in papers B-F.

4.3 Quantitative theory

The qualitative theory describes the relationships that need to be quantified. A large portion of the relationships could be quantified from the definition of constructs. An example of such a definitional relationship is that an attacker must possess an exploit code if he/she is to exploit a software vulnerability in a remote service. The relationships that cannot be determined from the definition of constructs were analyzed as in "probabilistic causal analysis" [122]. In other words, it was perceived as difficult to identify and control all variables that may influence the response variable's state. Since relevant variables are missing from the analysis the causal effect becomes uncertain (and probabilistic). In Bayesian terms, the omitted variables can be seen as marginalized [124].

Two methods were employed to assess probabilities. When reliable data could be found in the literature this data was used. When no reliable approximations could be found, data was elicited from domain experts.

Searches for data in literature were performed in article indexing services (e.g., Scopus and Google Scholar). They aimed at finding studies that contained data on the relationships specified in the qualitative theory. To quantify a relationship using secondary data the study should not only be of sufficient quality, but the variables studies should also match the variables and variable-relationships prescribed in the qualitative model. A number of relationships were possible to quantify using quantitative data from previous research in the field. Research on password security ([135–138]), network misconfigurations ([56], [57]) and social engineering ([52–55]) was directly used to determine variables' probability distributions given the conditions specified.

When the literature review was unable to find the data required it appeared not to be because the research community had ignored the relationship in question. The problem was rather that is was difficult for a researcher to quantify the relationship through observation in a manner that made the result generalizable. For instance, testing intrusion detection systems is associated with a number of issues, such as producing representative attacks and representative background traffic [88], [90]. In order to produce a quantitative theory that could approximate these relationships the judgment of domain experts was used.

Experts in the scientific community were the primary respondents in these surveys. However, a number of practitioners were also included. Researchers were identified from their publications; practitioners were identified based on peer-recommendation. Web surveys were used as the elicitation instrument. Since estimation of probability distributions is known to be problematic [139] care was taken with the construction of the web survey. The reliability of the question format was confirmed using Cronbach's alpha [140], [141] and all surveys were qualitatively reviewed by members of the target population.

Research in the field of expert judgment elicitation suggests that the result is better calibrated when multiple experts are used [142]. A number of techniques has been suggested for combining expert judgment, including: equal-weight, consensus methods [143], [144], the Cochran-Weiss-Shanteau index [145], self-proclaimed expertise [146], experience [147], certifications [147], peer-recommendations [147], and Cooke's classical method [148]. There is little research that compares the accuracy that these methods yield. This research uses the scheme proposed in Cooke's classical method [148]. Cooke's classical method has been shown to outperform both the best expert in a group, and the equal-weight combination of all experts' assessments. It is a performance based method which assigns weights based on the experts' ability to answer a set of test questions (called "seed questions") in a calibrated (i.e., accurate) and informative (i.e., precise) way. In the presented research these questions were constructed from previous research results in the field in question.

More elaborate descriptions of the elicitation process and the implementation of Cooke's classical method are given in papers B-E.

4.4 Validation

The interviews undertaken during theory development provided a qualitative validation of the relationships included in the theory. The surveys described in papers B-E also validated the prioritizations underlying the theory by asking respondents to suggest improvements. The few changes suggested by the respondents were diverse. In addition to this validation, CySeMoL's practical utility has been validated in three case studies, and the reasonableness of its assessments has been validated with a variant of the Turing test.

The scopes of the three case studies were: (1) the control center and adjacent environments in one of Sweden's three largest electrical power utilities, (2) electrical substations and remote communication to these owned by one of Sweden's largest power system owners and (3) reference architectures for one of the world's most commonly used electrical power management systems. The case studies demonstrated that the theory served as a usable tool for architecture analysis and pointed to practical improvements which would increase usability of the software tool.

A variant of the Turing test was used to test CySeMoL's validity [149]. In the classical Turing test a machine shall behave in a way indistinguishable from humans. These tests are especially useful for testing expert systems in situations such as the present – where the true answers to test cases are unknown (or very costly to determine), and it cannot be assumed that one particular domain expert is correct [150]. The test of CySeMoL was similar to the tests described in [68] and [71] and had two pools of human experts: one that produced assessments of the same type as the expert system and one that evaluated the first pool's assessments and the expert system's assessments based on how reasonable they are. The idea is that the expert system (i.e., CySeMoL) should receive ratings for the evaluators that are similar to the ratings received by the real experts. To test if the evaluators could recognize expertise, the test also included a pool of information system experts which were novices in the cyber security field. These novices' assessments were evaluated in the same manner as the assessments made by the experts and CySeMoL. If the evaluators recognize expertise the novices should receive comparably low ratings.

The pool of experts that produced assessments of the same type as CySeMoL consisted of five persons. The pool of cyber security novices consisted of three persons, and the pool that rated the assessments reasonableness consisted of two persons. The sample size prohibits reliable statistical conclusions from this test. The variation between the evaluators' scoring of the solutions suggests that the result should be interpreted with care. However, the summary statistics indicates that CySeMoL's assessments are comparable to those of a domain expert. In terms of mean score CySeMoL's comes in a tied fourth place; in terms of median score CySeMoL is placed on fifth. It also appears as if the evaluators' ratings are meaningful – there is a clear difference between the ratings that novices receive and the ratings that experts receive.

A more thorough description of the qualitative validation made on variables and relationships can be found in papers A-E. In paper F a more thorough description of the validation Turing test is given.

As described in 3.6, some initial attempts were made to validate the theory through formal experiments. In [96], [130] two experiments concerning intrusion detection systems are described. In [96] a formal test of intrusion detection systems' operational effectiveness is made. This test roughly corresponds to one of the intrusion detection scenarios in CySeMoL. The test in [96] gave a detection rate of 58 percent, and the value CySeMoL predicts is 59 percent. In [130] a formal test is made concerning the possibility to detect zero-day attacks (i.e., new and novel attacks) with signature based intrusion detection systems. As predicted by CySeMoL (c.f. paper D) it shows that signature based systems can detect zero-day attacks. In addition to these experiments [129] describes less reliable observations made in conjunction to a cyber security exercise. The observations concern remote arbitrary code exploits performed by a different threat agent under tighter time-constraints than about which the threat agent CySeMoL makes predictions. The observations made in [129] correspond to two scenarios predicted in CySeMoL's theory (one variable in CySeMoL is unknown for the observations). CySeMoL predicts these two scenarios to be successful with 43 percent and 67 percent probability while the observed frequency was 33 percent. Since the observed threat agent was less resourceful than the one CySeMoL makes predictions about the lower value offers some (albeit weak) support for CySeMoL's theory. Additional testing and refinement of CySeMoL's theory is suggested as future work.

5 References

[1] T. Cegrell, *Power system control technology*. Cambridge, Great britain: Prentice hall International, 1986.

[2] J. Andersson, E. Johansson, M. Haglind, and L. Johansson, "State-of-the-art study of commercial industrial control systems," Royal Institute of Technology - KTH, Stockholm, Sweden, 1997.

[3] GLEG Ltd, "Agora Exploit Pack Developer- SCADA+ pack," *(website)*, 2012. [Online]. Available: http://www.gleg.net/agora_scada.shtml. [Accessed: 20-Jun-2012].

[4] T. Sommestad, G. Björkman, and M. Ekstedt, "Information System Architectures in Electrical Distribution Utilities," in *Proceedings of NORDAC 2010*, 2010.

[5] E. Johansson, T. Sommestad, and M. Ekstedt, "Issues of Cyber Security In Scada-Systems-on the Importance of Awareness," in *The 20th International Conference on Electricity Distribution (CIRED)*, 2009.

[6] R. Fink, D. Spencer, and R. Wells, "Lessons learned from cyber security assessments of SCADA and energy management systems," *US Department of Energy*, no. September, 2006.

[7] U.S. Department of Energy, "Common Cyber Security Vulnerabilities Observed in Control System Assessments by the INL NSTB Program," Idaho Falls, 2008.

[8] US Department of Homeland Security, "Catalog of Control Systems Security: Recommendations for Standards Developers," 2008.

[9] K. Stouffer, J. Falco, and K. Kent, "Guide to Industrial Control Systems (ICS) Security Recommendations of the National Institute of Standards and Technology," *Nist Special Publication 800- 82*, 2008.

[10] IEC, "TS 62351-1: Power systems management and associated information exchange Data and communications security, Part 1:Communication network and system security Introduction to security issues," Geneva,Switzerland, 2007.

[11] NERC, "NERC CIP 002-009," 2007.

[12] TOGAF, "The Open Group Architecture Framework (TOGAF) - version 9." The Open Group, 2009.

[13] MODAF, "MOD Architecture Framework (MODAF)," 2012. [Online]. Available: www.modaf.org.uk. [Accessed: 17-Jul-2012].

[14] Department of Defense Architecture Framework Working Group, "DoD Architecture Framework, version 1.5," 2007.

[15] P. Johnson, L. Nordström, and R. Lagerström, "Formalizing analysis of enterprise architecture," in *Interoperability for Enterprise Software and Applications Conference*, 2006, p. 10.

[16] P. Johnson, R. Lagerström, P. Närman, and M. Simonsson, "Extended Influence Diagrams for System Quality Analysis," *Journal of Software*, vol. 2, no. 3, pp. 30-42, Sep. 2007.

[17] MODAF, "FAQs: How does MODAF represent security?," 2008. [Online]. Available: http://www.mod.uk/NR/rdonlyres/6F2454B0-48A3-4E61-90A4-F420CF9F3F1C/0/20090521_MODAF_1_2_FAQs_How_MODAF_Can_Reflect_Security_Concerns_V1_0_U.pdf. [Accessed: 17-Jul-2012].

[18] International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC)., "ISO/IEC 27004: Information technology -- Security techniques -- Information security management -- Measurement.," 2009.

[19] M. Swanson, N. Bartol, J. Sabato, J. Hash, and Laurie Graffo, "Security Metrics Guide for Information Technology Systems," *NIST Special Publications*, vol. 800, no. 55, 2003.

[20] M. S. Lund, B. Solhaug, and K. Stolen, *Model-driven risk analysis: the CORAS approach*. Springer Verlag, 2011.

[21] "Common Criteria for Information Technology Security Evaluation Part 2 : Security functional components September 2007 Revision 2 Foreword," no. September, pp. 1-324, 2007.

[22] E. Johansson, "Assessment of Enterprise Information Security–How to make it Credible and efficient," KTH - The Royal Insitute of Technology, 2005.

[23] J. Sherwood, A. Clark, and D. Lynas, *Enterprise Security Architecture: A Business-Driven Approach*. USA: CMP Books, 2005.

[24] G. Stoneburner, "Underlying Technical Models for Information Technology Security," *Nist Special Publications 800-33*, Dec. 2001.

[25] A. Anderson, D. Longley, and L. F. Kwok, "Security modelling for organisations," in *CCS '94: Proceedings of the 2nd ACM Conference on Computer and communications security*, 1994, pp. 241-250.

[26] D. Bodeaum, "A conceptual model for computer security risk analysis," in *Proceedings Computer Security Applications Conference, 1992. ., Eighth Annual*, 1992, pp. 56–63.

[27] W. Schwartau, "Time-based security explained: Provable security models and formulas for the practitioner and vendor," *Computers & Security*, vol. 17, no. 8, pp. 693-714, 1998.

[28] L. Pirzadeh and E. Jonsson, "A Cause and Effect Approach towards Risk Analysis," *2011 Third International Workshop on Security Measurements and Metrics*, pp. 80-83, Sep. 2011.

[29] E. Jonsson, "Towards an integrated conceptual model of security and dependability," *First International Conference on Availability, Reliability and Security*, pp. 646-653, 2006.

[30] M. Beccuti et al., "Quantification of dependencies in electrical and information infrastructures: The CRUTIAL approach," in *2009 Fourth International Conference on Critical Infrastructures*, 2009, pp. 1-8.

[31] T. Fleury, H. Khurana, and V. Welch, "Towards a taxonomy of attacks against energy control systems," in *Critical Infrastructure Protection II*, no. March 2008, Springer US, 2009, p. 71-85.

[32] B. Schneier, "Attack trees: Modeling security threats," *Dr. Dobb's Journal*, 1999.

[33] M. Howard and D. C. LeBlanc, *Writing Secure Code*. Redmond, WA, USA: Microsoft Press, 2002.

[34] S. Bistarelli, F. Fioravanti., and P. Peretti., "Defense trees for economic evaluation of security investments," in *Proceedings of the First International Conference on Availability, Reliability and Security* , pp. 416-423, 2006.

[35] K. Edge, R. Raines, M. Grimaila, R. Baldwin, R. Bennington, and C. Reuter, "The Use of Attack and Protection Trees to Analyze Security for an Online Banking System," *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, p. 144b-144b, Jan. 2007.

[36] L. Piètre-Cambacédès and M. Bouissou, "Beyond Attack Trees: Dynamic Security Modeling with Boolean Logic Driven Markov Processes (BDMP)," *2010 European Dependable Computing Conference*, pp. 199-208, 2010.

[37] J. Hallberg, N. Hallberg, and A. Hunstad, "Crossroads and XMASS: Framework and method for system it security assessment," Linköping, Sweden, 2006.

[38] B. Karabacak and I. Sogukpinar, "ISRAM: information security risk analysis method," *Computers & Security*, vol. 24, no. 2, pp. 147-159, 2005.

[39] G. Stoneburner, A. Goguen, and A. Feringa, "Risk management guide for information technology systems," *NIST Special Publication 800-3*0, 2002.

[40] L. A. Gordon and M. P. Loeb, *Managing Cybersecurity Resources: A Cost-Benefit Analysis*, vol. The Mcgraw. New York, NY, USA: Mcgraw-Hill, 2006.

[41] H. Mouratidis, P. Giorgini, G. Manson, and I. Philp, "A natural extension of tropos methodology for modelling security," in *the Proceedings of the Agent Oriented Methodologies Workshop (OOPSLA 2002)*, 2002.

[42] C.-W. Ten, C.-C. Liu, and M. Govindarasu, "Vulnerability Assessment of Cybersecurity for SCADA Systems Using Attack Trees," *2007 IEEE Power Engineering Society General Meeting*, vol. 2, pp. 1-8, Jun. 2007.

[43] M. McQueen, W. Boyer, S. McBride, M. Farrar, and Z. Tudor, "Measurable Control System Security through Ideal Driven Technical Metrics," in *S4: SCADA Security Scientific Symposium*, 2008.

[44] G. Dondossola, F. Garrone, and J. Szanto, "Cyber Risk Assessment of Power Control Systems – A Metrics weighed by Attack Experiments," in *Proceedings of IEEE Power and Energy Society General Meeting*, pp. 1-9, 2011.

[45] R. Breu, F. Innerhofer-Oberperfler, and A. Yautsiukhin, "Quantitative Assessment of Enterprise Security System," *2008 Third International Conference on Availability, Reliability and Security*, pp. 921-928, Mar. 2008.

[46] E. J. Byres, M. Franz, and D. Miller, "The use of attack trees in assessing vulnerabilities in SCADA systems," in *International Infrastructure Survivability Workshop (IISW'04)*, 2004.

[47] I. Nai Fovinoa, A. Carcanoa, M. Maseraa, and A. Trombettab, "An experimental investigation of malware attacks on SCADA systems," *International Journal of Critical Infrastructure Protection*, vol. 2, no. 4, pp. 139-145, 2009.

[48] G. Dondossola, J. Szanto, M. Masera, and I. N. Fovino, "Effects of intentional threats to power substation control systems," *International Journal of Critical Infrastructures*, vol. 4, no. 1/2, p. 129, 2008.

[49] N. SCADA and U.S. Department of Energy, "Common Cyber Security Vulnerabilities Observed in Control System Assessments by the INL NSTB Program," Idaho Falls, 2008.

[50] G. Dondossola, F. Garrone, and J. Szanto, "Assessment of power control systems communications through testbed experiments," in *Electricity Distribution - Part 1, 2009. CIRED 2009. 20th International Conference and Exhibition on*, 2009, no. 0650, pp. 1-4.

[51] W. Jansen, "Directions in security metrics research," DIANE Publishing, Gaithersburg, MD, 2009.

[52] J. R. Jacobs, "Measuring the Effectiveness of the USB Flash Drive as a Vector for Social Engineering Attacks on Commercial and Residential Computer Systems," Embry Riddle Aeronautical University, 2011.

[53] S. Stasiukonis, "Social engineering, the USB way," *Dark Reading*, vol. 7, 2006.

[54] T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer, "Social phishing," *Communications of the ACM*, vol. 50, no. 10, pp. 94–100, Mar. 2007.

[55] R. Dodge and A. Ferguson, "Using Phishing for User Email Security Awareness," in *Security and Privacy in Dynamic Environments*, vol. 201, S. Fischer-Hübner, K. Rannenberg, L. Yngström, and S. Lindskog, Eds. Springer Boston, 2006, pp. 454-459.

[56] T. Sommestad, M. Ekstedt, H. Holm, and M. Afzal, "Security mistakes in information system deployment projects," *Information Management and Computer Security*, vol. 19, no. 2, 2011.

[57] A. Wool, "A quantitative study of firewall configuration errors," *Computer*, pp. 62–67, 2004.

[58] NIST Computer Security Resource Center (CSRC), "National Vulnerability Database," 2011. [Online]. Available: www.nvd.nist.org. [Accessed: 28-Apr-2011].

[59] The MITRE Corporation, "Common Weakness Enumeration," 2012. [Online]. Available: http://cwe.mitre.org/.

[60] Offensive Security, "Exploit Database," 2011. [Online]. Available: http://www.exploit-db.com/.

[61] A. Ozment, "Improving vulnerability discovery models," in *Proceedings of the 2007 ACM workshop on Quality of protection*, 2007, pp. 6–11.

[62] S.-W. Woo, H. Joh, O. H. Alhazmi, and Y. K. Malaiya, "Modeling vulnerability discovery process in Apache and IIS HTTP servers," *Computers & Security*, vol. 30, no. 1, pp. 50-62, Jan. 2011.

[63] O. H. Alhazmi and Y. K. Malaiya, "Modeling the Vulnerability Discovery Process," *16th IEEE International Symposium on Software Reliability Engineering (ISSRE'05)*, pp. 129-138, 2005.

[64] O. H. Alhazmi and Y. K. Malaiya, "Quantitative vulnerability assessment of systems software," in *Proceedings of Annual Reliability and Maintainability Symposium*, 2005, pp. 615-620.

[65] T. Gerace and H. Cavusoglu, "The critical elements of the patch management process," *Communications of the ACM*, vol. 52, no. 8, p. 117, Aug. 2009.

[66] B. David, P. Pongsin, S. Dawn, and Z. Jiang, "Automatic patch-based exploit generation is possible: Techniques and implications," *IEEE Symposium on Security and Privacy*, pp. 143-157, May 2008.

[67] M. A. McQueen, T. A. McQueen, W. F. Boyer, and M. R. Chaffin, "Empirical estimates and observations of 0day vulnerabilities," in *System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on*, 2009, pp. 1–12.

[68] J. Wilander and M. Kamkar, "A comparison of publicly available tools for dynamic buffer overflow prevention," in *Proceedings of the 10th Network and Distributed System Security Symposium*, 2003, pp. 149–162.

[69] C. Cowan, P. Wagle, C. Pu, S. Beattie, and J. Walpole, "Buffer Overflows : Attacks and Defenses for the Vulnerability of the Decade," in *Foundations of Intrusion Tolerant Systems, 2003 [Organically Assured and Survivable Information Systems]*, 2003, pp. 227-237.

[70] I. Simon, "A comparative analysis of methods of defense against buffer overflow attacks," *Web address: http://www. mcs. csuhayward. edu/\~ simon/security/boflo. html*, pp. 1-16, 2001.

[71] N. Frykholm, "Countermeasures against buffer overflow attacks," *RSA Tech Note*, pp. 1-9, 2000.

[72] Y. Kim, J. Lee, H. Han, and K.-M. Choe, "Filtering false alarms of buffer overflow analysis using SMT solvers," *Information and Software Technology*, vol. 52, no. 2, pp. 210-219, Feb. 2010.

[73] X. Wang, C. C. Pan, P. Liu, and S. Zhu, "SigFree: A Signature-Free Buffer Overflow Attack Blocker," | *IEEE Transactions on Dependable and Secure Computing*, pp. 65–79, 2008.

[74] B. Salamat, A. Gal, T. Jackson, K. Manivannan, G. Wagner, and M. Franz, "Multi-variant Program Execution: Using Multi-core Systems to Defuse Buffer-Overflow Vulnerabilities," in *2008 International Conference on Complex, Intelligent and Software Intensive Systems*, 2008, pp. 843-848.

[75] S. H. Yong and S. Horwitz, "Protecting C programs from attacks via invalid pointer dereferences," *ACM SIGSOFT Software Engineering Notes*, vol. 28, no. 5, p. 307, Sep. 2003.

[76] J. Wilander, N. Nikiforakis, Y. Younan, M. Kamkar, and W. Joosen, "RIPE: Runtime Intrusion Prevention Evaluator," in *In Proceedings of the 27th Annual Computer Security Applications Conference, ACSAC*, 2011.

[77] Y. Younan, "Efficient countermeasures for software vulnerabilities due to memory management errors," Katholieke Universiteit Leuven, 2008.

[78] H. Shacham, M. Page, B. Pfaff, E. J. Goh, N. Modadugu, and D. Boneh, "On the effectiveness of address-space

randomization," in *Proceedings of the 11th ACM conference on Computer and communications security*, 2004, pp. 298–307.

[79] PaX. Team, "PaX address space layout randomization (ASLR)." [Online] 2003. http://pax.grsecurity.net/docs/aslr.txt

[80] U. Erlingsson, "Low-level Software Security : Attacks and Defenses Low-level Software Security : Attacks and Defenses," Redmond, WA, USA, 2007.

[81] T. Jim, G. Morrisett, D. Grossman, M. Hicks, J. Cheney, and Y. Wang, "Cyclone: A safe dialect of C," in *USENIX*, 2002, pp. 275-288.

[82] J. Newsome, "Dynamic taint analysis for automatic detection, analysis, and signature generation of exploits on commodity software," *Network and Distributed System Security*, no. May 2004, 2005.

[83] R. Lippmann et al., "Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation," *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*, pp. 12-26, 1998.

[84] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 DARPA on-line intrusion detection evaluation," *Computer Networks*, vol. 34, 2000.

[85] K. Salah and a. Kahtani, "Improving Snort performance under Linux," *IET Communications*, vol. 3, no. 12, p. 1883, 2009.

[86] F. Alserhani, M. Akhlaq, I. U. Awan, J. Mellor, A. J. Cullen, and P. Mirchandani, "Evaluating Intrusion Detection Systems in High Speed Networks," *2009 Fifth International Conference on Information Assurance and Security*, pp. 454-459, 2009.

[87] F. B. Ktata, N. E. Kadhi, and K. Ghédira, "Agent IDS based on Misuse Approach," *Journal of Software*, vol. 4, no. 6, pp. 495-507, Aug. 2009.

[88] J. McHugh, "Testing Intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory," *ACM Transactions on Information and System Security*, vol. 3, no. 4, pp. 262-294, Nov. 2000.

[89] B. I. A. Barry and H. A. Chan, "Intrusion detection systems," in *Handbook of Information and Communication Security*, vol. 2001, no. 6, P. Stavroulakis and M. Stamp, Eds. Springer, 2010, pp. 193-205.

[90] P. Mell, V. Hu, R. Lippmann, J. Haines, and M. Zissman, "An overview of issues in testing intrusion detection systems," *Citeseer*. National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, 2003.

[91] M. J. Ranum, "Experiences Benchmarking Intrusion Detection Systems," *Security*, pp. 1-10, 2001.

[92] R. Werlinger, K. Hawkey, and K. Muldner, "The challenges of using an intrusion detection system: is it worth the effort?," *SOUPS '08 Proceedings of the 4th symposium on Usable privacy and security*, no. 1, 2008.

[93] R. Werlinger, K. Muldner, K. Hawkey, and K. Beznosov, “Preparation, detection, and analysis: the diagnostic work of IT security incident response,” *Information Management & Computer Security*, vol. 18, no. 1, pp. 26–42, 2010.

[94] J. R. Goodall, W. G. Lutters, and A. Komlodi, “I know my network: collaboration and expertise in intrusion detection,” in *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, 2004, pp. 342–345.

[95] J. R. Goodall, W. G. Lutters, and A. Komlodi, “Developing expertise for network intrusion detection,” *Information Technology & People*, vol. 22, no. 2, pp. 92–108, 2009.

[96] T. Sommestad and A. Hunstad, “Intrusion detection and the role of the system administrator,” in *Proceedings of International Symposium on Human Aspects of Information Security & Assurance*, 2012.

[97] R. Chertov, S. Fahmy, and N. B. Shroff, “Emulation versus simulation: A case study of TCP-targeted denial of service attacks,” in *Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006. 2nd International Conference on*, 2006, p. 10–pp.

[98] C. Sangpachatanaruk, S. M. Khattab, T. Znati, R. Melhem, and D. Mossé, “A simulation study of the proactive server roaming for mitigating denial of service attacks,” in *Proceedings of the 36th annual symposium on Simulation*, 2003, p. 7.

[99] S. M. Khattab, C. Sangpachatanaruk, R. Melhem, and T. Znati, “Proactive server roaming for mitigating denial-of-service attacks,” in *Information Technology: Research and Education, 2003. Proceedings. ITRE2003. International Conference on*, 2003, pp. 286–290.

[100] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage, “Inferring Internet denial-of-service activity,” *ACM Transactions on Computer Systems*, vol. 24, no. 2, pp. 115-139, May 2006.

[101] V. Gupta, S. Krishnamurthy, and M. Faloutsos, “Denial of service attacks at the MAC layer in wireless ad hoc networks,” in *MILCOM 2002*, 2002, vol. 2, pp. 1118–1123.

[102] J. Jung, B. Krishnamurthy, and M. Rabinovich, “Flash crowds and denial of service attacks: Characterization and implications for CDNs and web sites,” in *Proceedings of the 11th international conference on World Wide Web*, 2002, pp. 293–304.

[103] J. Mirkovic et al., “Testing a Collaborative DDoS Defense In a Red Team/Blue Team Exercise,” *IEEE Transactions on Computers*, vol. 57, no. 8, pp. 1098-1112, Aug. 2008.

[104] J. Mirkovic and P. Reiher, “A taxonomy of DDoS attack and DDoS defense mechanisms,” *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, p. 39, Apr. 2004.

[105] J. Mirkovic and P. Reiher, “A Taxonomy of DDoS Attack and DDoS Defense Mechanisms,” *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, 2004.

[106] C. Douligeris, "DDoS attacks and defense mechanisms: classification and state-of-the-art," *Computer Networks*, vol. 44, no. 5, pp. 643-666, Apr. 2004.

[107] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-based defense mechanisms countering the DoS and DDoS problems," *ACM Computing Surveys*, vol. 39, no. 1, p. 3-es, 2007.

[108] M. Glenn, "A Summary of DoS/DDoS Prevention, Monitoring and Mitigation Techniques in a Service Provider Environment," 2003.

[109] K. A. Lee, "CS2SAT : The Control Systems Cyber Security Self-Assessment Tool Tool," Idaho Falls, Idaho, USA, 2008.

[110] US-CERT, "Cyber Security Evaluation Tool (CSET)," 2012. [Online]. Available: http://www.us-cert.gov/control_systems/satool.html. [Accessed: 30-Apr-2012].

[111] T. Heberlein, M. Bishop, E. Ceesay, M. Danforth, and CG, "A Taxonomy for Comparing Attack-Graph Approaches," *netsq.com*, pp. 1-14, 2004.

[112] S. Roschke, F. Cheng, R. Schuppenies, and C. Meinel, "Towards Unifying Vulnerability Information for Attack Graph Construction," in *Proceedings of the 12th International Conference on Information Security*, 2009, p. 233.

[113] L. P. Swiler, C. Phillips, D. Ellis, and S. Chakerian, "Computer-attack graph generation tool," in *Proceedings DARPA Information Survivability Conference and Exposition II. DISCEX'01*, 2000, pp. 307-321.

[114] O. M. Sheyner, "Scenario graphs and attack graphs," Carnegie Mellon University, 2004.

[115] R. Lippmann, "Netspa: A network security planning architecture," Massachusetts Institute of Technology, 2002.

[116] R. Lippmann et al., "Validating and restoring defense in depth using attack graphs," in *MILCOM*, pp. 1-10, 2006.

[117] S. Jajodia, "Topological analysis of network attack vulnerability," *Proceedings of the 2nd ACM symposium on Information, computer and communications security - ASIACCS '07*, p. 2, 2007.

[118] S. Jajodia, S. Noel, and B. O'Berry, "Topological analysis of network attack vulnerability," in *Managing Cyber Threats: Issues, Approaches and Challanges, chapter 5. Kluwer Academic Publisher*, V. Kumar, J. Srivastava, and A. Lazarevic, Eds. Springer US, 2003, pp. 247-266.

[119] S. Noel, M. Elder, S. Jajodia, P. Kalapa, S. O'Hare, and K. Prole, *Advances in Topological Vulnerability Analysis*. Washington, DC: IEEE, 2009, pp. 124-129.

[120] J. Homer, K. Manhattan, X. Ou, and D. Schmidt, "A Sound and Practical Approach to Quantifying Security Risk in Enterprise Networks," Kansas State University, 2010.

[121] H. Holm, T. Sommestad, J. Almroth, and M. Persson, “A quantitative evaluation of vulnerability scanning,” *Information Management & Computer Security*, vol. 19, no. 4, 2011.

[122] S. Gregor, “The nature of theory in information systems,” *Management Information Systems Quarterly*, pp. 1-45, 2006.

[123] L. Getoor, N. Friedman, D. Koller, A. Pfeffer, and B. Taskar, “Probabilistic Relational Models,” in *Introduction to Statistical Relational Learning*, L. Getoor and B. Taskar, Eds. MIT Press, 2007, pp. 129-175.

[124] F. . Jensen, *Bayesian Networks and Decision Graphs*. Secaucus, NJ, USA.: Springer New York, 2001.

[125] OMG, “OMG Unified Modeling Language (OMG UML), Infrastructure,” 2009.

[126] F. . Jensen, *Bayesian Networks and Decision Graphs*. Secaucus, NJ, USA.: Springer New York, 2001.

[127] P. Mell, K. Scarfone, and S. Romanosky, “A Complete Guide to the Common Vulnerability Scoring System (CVSS), Version 2.0, Forum of Incident Response and Security Teams.” 2007.

[128] D. Ahmad, “The Contemporary Software Security Landscape,” *IEEE Security & Privacy Magazine*, vol. 5, no. 3, pp. 75-77, May 2007.

[129] H. Holm, T. Sommestad, U. Franke, and M. Ekstedt, “Success rate of remote code execution attacks – expert assessments and observations,” *Journal of Universal Computer Science*, vol. 18, no. 6, pp. 732-749, 2012.

[130] H. Holm, “Empirical analysis of signature based intrusion detection for zero-day exploits,” *Working paper,* Royal Institute of Technology, 2012.

[131] M. Gammelgård, “Business value assessment of it investments an evaluation method applied to the electrical power industry,” Royal Institute of Technology (KTH), 2007.

[132] H. Cavusoglu, B. Mishra, and S. Raghunathan, “A model for evaluating it security investments,” *Communications of the ACM*, vol. 47, no. 7, pp. 87-92, 2004.

[133] R. J. Anderson, *Security Engineering: A guide to building dependable distributed systems*. New York, NY, USA: Wiley Publishing, 2008.

[134] The MITRE Corporation, “The Common Attack Pattern Enumeration and Classification,” *(website)*, 2011. [Online]. Available: http://capec.mitre.org/.

[135] S. Marechal, “Advances in password cracking,” *Journal in Computer Virology*, vol. 4, no. 1, pp. 73-81, 2007.

[136] M. Dell’ Amico, P. Michiardi, and Y. Roudier, “Password Strength: An Empirical Analysis,” *2010 Proceedings IEEE INFOCOM*, pp. 1-9, Mar. 2010.

[137] J. Cazier, “Password security: An empirical investigation into e-commerce passwords and their crack times,” *Information Security Journal: A Global*, 2006.

[138] "Free Rainbow Tables," 2011. [Online]. Available: http://www.freerainbowtables.com/. [Accessed: 01-Apr-2011].

[139] P. H. Garthwaite, J. B. Kadane, and A. O'Hagan, "Statistical methods for eliciting probability distributions," *Journal of the American Statistical Association*, vol. 100, no. 470, pp. 680-701, 2005.

[140] L. J. Cronbach and R. J. Shavelson, "My Current Thoughts on Coefficient Alpha and Successor Procedures," *Educational and Psychological Measurement*, vol. 64, no. 3, pp. 391-418, Jun. 2004.

[141] L. J. Cronbach, "Coefficient alpha and the internal structure of tests," *Psychometrika*, vol. 16, no. 3, pp. 297-334, 1951.

[142] R. T. Clemen and R. L. Winkler, "Combining probability distributions from experts in risk analysis," *Risk Analysis*, vol. 19, no. 187, pp. 187-204, 1999.

[143] A. Fink, J. Kosecoff, M. Chassin, and R. H. Brook, "Consensus methods: characteristics and guidelines for use.," *American journal of public health*, vol. 74, no. 9, pp. 979-83, Sep. 1984.

[144] A. H. Ashton, "Does consensus imply accuracy in accounting studies of decision making?," *The Accounting Review*, vol. 60, no. 2, pp. 173–185, 1985.

[145] D. J. Weiss and J. Shanteau, "Empirical Assessment of Expertise," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 45, no. 1, pp. 104-116, 2003.

[146] M. J. Abdolmohammadi and J. Shanteau, "Personal attributes of expert auditors," *Organizational Behavior and Human Decision Processes*, vol. 53, no. 2, pp. 158–172, 1992.

[147] J. Shanteau, D. J. Weiss, R. P. Thomas, and J. C. Pounds, "Performance-based assessment of expertise: How to decide if someone is an expert or not," *European Journal of Operational Research*, vol. 136, no. 2, pp. 253–263, 2002.

[148] R. M. Cooke, *Experts in Uncertainty: Opinions and Subjective Probability in Science*. New York, New York, USA: Open University Press, 1991.

[149] R. French, "The Turing Test: the first 50 years.," *Trends in cognitive sciences*, vol. 4, no. 3, pp. 115-122, Mar. 2000.

[150] R. M. O'Keefe and D. E. O'Leary, "Expert system verification and validation: a survey and tutorial," *Artificial Intelligence Review*, vol. 7, no. 1, pp. 3-42, Feb. 1993.

[151] R. Agarwal, R. Kannan, and M. Tanniru, "Formal validation of a knowledge-based system using a variation of the Turing test," *Expert Systems with Applications*, vol. 6, no. 2, pp. 181-192, Apr. 1993.