Karin Sallhammar

Stochastic Models for Combined Security and Dependability Evaluation

Thesis for the degree of philosophiae doctor

Trondheim, June 2007

Norwegian University of
Science and Technology
Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Telematics

NTNU
Innovation and Creativity

NTNU
Norwegian University of Science and Technology

Thesis for the degree of philosophiae doctor

Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Telematics

ISBN 978-82-471-3280-7 (printed ver.)
ISBN 978-82-471-3277-7 (electronic ver.)
ISSN 1503-8181

Theses at NTNU, 2007:150

Printed by Tapir Uttrykk

Abstract

Security is a topic of ever increasing interest. Today it is widely accepted that, due to the unavoidable presence of vulnerabilities, design faults and administrative errors, an ICT system will never be totally secure. Connecting a system to a network will necessarily introduce a risk of inappropriate access resulting in disclosure, corruption and/or loss of information. Therefore, the security of a system should ideally be interpreted in a probabilistic manner. More specifically, there is an urgent need for modelling methods that provide operational *measures* of the security.

Dependability, on the other hand, is the ability of a computer system to deliver service that can justifiably be trusted. In a dependability context one distinguishes between accidental faults, which are modelled as random processes, and intentional faults, i.e., attacks, which in most cases are not considered at all. A major drawback of this approach is that attacks may in many cases be the dominating failure source for today's networked systems. The classical way of dependability evaluation can therefore be very deceptive: highly dependable systems may in reality fail much more frequently than expected, due to the exploitation from attackers.

To be considered trustworthy, a system must be both dependable *and* secure. However, these two aspects have so far tended to be treated separately. A unified modelling framework for security and dependability evaluation would be advantageous from both points of view. The security community can benefit from the mature dependability modelling techniques, which can provide the operational measures that are so desirable today. On the other hand, by adding hostile actions to the set of possible fault sources, the dependability community will be able to make more realistic models than the ones that are currently in use. This thesis proposes a stochastic modeling approach, which can be used to predict a system's security and dependability behavior. As will be seen, the basic model has a number of possible applications. For example, it can be used as a tool for trade-off analysis of security countermeasures, or it can be used as a basis for real-time assessment of the system trustworthiness.

Keywords Stochastic modeling and analysis, security quantification, security measures, security evaluation, integrating security and dependability, attack prediction, game theory, stochastic games, real-time risk assessment, agent-sensor architectures, distributed intrusion detection, hidden Markov models.

Preface

This dissertation was delivered in partial fulfilment of the requirement of the philosophiae doctor (Ph.d.) degree at the Norwegian University of Science and Technology. The work was performed at the Centre for Quantifiable Quality of Service (Q2S), Centre of Excellence (CoE), during 2003-2006, and has been supervised by Professor Svein J. Knapskog and Professor Bjarne E. Helvik. The Centre of Excellence is a national effort - initiated by the Norwegian Department of Education and Science

The document has been formatted in LaTeX under Mandriva Linux using a modified version of the document class *kapproc.cls* provided by Kluwer Academic Publishers.

Acknowledgements

Numereous people have directly or indirectly contributed to the work presented in this thesis. First of all I would like to thank my thesis advisors Professor Svein J. Knapskog and Professor Bjarne E. Helvik, who also are co-authors of most of the papers presented in this thesis. Your contributions, encouragement and insightful feedback have been invaluable during my work with the papers. Having two advisors did turn out to be so much better than having only one. Also during the most hectic periods at least one of you has found time to discuss and comment on my work. I would never have been able to finish this thesis without your support.

Thanks to the present and former Ph.D. students and postdocs in the security group at the Q2S Centre: Kjetil Haslum, André Årnes, Marie Moe and Tønnes Brekne. Working together with you have been both fun and academically inspiring. The numerous discussions that we have had have been both fruitful and valuable to the research presented in this thesis. Thanks also to all the other people at the Q2S Centre for contributing to the open and friendly work atmosphere that we share.

I would like to give my special appreciation to the administrative and technical staff at the Q2S Centre. Many thanks to Anniken Skotvoll for handling all kinds of administrative matters with a never-ending patience and accuracy, and to Hans Almåsbakk for providing a reliable working environment where, during my 3.5 years at the Centre, my work has never been disrupted by computer or network problems. Thanks also to Otto Wittner for your helpfulness regarding the formatting of this thesis.

Finally, I would like to thank my family: my boyfriend Karl-Johan and our baby Alva, my parents Jan and Eva-Lena and my siblings Olle and Malin for all the support and inspiration you have given me.

Contents

List of Papers

Publications Included in the Thesis

- PAPER A:
  Karin Sallhammar and Svein J. Knapskog. *Using Game Theory in Stochastic Models for Quantifying Security*. In Proceedings of the 9th Nordic Workshop on Secure IT-systems (Nordsec 2004). Espoo, Finland. November 4-5, 2004.

- PAPER B:
  K. Sallhammar, S. J. Knapskog and B. E. Helvik. *Using Stochastic Game Theory to Compute the Expected Behavior of Attackers*. In Proceedings of the 2005 International Symposium on Applications and the Internet (Saint 2005). Trento, Italy. January 31 - February 4, 2005.

- PAPER C:
  Karin Sallhammar, Bjarne E. Helvik and Svein J. Knapskog. *Incorporating Attacker Behavior in Stochastic Models of Security*. In Proceedings of the 2005 International Conference on Security and Management (SAM'05). Las Vegas, Nevada, USA. June 20-23, 2005.

- PAPER D:
  Karin Sallhammar, Bjarne E. Helvik and Svein J. Knapskog. *On Stochastic Modeling for Integrated Security and Dependability Evaluation*. The Journal of Networks (ISSN 1796-2056), Vol. 1, No. 5, September/October 2006.

- PAPER E:
  André Årnes, Karin Sallhammar, Kjetil Haslum, Tønnes Brekne, Marie Elisabeth Gaup Moe and Svein Johan Knapskog. *Real-time Risk Assessment with Network Sensors and Intrusion Detection Systems*. In Proceedings of the 2005 International Conference on Computational Intelligence and Security (CIS'05). Xian, China. December 15-19, 2005.

- PAPER F:
  Karin Sallhammar, Bjarne E. Helvik and Svein J. Knapskog. *A Framework for Predicting Security and Dependability Measures in Real-time*. International Journal of Computer Science and Network Security (IJCSNS), Vol. 7, No. 3, pp. 169-183, March 2007.

These papers are included as Part II of this thesis. Note that some of the papers have been subject to minor editorial changes since their publication.

Other Publications by the Author

- Siv Hilde Houmb and Karin Sallhammar. *Modeling System Integrity of a Security Critical System using Coloured Petri Nets*. In Proceedings of the 1st International Conference on Safety and Security Engineering (SAFE 2005). Rome, Italy. June 13-15, 2005.
- Karin Sallhammar, Bjarne E. Helvik and Svein J. Knapskog. *Towards a Stochastic Model for Integrated Security and Dependability Evaluation*. In Proceedings of the First International Conference on Availability, Reliability and Security (AReS 2006). Vienna, Austria. April 20-22, 2006.
- Karin Sallhammar, Bjarne E. Helvik and Svein J. Knapskog. *A Game-theoretic Approach to Stochastic Security and Dependability Evaluation*. In Proceedings of the 2nd IEEE International Symposium on Dependable Autonomic and Secure Computing (DASC'06). Indianapolis, Indiana, USA. September 29 - October 1, 2006.
- André Årnes, Karin Sallhammar, Kjetil Haslum and Svein Johan Knapskog. *Real-time Risk Assessment with Network Sensors and Hidden Markov Models*. In Proceedings of the 11th Nordic Workshop on Secure IT-systems (Nordsec 2006). Linköping, Sweden. October 19-20, 2006.
- Bjarne E. Helvik, Karin Sallhammar and Svein J. Knapskog. *Integrated Dependability and Security Evaluation using Game Theory and Markov Models.* In "Information Assurance: Dependability and Security in Networked Systems" (to be published in 2007, by Morgan Kaufmann Publishers, an imprint of ELSEVIER, INC.).

I

THESIS INTRODUCTION

Introduction

The main part of this thesis, Part II, is a paper collection consisting of six papers written during 2004-2006. In this part an introduction to the papers is given. Section 1 describes the background and explains the underlying motivation for this thesis. In Section 2 the main ideas that the research is derived from are presented. Section 3 provides an overview over the state of the art in research areas closely related to this thesis. Section 4 discusses the methodology that has been applied when working with the thesis. In Section 5 the underlying assumptions that the research is based on are pointed out. Section 6 gives a summary of the included papers and explains how the papers are related to each other. Guidelines for reading this thesis are provided in Section 7. Section 8 concludes the thesis by summarizing the main results obtained. Finally, possible future work is discussed in Section 9.

1. Background

The new paradigms of ubiquitous computing and high capacity data transfer have turned the Internet into today's main area for information interchange and electronic commerce. As network systems become more and more complex and interconnected, their security play an increasingly important role mainly because they are supporting critical applications. Attacks against computer networks used by modern society and economics for communication and finance can therefore threaten the economical and physical well-being of people and organizations. The security of an ICT system is hence a research area of ever increasing interest.

Lately, there has been an astonishing rapid deployment of new network services in complex applications. New requirements and the increasing competition in the industry put high demands on the product "time to market" when developing and releasing new computing systems. Security often plays a secondary role and is then a trade-off regarding the design and choice of security mechanisms versus the implementation effort and cost. In many cases developers do not take enough precaution to avoid that previous mistakes are repeated or that new vulnerabilities are introduced when releasing new products on the market. Numerous computing systems providing services to users that are connected to the Internet are therefore vulnerable to attacks, already from the very first beginning of their operation [LEH+97].

As if this was not enough, Internet is in itself a vulnerable place. Many of the early protocols that the network infrastructure of today make use of were designed without security in mind. The lack of a fundamentally secure infrastructure makes system and network defense much more difficult. Moreover, large parts of the Internet are dynamic network environments, built up of ad-hoc networks, wireless access points, and so on. The explosive growth of such technology during the last few years has contributed to the constant shift of the network topology. The security mechanisms available in the Internet of today may therefore be highly variable.

Due to the global interconnection of systems, an adversary do not need to be physically present to compromise the security of a system. Attacks can easily be carried out from a distance, and are often easy and quick to perform but hard to detect and trace. The wide-spread use of exploit tools makes it possible, also for novice crackers, to search for and find possible targets on-line. System administrators face a dilemma when striving to maximize the availability of services to authorized users while simultaneously minimizing the opportunities for adversaries to exploit the system.

Trustworthy Systems

The present security tools and methodologies are only adequate for securing systems on a small scale [GGK+03]. For example, cryptography is one of the most well-studied and rigorously modeled aspect in the security field. Still, cryptography alone is not sufficient to secure a system. Most security breaches are caused by faulty software that can be exploited by for example buffer overflows, which unfortunately cannot be avoided by cryptographic techniques. As a consequence, 100% security is very difficult, if not impossible, to achieve. To rely on the service that a networked system provides, the user needs to know to what extent it can be trusted. More specifically, there is an urgent need for modeling methods that can be used to analyze and evaluate the trustworthiness of the system. Today there exists several methods for assessing the qualitative security level of a system, one of the most well-known being the Common Criteria [ISO99]. However, even though such methods give an indication of the quality of the security achieved during design and implementation they do not say anything about how the system will actually behave when operating in a particular threat environment. To be able to measure security, a new approach for quantitative evaluation is needed.

Security is usually defined in terms of the attributes confidentiality, integrity and availability [ISO05], often referred to as CIA. Sometimes additional aspects are considered, such as authentication, access control and nonrepudiation [Sta03]. Dependability, on the other hand, is the ability to deliver services than can justifiably be trusted [ALRL04]. This field has a rich tradition of system evaluation models, which can be used to assess and predict the current and future system behavior when considering random failures. Unfortunately, malicious behavior is rarely considered as a possible fault source in these models. In order to be trustworthy, a system needs to be both dependable *and* secure. These two distinguished research fields share many

similarities but are also fundamentally different in several aspects, which probably is the main reason why they (so far) have tended to be treated in separate frameworks and evaluated by models developed by separate research communities.

Based on the above reasoning, the overall motivation for the research presented in this thesis can be identified by these three questions

1 How can security be quantified and measured?

2 What is the relation between security and dependability?

3 Are there methods that can be used to evaluate a system's trustworthiness, in terms of its security *and* dependability behavior?

As will be seen in the subsequent sections, even though there exist promising research results related to the first two questions, not much effort has been put in the third one. The purpose of the research presented in this thesis is therefore to search for, and hopefully provide, an answer to the third question. This is a long-term goal, which requires novel thinking and redefinition of the old concepts. Even though there may not be a straight-forward solution, the development of new modeling and analysis methods may in itself be an important step towards a future framework where both security and dependability can be quantified and measured.

A Note on Terminology

In some research communities, the term "reliability" is used rather than "dependability" to describe the overall operational characteristic of a system. To avoid confusion, this thesis advocates the use of the terminology suggested by
Avizienis et.al. [ALRL04], which is illustrated in Fig. 1, where dependability is stated as a global concept that encompasses the attributes reliability, availability, safety, integrity and maintainability. Reliability is then defined as "continuity of correct service", which can be viewed as a measure of the achieved system dependability.

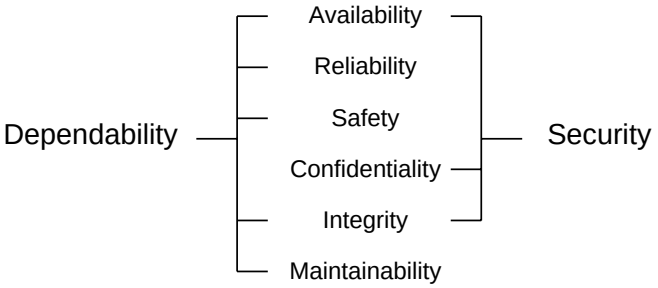


Figure 1. Dependability and security attributes [ALRL04].

2. Thesis Idea

The purpose of this section is to explain the main ideas that the research in this thesis is based on. The text is intended to be readable also for those that are unfamiliar with mathematical modeling and analysis. All obstructing details have therefore been omitted from this section. For more details, the reader is referred to introductory literature on probability models [Ros03], dependability evaluation [Hel] and game theory [Sta91, Owe01].

Stochastic Modeling

A common approach to dependability evaluation of computing systems is to use stochastic modeling techniques [Hel]. A stochastic model is a model that involves probabilities, or randomness, associated with time and events. When using such a model, a stochastic process will represent the system behavior. The stochastic model can be depicted as a state transition diagram, which describes all relevant operational system states and the possible transitions between these states. To describe time aspects between events, a rate matrix has to be specified. One usually assumes that the event that will occur next, as well as the time until the next event, is random. Hence, the behavior of the system is a stochastic process. The main advantage of this modeling approach is that it captures the dynamic system behavior, i.e., the sequence and time aspects of events, such as failures and repairs. The stochastic process can then be used as a basis for quantitative analysis of the modeled system. By using mathematical analysis techniques, closed-form solutions may be obtained, which describe how the failure- and repair rates affects the expected system dependability in terms of its reliability, availability and so forth. In many cases, the stochastic modeling approach is the most appropriate system evaluation method when quantitative dependability measures are needed.

As pointed out in Section 1, according to the definition of dependability provided in [ALRL04], dependability comprises several system properties, amongst them also the CIA security attributes. One would therefore expect that security can be modeled and analyzed by the same methodologies as the other dependability properties. However, it turns out that this is not the case[1]. The main reason is that malicious behavior is rarely considered as a possible fault source when evaluating system dependability. This means that the stochastic modeling approach that is so useful when analyzing systems to obtain quantitative measures cannot be applied as it is to evaluate security properties. This thesis aims to overcome this problem by developing a methodology that makes it possible to incorporate attacker behavior into the transition rates of a stochastic model, so that a comprehensive system evaluation can be performed.

[1]The exception is *availability*, an attribute that is included in both the security and the dependability taxonomy.

The Failure Process

In a stochastic model, one usually separates between good system states and failed system states. To model the failure process in a security context, we note that the "fault-error-failure" pathology used for dependability analysis, share some similarities with the security domain. By definition, the fault-error-failure process is a sequence of events. A *fault* is an atomic phenomenon, that can be either internal or external, which causes an *error* in the system. An error is a deviation from the correct state of the system. An error is always internal and will not be visible from the outside of the system. Even though a system is erroneous it may still manage to deliever its intended services. An error may lead to a *failure* of the system. In a dependability context, a failure is an event that causes the delivered service to deviate from the correct service, as described in the system's functional specification. Using a similar line of reasoning, a security failure is then an event that causes a system service to deviate from its security requirements, as specified in, e.g., a security policy. Given that also a system's security behavior can be represented as either good states or failed states, one can then use the stochastic process to compute measures such as the system's expected time to next (security) failure.

There are a number of different ways to prevent failures. The taxonomy in [ALRL04] divides these into four categories: fault prevention, fault tolerance, fault removal and fault forecasting. This thesis concentrates on the last aspect, fault forecasting, which means to evaluate the system behavior with respect to future fault occurrence or activation[2]. Modeling and analysis of a system for predictive purposes can be performed by static or dynamic methods. Examples of static models are fault trees and reliability block diagrams. The use of stochastic models is a dynamic method, which provides probabilistic system measures, such as its mean time spent in the good states, or mean time to failure as previously discussed. To facilitate analytical analysis of the model, all transition rates are assumed to be exponentially distributed in this thesis. The validity of this assumption will be further discussed in Section 8.

Modeling Malicious Behavior

Given that a system is represented by a stochastic model, the execution of a transition caused by malicious behavior will henceforth be referred to as an *attack action*. In this thesis it is assumed that a large number of adversaries, i.e., attackers, targeting the system simultaneously. This is a realistic assumption for most of the networked ICT systems of today, which are on-line round the clock. By studying log files one can see that these systems are constantly subject to more or less suspicious activity, such as probing, worm activity or other kinds of vulnerability exploitation. The rate

[2]In fact, what this thesis concentrates on is *failure* forecasting rather than fault forecasting. As pointed out in [MM99], regarding security, there seems to be more interest in predicting failures rather than faults, most likely because most security problems either are due to software bugs, which are extremely difficult to predict, or basic design flaws, which are extremely difficult to repair.

value of a transition in the stochastic model, which represents an attack action, will then model the accumulated failure intensity, given that all attackers will always try to attack the system. Unfortunately, this rate value is in itself not enough to accurately describe the expected time before the transition actually will occur. One of the main reasons is that attacks are not truly random processes. Because attackers act with intent, they are not always well characterized by models of a random nature [Coh99]. For example, assume that the system that is to be evaluated is a small corporate local area network (LAN) consisting of a private fileserver, a publicly accessible webserver and a router connecting the LAN to the Internet. Now assume that the expected time a remote attacker would need to break into and read access restricted files on the fileserver is about the same as the expected time needed to break into and deface the webserver. The latter can be characterized as an integrity failure and the former as a confidentiality failure. However, in practice it may be much more common that webservers get defaced than that fileservers get compromised. In fact, the network administrator of this particular LAN assess the frequency of the former to be five times as high as the latter. When using a stochastic model to evaluate this system, the rate values of these two security failures must represent the actual occurrence rates of the events, rather than the success rates of the individual attack actions.

Attacks that are caused by human beings, and that lead to security failures, are very often highly intentional with the specific aim of causing maximum benefit to the adversary or damage to the system. The basic idea that has been pursued in this thesis is that the probability of an attack will depend on not only the expected time (or effort) required to perform the attack but also on how motivated the particular attacker is. As will be seen, there are a number of factors that drive humans to attack computing system, such as financial gain, curiosity, pure entertainment, a rise of ego, etc. On the other hand, a number of factors may reduce the attacker's motivation and make him refrain from certain attack actions. For example, an employee, with a user account on the corporate LAN discussed above, may put his future career at risk if he tries to abuse his insider privileges to attack the local computer network. The gain from a successful break-in into the fileserver may therefore be smaller than the possible consequences he will experience if the intrusion is detected by the system administrator. As another example, the illegal aspect of actions (criminal offense) may prevent even a remote attacker to use available tools to exploit vulnerabilities in such networks. Even though the expected time or effort to perform an attack action may be randomly distributed, the *decision* to perform the attack will therefore be a trade-off between the gain from a successful attack and the possible consequences of detection.

In this thesis attacker behavior is represented as a probability distribution over all the possible attack actions available in a particular system state. These probabilities are then reflected in the transition rates of the stochastic model by weighting the corresponding (accumulated) attack intensities. For example, if an attacker will choose a particular attack action with probability 0.5, then we can expect 50% of all attackers

to take this action, given that they all share the same motivation. Hence, by introducing attack probabilities as parts of the transition rates, the result from a successful attack can be modeled as one or more intentional state changes of the underlying stochastic process, which represents the dynamic behavior of the system. This is illustrated in Fig. 2 where 1 is a good system state, 2 is a (security) failed system state, a is an attack action, $\lambda_{12}(a)$ is the accumulated attack intensity (given that all attackers always take action a) and $\pi_1(a)$ is the probability of action a in state 1.

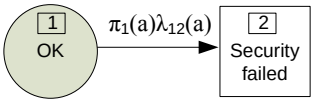


Figure 2. A stochastic model with assigned failure rate (from PAPER D).

The stochastic modeling approach proposed in this thesis aims to be high-level in that it focus on the *impact* of the intrusions on the system rather than on the specific attack procedures themselves. This facilitates the modeling of unknown attacks in terms of generic state transitions. For example, in the stochastic model depicted in Fig. 2 the attack a can simply be explained as "the action that seeks to transfer the system from the good state 1 to the failed state 2".

Predicting the Attack Probabilities

So, how can the attack probabilities be computed? To model an attacker's motivation this thesis make use of a reward- and cost concept. "Reward" is a generic concept, which can be used to quantify the value of an attack action in terms of social status, money, satisfaction, etc, as previously discussed. To model the possible consequences experienced by risk adverse attackers, a negative reward, a "cost", is used to quantify the impact on the attacker whenever an attack action is detected and reacted to. In order to create a generic and sound framework for computing the expected attacker behavior in terms of attack probabilities, this thesis applies *game theory* as the mathematical tool. Each atomic attack action, which may cause a transition of the current system state, is regarded as an action in a game where the attacker's choice of action is based on a consideration of the possible consequences. The interactions between the attacker and the system can then be modelled as a game, as illustrated in Fig. 3. As can be seen, the aspects that can be included in the game are the detection probabilities of attack actions, the operational activities that may affect the current system state, random software- and hardware failures that may occur, and of course the cost- and reward values associated with the available attack actions (not depicted in the figure). In the second part of this thesis, PAPER A-D and PAPER F will pursue these ideas in depth and, by using simple case studies, demonstrate how the proposed approach can be applied for real-world system modeling and evaluation.

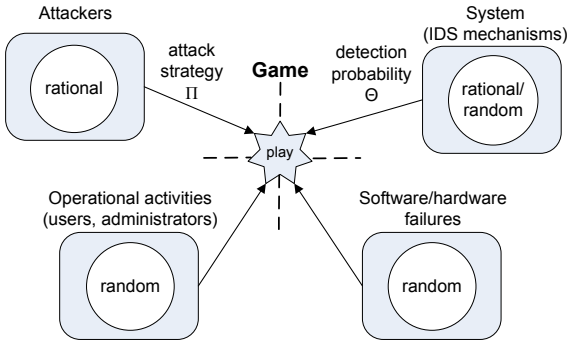


Figure 3. The interactions between an attacker and the system modelled as a game (from PAPER C)

Predicting the System Current and Future Behavior

Given that a system's security behavior can be represented by a stochastic model, another interesting application arises. It turns out that the same model can be used as a basis for risk assessment. "Risk" is usually defined and measured in terms of probabilities and consequences. Suppose that cost values are assigned to the different system states. These are not the same cost parameters as was used in the game theoretic approach, but rather quantitative consequence values, which describe the system administrator's (or any other stakeholder's) loss experienced due to system or service failures. By estimating the current system state probability, the risk of the system can be computed as a function of the failure probabilities and the cost values associated with the failed states. PAPER E and F in Part II of this thesis will demonstrate how the stochastic model can be used as a part of a distributed agent-sensor architecture for real-time risk assessment of the system, and how the proposed agent-sensor architecture can be used to predict the system's future security and dependability behavior. In PAPER E the system measure that is computed is the total system risk at time t, denoted $\mathcal{R}_t$, where risk is defined as the sum of the estimated system state probabilities times the corresponding cost values. This measure reflects the expected cost due to failures, similarly to the output resulting from traditional quantitative risk analysis methods. In PAPER F two new types of measures are used: the probability that the time until next failure is greater than t, denoted $P_F(t)$, and the mean time to next failure ($MTNF$), assuming that the system will sooner or later fail. In contrast to the risk measure used in PAPER E, these measures relate to the expected failure times rather than the possible consequences of failures.

3. Foundation and Related Work

This section presents the previously published research results, which have served as the main inspiration when writing this thesis. The areas that have been emphasized are "stochastic modeling", "security quantification", "attack modeling" and "intru-

sion detection". Relevant research projects are also presented. Note that parts of this research have also been cited in the papers included in Part II.

Stochastic Modeling

Stochastic modeling and analysis techniques have long been used for dependability evaluation by computing failure times of systems when considering accidental fault sources [Buz70, RAS96, Hel]. A common modeling approach is the use of continuous time Markov chains (CTMCs), which are frequently applied due to their strength in describing dynamic behavior and their advantage of obtaining closed form solutions from mathematical analysis. An introduction to the topic is given in [Ros03]. Unfortunately, most of the stochastic modeling approaches tend to ignore security in that malicious behavior is not considered as a possible failure cause. In [Lap92, ALR00, ALRL04] Avizienis et.al. provide a thorough definition of the fundamental concepts of dependability. Here, dependability is used as an umbrella concept and security is treated as an attribute in line with the other attributes reliability, availability and safety. Several other research papers and projects have refined these concepts by discussing how fault prevention, removal, tolerance and forecasting can be reinterpreted in a security related context [Pe01, Mea95, MM99, NST04], and suggest frameworks for integrated security and dependability evaluation [JO92, Jon98, JSL99, MKF03]. Stochastic modeling has also been applied to measure survivability, see e.g., [LT04, MN03, McD05], where survivability usually is defined as "the capability of a system to fulfill its mission, in a timely manner, in the presence of attacks, failures and accidents" [EFL$^+$97].

Stochastic Petri nets (SPN) [Mur89] and Coloured Petri nets (CPN) [Jen97a, Jen97b, Jen97c] are modeling methods commonly used for stochastic dependability and performance analysis (see e.g., [MT95, KBMP99]). One advantage of these models over the traditional Markov models is the possibility of showing explicit the conditions for further events to take place. There are several software tools available for solving Petri net models; the most well-known being UltraSAN [SOQW95] for SPN and CPN/Tools [BLMJ$^+$01, CPN06] for CPN. In a few cases, also the security aspects of system have been modelled by means of Petri net models [SCS03, WMT03, HS05, GLR$^+$03].

Security Quantification

Quantifiable security is a topic that has gained a lot of interest in the research community during the last decade. The modeling approach used in this thesis is based on ideas initially presented in the groundbreaking paper by Littlewoord et.al. [LBF$^+$93], published in 1993. To our knowledge, the authors of this paper are the first to point out that the security measures of a system should be *operational*, which means that such measures should reflect the system's ability to remain secure under particular conditions of operation, including attacks. By relating the security concept to the reliability domain, the authors suggest a new approach to security evaluation based on

an analogy between system failure and security breach. By introducing the random "effort to breach" variable [BLOJ94], a security function, which corresponds to the reliability function used in traditional dependability analysis, can be defined. The proposed approach opens up for new types of quantitative measures for a system, such as its mean effort to security breach. Based on [LBF+93, BLOJ94], a number of other research groups have developed models for quantitative measures of security. An outstanding example is the work by Ortalo et.al. [ODK99]. Relating the fault tolerance concept to intrusion tolerance, the same concept has also been applied by using either probabilistic or stochastic models that captures attacker behavior and system response mechanisms to perform quantitative security analysis of intrusion tolerant systems [GPWW+01, SCS03, MVT02, WMT03, BBMGPT04, SCS+04]. Another interesting approach, which is closely related to security quantification, is the trust metrics for authentication presented in [RS97].

Risk assessment can also be used to compute metrics for security quantification. As discussed in Section 2, by measuring risk in terms of probabilities and costs of attacks (or other unwanted incidents), the security of a system can be quantified. Relevant standards for security metrics and risk assessment are the recently published NIST 800-55 "Security Metrics Guide for Information Technology Systems" [NIoST03], as well as the upcoming standards ISO 27004 "Information Security Management Metrics and Measurement" [ISOb] and ISO 27005 "Guidelines for Information Security Risk Management" [ISOa]. A comparable standardized framework is the AZ/NZS 4360 "2004 Risk Management" [Sta04]. Risk-based security engineering [SDE+04, SJ05], where risk analysis is integrated with classical system engineering, is another approach to security quantification. Security metrics for risk assessment are also discussed in e.g., [Sah05]. A model for assessing the risk of using vulnerable system components is suggested in [BMS05]. Risk has traditionally been interpreted as a static concept. Lately, real-time risk assessment has gained some interest in the security community. A notable example is presented in [GK04], which introduces a formal model for the real time characterization of risk faced by a host.

Attack graphs (or attack trees) [Sch99, PS98, JSW02b, JSW02a, SHJ+02, AWK02] provide a formal and methodical way of describing the security of systems, similarly to how fault trees are used to describe dependability. An attack graph is a structure that represents the set of actions that an attacker can take to achieve a predefined goal. By applying traditional graph based analysis on attack graphs, optimal security countermeasures can be identified and system measures can be computed [JSW02a]. Another interesting approach to security quantification is the definition and analysis of a system's "attack surface" [HPW03, MW04, MW05], which is defined as the set of ways an attacker can attack the system. By identifying the resources that can be used to attack the system, the system security can be measured in terms of an attack surface metric, which indicates the level of damage that may be caused, together with the effort required to cause this damage.

Finally, Quality of Service (QoS) architectures that comprise security have been discussed in a number of research papers, e.g., by Lindskog and Jonsson [LJ02, Lin05]. To be able to include security as a part of QoS, quantification is necessary. A promising approach is the tunable encryption services introduced in [LSHJ04, LB05, LLBFH05].

Attack Modeling

In order to obtain quantitative measures of security, the process of attack modeling and prediction will be a crucial part. To produce measures, all of the models discussed above first need to be parameterized with attack data, which can be either probabilities of different attack actions, failure rates, or other kinds of statistics, depending on the particular modeling approach. A well-cited paper is the work of Jonsson and Olovsson [JO97], which presents a quantitative model of the intrusion process. Based on empirical data collected from experiments performed by students in a controlled environment, this paper demonstrates that a typical intrusion process can be viewed as three different phases; a learning phase, a standard attack phase and an innovative attack phase. The data collected during the standard attack phase indicates that the time to break into a system is exponentially distributed, which has been one of the underlying assumptions for the stochastic modeling approach applied in several of the previously published papers on security quantification (as well as the papers in this thesis) to be valid. Another interesting paper is the model to forecast security breach rates presented in [Sch05].

A honeynet [The06, Pro04] is an architecture that has been developed in order to to learn about security threats and to obtain empirical data from real-life attacks. The main purpose of a honeynet is to gather information. It provides real systems and applications for an attacker to interact with, which makes it possible to detect and terminate botnets, capture and analyze malware for anti-virus, and so on. In many cases the honeynet simply function as a testbed for studying and learning about attacker behavior. Several ongoing projects aim to collect data from a number of different sources (for example honeynets) in order to predict attacks, such as [ADD$^+$05, Aro]. There also exist more theoretical studies that aim to classify attackers, which can be applied in order to facility attack modeling and prediction. For example, a taxonomy that has turned out to be very useful for the attack modeling used in the papers included in this thesis is presented in [Hac05]. In [Ins04, CINU05], a specific type of threat, the insider attacker, is studied.

Game theory has frequently been used to predict human behavior in areas such as economics and social science. Recently, game theory has gained interest also amongst researchers in the security community as a means to model the interactions between an attacker and a system. It can be used both as a method to predict attacker behavior and to analyze and facilitate the decision process and intrusion response strategy during ongoing attacks. Examples are [LZ03, AB03, AB04, LW02, LW05]. Another useful application of game theory is for trade-off analysis of security coun-

termeasures and to evaluate security investments [But02, CMR04] before system implementation. Good introductions to the topic of game theory are the books by Stahl [Sta91] and Gibbons [Gib92]. Gambit [MT04] is a software tool that can be used to construct and solve finite extensive and strategic games. An interesting up-to-date discussion of the Nash equilibrium solution of a game, and its applications, is provided in [HR04]. Some of the papers in this thesis make use of *stochastic* game models, based on the theory in [Sha53, Owe01]. Algorithms for solving these games are provided in [Som04].

Intrusion Detection

Intrusion detection systems (IDS) are systems designed to identify misuse or unauthorized use by authorized users or external adversaries [MHL94, NN99]. Comprehensive surveys of IDS are found in e.g., [ACF+00, Lun88]. An IDS can be either signature (pattern) based or statistical anomaly detection based. The former has an advantage in its low false alarm rates but can only detect already known attacks, whereas the latter are required to have full knowledge of the normal behavior of the system in order to detect all attacks. Markov models for statistical anomaly detection in IDS architectures are presented in e.g., [JTM01]. STAT [SEK02] is a state-based attack description language for intrusion detection, developed at the University of California in Santa Barbara. Distributed IDS have been demonstrated in several prototypes and research papers, such as [SCCC+96, SBD+91]. An important development in distributed intrusion detection is the recent IDMEF (Intrusion Detection Message Exchange Format) IETF Internet draft [DCF05]. Multiagent systems for intrusion detection, an approach where several independent entities (autonomous agents) collaborate to facilitate distributed IDS, are proposed in [CS95, BGFI+98] and demonstrated in e.g., [HWH+03].

Hidden Markov models (HMMs) have recently been introduced as a part of IDS architectures to detect multi-stage attacks [OMSH03], and as a tool to detect misuse based on operating system calls [WFP99]. A very well-written tutorial on HMMs and their application on speech recognition is provided by Rabiner [Rab90]. A more comprehensive treatment of the HMM topic is the book by Cappé et. al. [CMR05].

Research Projects

There are several Europeian research projects related to the area of security and dependability. The Information Society Technologies (IST)[3] has sponsored a number of relevant projects, such as the MAFTIA project from 2003 [MAF], the recently closed Beyond-the-Horizon project [Bey] and the ongoing EuroNGI Workpack 6.3 on trust creation [Eur]. Another relevant project is the European Workshop on Industrial Computer Systems Reliability, Safety and Security (EWICS) [EWI]. On an international basis, the IFIP Working Group 10.4 [IFI] concentrates on understand-

[3] See http://cordis.europa.eu/ist/

ing and exposition of the fundamental concepts of dependable computing, including security. This working group organizes and sponsors, amongst others, the IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)[4] and cooperates with e.g., EWICS.

At first sight it may seem like the background material discussed in this section is both divergent and incoherent. However, as will be seen, the results presented in the papers included in Part II of this thesis span all these areas.

4. Research Methodology

The research presented in this thesis has been performed at the Centre for Quantifiable Quality of Service (Q2S), Centre of Excellence (CoE) at the Norwegian University of Science and Technology in Trondheim. As stated in the Centre vision[5]

> *"The Centre will study principles, derive mechanisms, methods and technical solutions and assess their properties and performances by means of experiments and models."*

and

> *"The main research goal is to do basic research in the areas specified for the Centre, with a coordinated cross-disciplinary emphasis on QoS."*

much of the research conducted at the Centre is of a fundamental kind. Also the methodology used in this thesis is mainly a theoretical study, rather than an empirical one. The main reason for this approach is the lack of a formal foundation in the particular area of combined security and dependability evaluation. The major effort has therefore been put into the development of a novel modeling method, which aims to bridge this gap. By studying the stochastic models used for traditional dependability analysis, as well as game theoretic models for predicting human behavior, a method for evaluating the trustworthiness of a system (in terms of its security *and* dependability) has been developed. The characteristics of the proposed models in this thesis have been demonstrated by mathematical analysis.

The CTMC Approach

To be able to find a method that can be used to evaluate both a system's security and dependability behavior (see question 3 in Section 1), the concepts and methodologies for traditional dependability evaluation have been surveyed. The chosen modeling method applied in this thesis is the use of stochastic processes, or more specifically: continuous time Markov chains (CTMCs). Such models have been proved suitable for representing the behavior of computing system, in terms of (accidental) failures and repairs. As pointed out in [Hel], stochastic models are particularly useful for

[4] See http://www.dsn.org/
[5] See http://www.q2s.ntnu.no/

capturing dynamic system behavior and time intervals between failures. Among the advantages of the CTMC approach is the possibility of obtaining closed form solutions when performing system analysis. However, it is an idealized model that requires an heavy abstraction of the true system behavior. The validity of this and other assumptions will be further discussed in the next section.

The Published Results

The thesis is based on the result presented in six papers, which have been presented and published at international conferences and workshops during 2004-2006. The grand part of the work in five of the papers (PAPER A-D and F) has been performed by the thesis author, under supervision of Professor Svein J. Knapskog and Professor Bjarne E. Helvik. It should be emphasized that Professor Helvik was the one who first suggested the use of game theory as a tool to model and compute attacker behavior, an idea for which the thesis author is very grateful. The game theoretic approach has then been pursued in depth by the thesis author, with helpful and highly valuable feedback from both Professor Helvik and Professor Knapskog. The remaining paper (PAPER E) has multiple authors, several of them PhD candidates. The major part of this work has been compiled by the thesis author, together with Ph.D candidates André Årnes and Kjetil Haslum. Since this paper is comprised from a joint effort, it is not straight-forward to identify the contributions of a single author. The cooperation was initiated by André during spring 2005. André was the one who proposed the initial idea of using a distributed agent-sensor architecture for monitoring and assessing network risk. During the summer of 2005, Kjetil and the thesis author wrote down and formalized the idea of using a HMM as a tool for agents to interpret the data collected from multiple sensors. André and the thesis author then compiled a case study and wrote down and submitted the paper, which was accepted for publication later on this year. Also this work was supervised by Professor Knapskog.

5. Research Assumptions

The models and methods applied in the thesis relies on three main assumptions, which need to be highlighted. Note that future work aimed at finding methods to deal with these assumptions will be discussed in Section 9.

The Markov Property

To facilitate mathematical analysis of the stochastic models used in the papers in Part II of this thesis, Markov processes were used to model the transitions between the possible states of a system. The Markov assumption implies that the transition probabilities between system states depend only on the current system state, and not on any of the previously visited states. All the system state holding times are assumed to be negatively exponentially distributed in the examples provided in the papers. Even though these papers use CTMCs, it is not a necessity for the stochastic modeling approach to be valid. In cases where non-Markovian stochastic processes are more suitable, simulation can be used to predict system measures.

The Game Models

In PAPER A-D game theoretic models are applied to compute the expected attack probabilities. When using a game to predict attacker behavior two important assumptions have to be made. First, one assumes that the attackers are rational, which means that they want to gain as much as possible in accordance to the specified reward- and cost values. Second, it is assumed that the attackers always have a complete overview over the parameters of the ongoing game, i.e., the vulnerable system states and all the possible attack actions and their possible consequences. The optimal solution to the game is then used as a prediction of how the attackers will behave.

Independent Observations

The HMM approach applied in PAPER E and F of the thesis relies on the assumption that subsequent observations produced by the network sensors monitoring the system are independent. Or in other words: the probability of an observation at a particular time instant t is determined of the current system state only, and not on any previously visited states or any previously received observations. However, in practice the behavior of some types of IDS (for example Snort [Sno02], which is based on misuse detection) is deterministic in that the IDS will always provide the same observation in a particular situation. Consequently, specific types of repeatedly attack incidents, such as probing or worm attacks, might not be well described by the HMM approach.

6. Summary of the Papers

This section summarizes the main contributions of the papers and discusses how the content of each paper relates to the other papers.

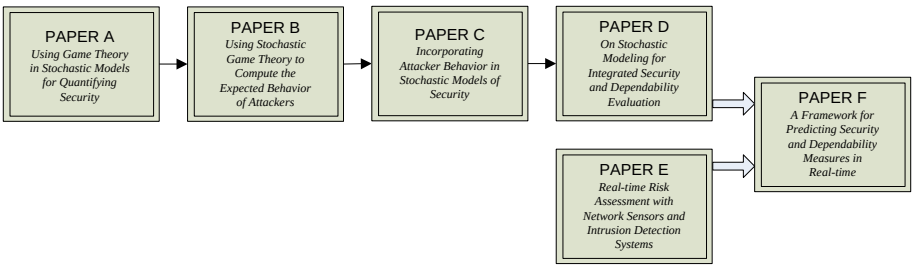


Figure 4. The relation of the papers included in Part II of the thesis.

The mutual relation of the six papers included in Part II of this thesis is depicted in Fig. 4. As indicated in the figure, PAPER A-D are closely related. The basic research idea, which is introduced and further developed in these papers, is the use of game theory to determine the transition rates for stochastic models of security and dependability. The initial, very simple, model presented in PAPER A is gradually refined in the subsequent three papers. PAPER D contains the most recent research results

in this particular field. The context in these four papers do overlap in more than one aspect. As the model gradually became more complex, the use of the parameters had to be carefully reconsidered. The notation has therefore been refined during the work with the model. As can be seen when reading Part II of this thesis, also the applied terminology has evolved during the work with these papers.

PAPER E presents a novel approach to real-time risk assessment of network systems. The method suggested in this paper is, in itself, detached from the research results in PAPER A-D. However, PAPER F connects the stochastic modeling approach used in PAPER A-D with the basic monitoring architecture in PAPER E in a unique way.

PAPER A

Using Game Theory in Stochastic Models for Quantifying Security

In the first paper, game theory is suggested as a method for modelling and computing the probabilities of expected behavior of attackers in a quantitative stochastic model of security. The stochastic model presented here is very simple, modeling a penetration attempt as a series of intentional state changes that lead an ICT system from an assumed secure state to a state where one or more of the systems security aspects are compromised. The attack transition rates consists of rate values, which are multiplied by attack probabilities, as discussed in Section 2. In this paper, the game situation models the actions of an attacker under the condition that at each intermediate stage of the attack, the attempt may be detected and measures taken by the system owner to bring the system back to the originating secure state. Assumptions are made for the possible rewards for the players of the game, allowing the calculation of the mean time to first security breach (MTFSB) for the system. An example of the possible use of the model is provided by calculating the MTFSB for a root privilege attack on a UNIX system.

PAPER B

Using Stochastic Game Theory to Compute the Expected Behavior of Attackers

This paper refines the initial model in PAPER A by suggesting the use of stochastic game theory, rather than a simple game model, as the mathematical tool for computing the expected behavior of attackers. The possible use of the Nash equilibrium as a part of the transition probabilities in stochastic models is defined and motivated. To demonstrate the approach, a simple example of an attack against a computer network is modeled and analyzed.

PAPER C

Incorporating Attacker Behavior in Stochastic Models of Security

The third paper continues where PAPER B ends, by elaborating the method for computing expected attacker behavior for use in stochastic models of security. The paper includes the time aspect in the success probabilities of attack actions; an important aspect that was neglected in both PAPER A and B. Furthermore, it is demonstrated how the same game model can be used in three different threat environments, modeling different types of attackers. As in its predecessors, in order to illustrate the approach, the paper provides a small case study.

PAPER D

On Stochastic Modeling for Integrated Security and Dependability Evaluation

In this paper, the relation between dependability and security is discussed and the need for an integrated evaluation framework is pointed out. The paper suggests the use of stochastic modeling techniques as a suitable method for assessing the trustworthiness of a system, regardless of if the failure cause is intentional or not. This paper provides a more thorough background to the results previously presented in PAPER A-C and gives further motivation on why and how attacker behavior can be incorporated in the transition rates of a stochastic model. Here, the basic formulation of the game model used in PAPER A-C is generalized to consist of $n \times n$ game elements, which means that the interactions between an attacker and the system IDS mechanisms can be modelled in a more realistic way. This paper also includes a detailed evaluation of how the reward- and cost parameter will influence the expected attacker behavior. To illustrate the results of applying the model in a real-world context, a fairly detailed example is provided. The paper is an extended journal version of the results previously published by the thesis author in [SHK06c] and [SHK06a].

PAPER E

Real-time Risk Assessment with Network Sensors and Intrusion Detection Systems

This paper describes how a stochastic modeling approach can be used to perform real-time risk assessment of large networks. Similarly to PAPER A-D, the basic model in this paper relies on the assumption that the security of a system can be modelled as a finite number of states. By associating each state with a monetary cost value, the paper then proposes that the current risk of the system can be quantified in terms of state probabilities and costs. The basic model is intended to be implemented in a distributed agent- and sensor architecture, tailored for monitoring of large networks. The main task of the sensors is to provide the agents with observations regarding the security state of one or more systems that are under observation. Based on hidden Markov model theory, the paper provides a mechanism for handling and interpreting data from sensors with different trustworthiness, which makes it possible to compute the most likely state probability distributions for the observed systems, and thereby compute the current risk value for the network, in real-time.

PAPER F

A Framework for Predicting Security and Dependability Measures in Real-time

This paper presents a framework for implementing the stochastic modeling approach in PAPER A-D in the distributed agent-sensor monitoring architecture proposed in PAPER E. In this paper, two new probabilistic system measures are defined: a system's mean time to next failure ($MTNF$), and the probability that the system remains free from failures until a certain time instant ($P_F(t)$). These measures provide a new way of dynamically measuring a system's trustworthiness, in terms of its security and dependability behavior. The purpose of the framework is to facilitate the computation of the system measures, in real-time. By using the observations provided by the network sensors, the probabilities of the current system states can be estimated, which makes it possible to use the stochastic model to predict the current and future behavior of the monitored system. To demonstrate the approach, an illustrative example is included.

7. Guidelines for Reading

The purpose of this section is to explain the contents of the different parts of this thesis, how they are related to each other, and to suggest which parts that should be read in which order by readers with different backgrounds.

Part I - Thesis Introduction

The first part of the thesis explains the background and motivation to the research topics that have been pursued, discusses related work and gives some indications of how the obtained research results can be extended into future work. This part of the thesis is intended to be read as an introduction to the rest of the thesis.

Part II - Included Papers

The second part consists of six published papers, which comprises the main part of this thesis. All the papers in this part are self-contained and can therefore be read in any sequence. However, as indicated in Fig. 4, because the papers are related to each other their contents do overlap in some respect. Especially the contents of PAPER A-D are closely related, in that the model originally presented in the first paper is gradually refined in the subsequent three papers. The last paper (PAPER F) ties together the results developed in the first four papers with the novel approach presented in the fifth paper (PAPER E). To get a better understanding of the obtained research results, the reader is therefore encouraged to read the papers in alphabetical order.

Part III - Thesis Appendix

The last part of the thesis contains an appendix, which purpose is to explain a scaling procedure required to implement the algorithms in PAPER E and F. As pointed

out in the HMM tutorial by Rabiner [Rab90], scaling is required when dealing with large observation sequences. This is due to the computation of the forward variable α_t. Since α_t consists of a sum of a large number of multiplied terms, which each are generally significantly less than one, each term of the variables tend to zero exponentially fast as the number of observations in a sequence grows large. For large sequences ($t > 100$) the dynamic range of the computation will exceed the precision range of most machines. The best way to implement the algorithms is therefore to incorporate a scaling procedure. Unfortunately, the equations provided by Rabiner for computing the scaled forward variables [Rab90, Eq. (91)-(92b)] cannot be used in the modeling framework proposed in PAPER E and F, since the purpose of the algorithms in these papers is to compute the estimated state probabilities in real-time, rather than using (historic) observation sequences to re-estimate the model parameters. This appendix explains how the scaling coefficients are used in the framework presented in PAPER E and F and proves that the resulting state probability estimates provided by the scaling procedure are correct.

A Note on Notation

The simple model for attack prediction and security quantification that was introduced in PAPER A has gradually been refined to the much more comprising and complex modeling approach presented in PAPER D. Even though the first four papers in this thesis are closely related, the reader will notice that the notation has changed during the work with the model. To facilitate the use of additional variables and parameters, which had to be added when extending the model, also the notation had to evolve during the work with the papers. This is the reason why, for example, in PAPER D Greek symbols have replaced some of the variables used in the game model in PAPER A.

8. Summary and Conclusions

More than ten years after the need for quantitative measures of security initially was brought up (see [LBF+93]), there still does not exist any common methodology, which has been widely adopted for security quantification on a system-level basis. The efforts put in developing methods for quantitative security evaluation during the last decade can be viewed as either static or dynamic analysis methods. The static approach focus on aspects such as how the system was built and what types of vulnerabilities it may contain whereas the dynamic methods focus more on how the system is operated and how it is behaves in a certain environment. This thesis strives to follow the latter approach. To describe a system that is yet to be built or to describe an existing system whose vulnerabilities remain unknown, stochastic assumptions are needed [NST04]. By using a stochastic modeling approach, a system's inherent random behavior due to the introduction and removal of vulnerabilities, attacker behavior, normal user behavior and administrative activities as well as accidental hardware- and software failures can be modeled and analyzed. The papers in this thesis present a method for quantitative security and dependability evaluation, which is based on

such stochastic modeling techniques. The purpose of the developed method is to facilitate the process of obtaining measures of a system's trustworthiness, regardless of whether the failure cause is intentional or not. The thesis also demonstrates how the stochastic modeling approach can be used for real-time risk assessment of a system, and suggests how the system's security and dependability behavior can be predicted in real-time.

The Main Contributions

To summarize, the contributions of this thesis are related to three main areas.

- Integrating security and dependability
- Attack modeling and prediction
- Real-time assessment

More specifically, the main contributions of the work are as follows:

C1. It is discussed how the "fault-error-failure" pathology, which traditionally only has been used for system dependability evaluation, can be applied in a security related context.

C2. It is demonstrated how a stochastic modeling approach can be used to compute a system's expected time to failure. The proposed model includes not only accidental failures but also malicious behavior, i.e., attacks.

C3. Game theory is introduced as a tool to predict attacker behavior. By modeling the interactions between an attacker and the system IDS mechanisms as a game, the probabilities of expected attacker behavior can be computed.

C4. The use of attack (decision) probabilities as a part of the transition rates is proposed. By multiplying rate values with attack probabilities, failures due to malicious behavior can be modelled as intentional state changes of a stochastic model.

C5. A new definition of system risk is proposed, which is based on a state-based system modeling approach. Moreover, a method to compute this risk in real-time has been developed.

C6. It is demonstrated how autonomous agents can be used to collect sensor data and how hidden Markov model theory can be used to interpret these observations.

C7. Two new quantitative (combined) security and dependability measures have been defined; the systems mean time to next failure ($MTNF$) and the probability that the time until failure is greater than t, denoted $P_F(t)$.

An illustration of how these specific contributions are related to the three main areas is depicted in Fig. 5. These contributions have been published in ten papers so far: the six papers included in Part II of this thesis, and in four other publications by the thesis author [SHK06c, SHK06a, ÅSHK06, HSK07].

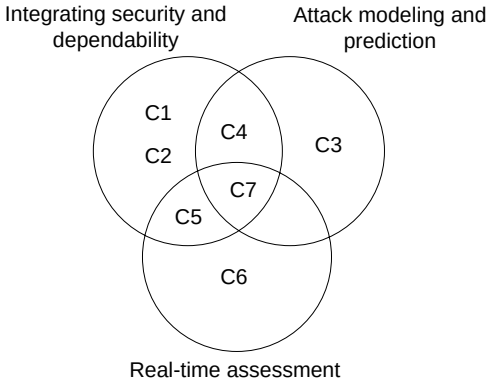


Figure 5. An illustration of how the individual contributions are related to the three main areas.

9. Ongoing and Future Work

This section presents ongoing work that is based on the research results presented in the papers in this thesis. It also point out directions for possible future work.

The Stochastic Modeling Approach

The stochastic modeling approach has been a common theme throughout all of the six papers in Part II of this thesis. Even though the method has been pursued in depth, there is still some work remaining before it can be applied in a real-world context. Here, three main research proposals (P1-P3), which are related to stochastic modeling, have been outlined.

P1: Assessing the attack intensities. To be able to use the stochastic modeling approach to obtain security related measures in practice, the attack intensities in the model needs to be parametrized. Forecasting attack rates poses a great challenge and none of the papers in this thesis enters deeply into this subject. This is therefore very much an open research question. A possible approach is to collect live attack data from, for example, honeynets in order to study time aspects and probabilities of attacks towards existing system vulnerabilities. These data might then be used as input to models for predicting future attack rates. Such models do not exist today and therefore need to be developed.

P2: Petri net modeling. Using a manually constructed Markov chain to capture all relevant architectural details of a real-life system is a difficult task. To avoid state space explosion when dealing with larger or more complex systems, the possibility of using stochastic Petri nets (SPN) as a modeling and analysis tool should be considered. When using SPN the system model will be much more concise, which will facilitate the modeling and analysis process. One additional benefit of this approach is that also other types of distributions, than just the negatively exponential

one, can be used to describe the transition rates between system states. Petri net models can easily be evaluated by means of software tools, such as UltraSAN [SOQW95]. If a numerical solution cannot be found, discrete-event simulations can be performed to obtain system measures.

P3: Verifying the game solutions. As discussed in the previous section, a simple game model that assumes perfect knowledge of the ongoing game may not always be suitable to analyze attacker behavior. It should be investigated whether other types of game models, such as Bayesian games or games with incomplete information [Har68], are more appropriate and, in that case, how and when to use these games. Anyhow, verifying the game model's ability to predict real-life attacks will require further research, including validation of the model against empirical data.

The Hidden Markov Modeling Approach

The HMM approach has been applied in PAPER E and F. However, the results presented in PAPER E have been significantly extended after the publication of this paper. As pointed out in PAPER E, HMMs are discrete-time models, inherently not suitable for interpreting sensor data in real-life applications, whose observations tend to arrive either in burst or sparsely distributed in time. This is discussed in [ÅSHK06], where the method is adapted to approximate continuous time system behavior. Also design issues, such as queuing of sensor observations is addressed in this paper. The paper also presents a discrete-event simulator, which has been used to perform simulation experiments demonstrating the risk assessment process during longer periods of time. A multi-sensor architecture is suggested in [HÅ06]. However, there is still some work remaining before the proposed model can be implemented in a real-world setting. Here, two main research proposals (P4-P5) have been outlined.

P4: Testing the system with live data. The experimental results presented in [ÅSHK06] are based on simulated state transitions and observations generated from a HMM. To obtain more realistic results it is of interest to also simulate the real-time risk assessment process based on real-life data. In [ÅVVK06] the basic risk assessment model is implemented in an existing IDS architecture at the University of California at Santa Barbara (UCSB). By running the proposed algorithms on real-life sensor data this paper demonstrates that the proposed model do indeed produce valuable results. The results indicate that, if the sensors are capable of detecting ongoing suspicious activity, the assessed risk level will reflect the true risk level with a high certainty. These experiments were conducted in an off-line mode. It still remains to test the system on-line with live traffic.

P5: New risk measures. Performability, originally introduced in [Mey89], is a term that encompasses both a system's dependability and performance. Performability modeling, where a performance level is associated with a state of the structural system model, can be used to quantify a system's ability to perform, by assigning reward values to the different system states. The definition of real-time risk in PAPER E

is in fact equivalent to the definition of "point performability" (see e.g., [dSeSG92]), which describes the system's performability for a single point in time. Apart from point measures, also interval-, transient-, and steady-state measures can be obtained by performability modeling. A possible future direction of the proposed risk model in PAPER E could therefore be to define similar time-related measures for risk. By computing not only the real-time risk, but also e.g., the cumulative risk of the system over a certain period of time, a more complete view of the ongoing security incidents in the monitored system can be obtained. It could also be of interest to assign cost values to transitions, rather than to states, so that also the risk incurred in maintaining and restoring the system can be assessed.

The Long-term Perspective

As can be seen, all these aspects (P1-P5) are very closely related to the research results obtained in this thesis. In fact, all five proposals can be viewed as technical improvements of the results presented in PAPER A-F in the subsequent part of the thesis. In a more long-term perspective, other aspects will be important for future research. As an example, for the proposed method for combined security and dependability evaluation to be useful in practice, it needs to be implemented in such a way so that also security analysts without stochastic modeling expertise can use the methodology.

II

INCLUDED PAPERS

PAPER A

Using Game Theory in Stochastic Models for Quantifying Security

Karin Sallhammar and Svein J. Knapskog

In Proceedings of the 9th Nordic Workshop on Secure IT-systems (NordSec 2004)

Espoo, Finland, November 4-5, 2004

USING GAME THEORY IN STOCHASTIC MODELS FOR QUANTIFYING SECURITY

Karin Sallhammar and Svein J. Knapskog
Centre for Quantifiable Quality of Service in Communication Systems
Norwegian University of Science and Technology,
Trondheim, Norway
{ sallhamm, knapskog } @q2s.ntnu.no

Abstract In this paper, game theory is suggested as a method for modeling and computing the probabilities of expected behavior of attackers in a quantitative stochastic model of security. The stochastic model presented her is very simple, modeling a penetration attempt as a series of intentional state changes that lead an ICT system from an assumed secure state to a state where one or more of the systems security aspects are compromised. The game situation models the actions of the attacker under the condition that at each intermediate stage of the attack, the attempt may be detected and measures taken by the system owner to bring the system back to the originating secure state. Assumptions are made for the possible rewards for the players of the game, allowing the calculation of the mean time to first security breach (MTFSB) for the system. An example of the possible use of the model is provided by calculating the MTFSB for a root privilege attack on a UNIX system.

Keywords: Computer security, quantification, stochastic analysis, attack models, game theory

1. Introduction

The security of operating computer systems has traditionally only been expressed in a qualitative manner. However, to be able to offer a security dimension to QoS architectures, it is important to find quantitative measures of security. In the dependability community, methods for quantifying reliability, availability and safety, are well-known and effective. By using state space modeling methods, operational measures for the system, such as the mean time between failures (MTBF), the mean time to failure (MTTF) or the mean time to first failure (MTFF), can be computed. During the past decade, some research on applying the dependability paradigm to security has been performed, using an analogy between system failure and security breach, aiming and attempting to quantify security by calculating measures such as the mean time to security compromise.

However, in contrast to failures, attacks may not always be well characterized by models of a random nature. Most attackers will act with an intent and consider the

possible consequences: satisfaction, profit and status versus effort and risk of the actions before they act. This paper uses a game theoretic approach to model the expected behavior of attackers for use in stochastic modeling techniques.

The gain of using a game theoretic approach in a security related context is twofold. First, we believe it can provide a more accurate model of the attackers' expected behavior, which can be used to assign more realistic transitions probabilities in the stochastic models. Second, it may also help administrators to find the optimal defense strategies of a system and to calculate the expected loss associated with different defense strategies. This work is a demonstration of the former application.

In order to keep the focus on the game theoretic model, issues relating to model parametrization is ignored. Therefore only guessed values are used to demonstrate how the model can be used to obtain quantitative measures.

2. Related Work

There are several papers on quantification of security. In [7], a first step towards operational measures of computer security is discussed. The authors point to the lack of quantitative measures for determining operational security and relate security assessment to the dependability domain. Quantitative measures, such as the mean effort to security breach (MESB), are defined and discussed. [12] presents a quantitative model to measure known Unix security vulnerabilities using a privilege graph, which is transformed into a Markov chain. The model allows for the characterization of operational security expressed as the mean effort to security failure, as proposed by [7]. Further, in [10, 16, 2] traditional stochastic modeling techniques are used to capture attacker behavior and the system's response to attacks and intrusions. A quantitative security analysis is carried out for the steady state behavior of the system.

Game theory in a security related context has also been utilized in previous papers. In [1], a model for attacker and intrusion detection system (IDS) behavior within a two-person, nonzero-sum, non cooperative game framework is suggested. The possible use of game theory for development of decision and control algorithms is investigated. In [9], a game theoretic method for analyzing the security of computer networks is presented. The interactions between an attacker and the administrator are modeled as a two-player stochastic game, for which best-response strategies (Nash Equilibrium) are computed. In [8] a preliminary framework for modeling attacker intent, objectives and strategies (AIOS) is presented. To infer AIOS a game theoretic approach is used and models for different threat environments are suggested.

Based on the game theoretic work of [8, 1, 9], a method to model and compute the probabilities of malicious user actions for use in stochastic models is suggested. To demonstrate how to use the method, a real-world example of an attack against a system is modeled, the optimal strategy of the attack is calculated and, following the approach of [10, 2, 12], the quantitative measure mean time to first security breach (MTFSB) is obtained for the system.

3. The Stochastic Model

Analogously to dependability analysis where system failure is a concept denoting the system's inability to deliver its services, in the security community one often talks of *security breach*; a state where the system deviates from its security requirements. A security breach might accidentally be caused by normal usage operation, but more likely by *intentional attacks* upon the system. Such attacks on an operating computer system can often be modeled as a series of state changes of the system that lead from an initial secure state to one or more target compromised states, i.e., security breach states. A successful attack against the system may therefore consist of many subsequent elementary attack actions. At each intermediate stage of the attack, the attacker will therefore have the choice of either

- *Attack* by performing the next elementary step in the attack.
  - If the attacker succeeds the system will be transferred from state i to state $i + 1$.
  - If the attacker fails the system will remain (temporary) in state i.

  or
- *Resign* and interrupt the ongoing attack.
  - The system will be remain (temporary) in state i.

On the other hand, at each intermediate stage, the system administrator may

- *Detect* the attack and bring the system back to a secure state.
  - The system will be transferred from state i to state 0, hence, the attacker will not have the possibility of continuing the attack.

This is illustrated in Fig. 1. In the model it is assumed that once an attack is initiated, the attacker will never voluntarily try to revert the system to any of the previous states. The model also assumes there is only one single path to the security breach state; a somewhat simplified view of reality.

3.1 Sojourn Time

Since the state transition model presented in Fig. 1 is stochastic by nature, the calender time spent in each state of the system model will be a random variable. The time or effort taken for an attacker to cause a transition will depend on several factors, such as the attacker's knowledge and background, robustness of the system etc. See e.g., [3] for a thorough discussion on this topic and [6] for empirical data collected from intrusion experiments. As mentioned in the introduction, to keep the focus on how to apply game theory in stochastic models, the differences between time and effort and the problems regarding finding suitable distributions and parameters for the model will be ignored in this paper.

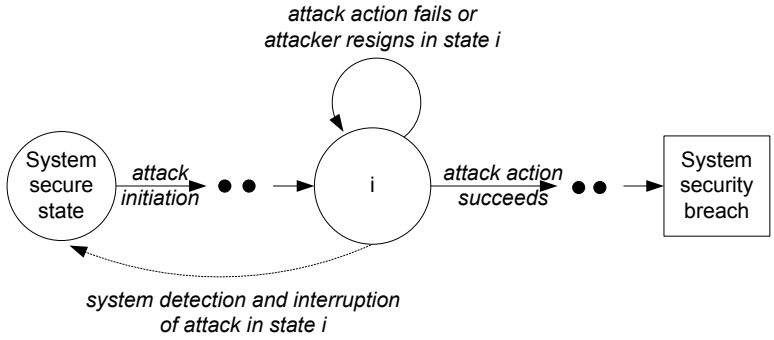


Figure 1. Penetration of a computer system modeled as a series of state changes

For simplification, we therefore model the time between two subsequent attacks are initiated against the system as an negatively exponentially distributed (n.e.d.) variable with parameter λ_{attack}, i.e.,

$$P(t) = 1 - exp(-\lambda_{attack}t). \tag{1}$$

Once an attack is initiated, the initial secure system will be transferred into the subsequent state $i = 1$. As an attack has been initiated, the time needed for the attacker to perform the next elementary step of the attack when the system is in state i, and the corresponding time needed for the system to detect and interrupt the ongoing attack during state i, are also modelled by the n.e.d, i.e.,

$$P_{attack(i)}(t) = 1 - exp(-\lambda_i t), \tag{2}$$

and

$$P_{detect(i)}(t) = 1 - exp(-\mu_i t), \tag{3}$$

respectively. Thus, $\frac{1}{\lambda_i}$ and $\frac{1}{\mu_i}$ will be the respective mean time an attacker and the system spend in state i of the model before causing a transition. The two "competing" processes with rates λ_i and μ_i representing the attacker and system actions in state i can be merged into one Poisson process, hence, due to the memoryless property of the exponential distribution, the state transition model then will be transformed into a continuous time Markov chain (CTMC) [14] with discrete state space, formally described as

$$\{X(t) : t \geq 0\}, X_s = \{0, 1, 2, .., n\}, \tag{4}$$

for which analytic analysis is possible. This model during the given assumptions is displayed in Fig. 2.

In reality, there may be other types of distributions than the negative exponential one, which are more suitable to model the transitions of the stochastic model. However, to facilitate analytic analysis the n.e.d. was chosen for all transitions in the stochastic model.

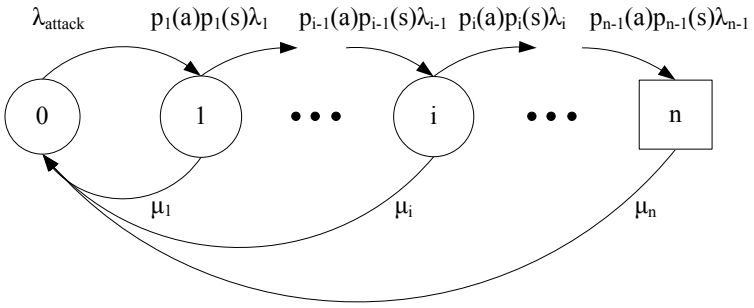


Figure 2. A general stochastic model for penetration of a system.

3.2 Transition Probabilities

As previously mentioned, in each intermediate state i an attacker has two possible choices of action: with probability $p_i(a)$ he will decide to continue the attack and with probability $1 - p_i(a)$ he will resign and interrupt the attack. This *decision probability* represents an important difference between dependability and security analysis when using stochastic modeling techniques. In traditional dependability analysis only accidental failures are modeled; there is no human decision involved in the occurrence of a failure, hence $p_i(a) = 1$, for all i. However, when modeling attacks rather than failures one must keep in mind that an attacker will consider the consequences of his actions and compare the possible payoff versus the risk of each elementary attack action. An attacker may therefore choose to interrupt an ongoing attack at a certain stage or to not start the attack at all. Therefore, for each transition representing an elementary step in the attack $p_i(a)$ should be explicitly calculated.

To be able to bring the system closer to the security breach state, an attacker not only has to decide upon an action, but he must also *succeed* with the particular action. This probability of success is denoted by $p_i(s)$ and is also included in the stochastic model presented in Fig. 2.

Using the attacker's and system's action rates together with the transition probabilities, the *instantaneous* transition rates between the state i and $i + 1$ in the stochastic model can be calculated as

$$q_{i,i+1} = p_i(a)p_i(s) \cdot \lambda_i, \tag{5}$$

and between state i and 0 as

$$v_{i,0} = \mu_i. \tag{6}$$

As will be demonstrated in Section 5, the instantaneous transition rates can be used for quantitative security analysis of the operating system.

4. The Game Model

To determine the decision probabilities $p_i(a)$ game theory can be used. (A formal definition is given in Appendix). If one views each elementary attack action causing a transition in the stochastic model as an action in a game, where the attacker's choices of action is based on intelligent considerations of the possible consequences, then the interactions between the attacker and the system can be modeled as a two-player static game. This is displayed in Fig. 3.

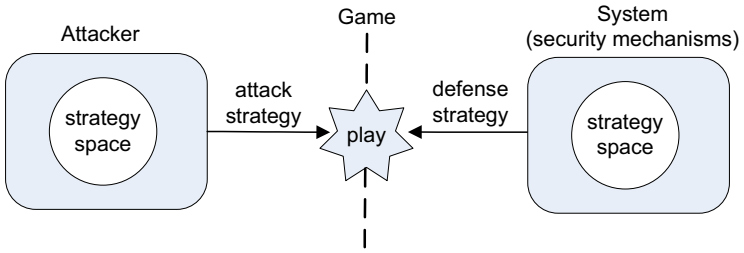


Figure 3. The interactions between an attacker and the system modeled as a two-player static game

The complete action set for the game then includes the attacker's two choices of action together with the possible consequences, i.e., for each state i:

- a_i is the elementary attack action bringing the system from state i to $i+1$,
- r_i is the resignation of the attack in state i,
- d_i represents that the elementary attack action a_i will be detected by the system,
- ϕ_i represents that the elementary attack a_i action will be undetected.

Hence, for each state i there is a game model $G(i)$ defined by

$$\begin{aligned} &N = \{1,2\} = \{\textit{attacker, system}\},\\ &A_i = \{a_i, r_i, d_i, \phi_i\},\\ &u_i = \begin{array}{|c|c|c|}\hline & d_i & \phi_i \\ \hline a_i & u_{i1} & u_{i2} \\ \hline r_i & u_{i3} & u_{i4} \\ \hline \end{array}, \end{aligned} \tag{7}$$

where $u_i = \{u_{i1}, u_{i2}, u_{i3}, u_{i4}\}$ is the payoff received by the attacker, for each possible combination of action and response from the system

For the game defined in (7) it is possible to use (A.2) in Appendix to calculate the attacker's expected payoff for a choice of action as

$$\begin{aligned} u_i(a_i) &= p(d_i)\cdot u_{i1} + p(\phi_i)\cdot u_{i2},\\ u_i(r_i) &= p(d_i)\cdot u_{i3} + p(\phi_i)\cdot u_{i4}. \end{aligned} \tag{8}$$

Since in most cases an attacker *do not know* the exact probability that his action will remain undetected, game theory says he should assume that his opponent (the system) is a conscious player of the game which seeks to minimize the attacker's expected payoff [15], hence, the minimax solution of the particular game $G(i)$ can be calculated as

$$\alpha_i^* = \underset{\alpha_i(d_i)\alpha_i(a_i)}{min\, max}\{u_i\big(\alpha_i(a_i)\big), u_i\big(\alpha_i(r_i)\big)\}, \tag{9}$$

as defined in Appendix. In reality, since the reward experienced by the attacker from an outcome rarely coincide with the system's loss, the game is usually not truly zero-sum. However, defining payoff values for the system is irrelevant in game models like these where the attacker is the only player who is capable of making intelligent decisions.

The minimax solutions $\alpha_i^*(a_i)$ of the games $G(i)$ for all the elementary attack actions $i = 1, .., (n-1)$ represent a complete attack strategy, which has the property that, by following it, an attacker will know that he has maximized his expected payoff of the attack. This gives him a guarantee of the result from the attack regardless of if one of his elementary attack actions will be detected by the system or not; the "no regrets property" of game theory. Several experiments indicate that this search for guarantees is a very strong motivator of human behavior and assuming that the attacker population targeting the system will make rational choices relative to their objectives, the situation will in the long run naturally gravitate towards the minimax solution (i.e., the Nash equilibrium) [4, 15]. The minimax strategy will therefore indicate how *rational* attackers will behave.

The attacker decision probability in state i of the stochastic model can therefore be directly derived from the minimax solution of the corresponding game as

$$p_i(a) = \alpha_i^*(a_i). \tag{10}$$

Note that, when α_i^* is the solution of a *static* game, it is only correct to use (10) for the stochastic model as long as the decision probability $p_i(a)$ depends on the current state i in the stochastic model only (i.e., the Markov property holds).

5. Quantitative Analysis

Returning to the stochastic model presented in Section 3 and illustrated in Fig. 2, since the stochastic model is a homogeneous continuous time Markov chain [14], it is straight-forward to determine the limiting probabilities of each state in the model. By solving the equation system

$$\begin{cases} P_{S0} \cdot \lambda_{attack} = P_{S1} \cdot (q_{12} + v_{10}) \\ P_{S1} \cdot q_{12} = P_{S2} \cdot (q_{23} + v_{20}) \\ \vdots \\ P_{Sn-1} \cdot q_{n-1,n} = P_{Sn} \cdot v_{n0} \\ \sum_{i=0..n} P_{Si} = 1 \end{cases} \tag{11}$$

one can obtain an expression for P_{Sn}: the stationary probability of being in the security breach state. Now, one can use traditional dependability techniques (see e.g., [13] or [5]) to compute quantitative measures of the system's operational security. Using the approach outlined in [5], the mean time to first security breach (MTFSB) for our model can be computed as

$$MTFSB = \frac{1 - P_{Sn}}{P_{Sn} \cdot v_{n0}}. \tag{12}$$

Hence, by parameterizing the model, solving (11) and using (12), MTFSB can easily be obtained for the system. The MTFSB measure provides the mean time it takes before the system reaches its defined security breach state for the first time. For a system, which starts in a initially secure state, MTFSB will be a quantitative measure of the system's operational security when considering a certain kind of attack posed upon the system.

As previously mentioned, using other types of distributions than the n.e.d. will exclude many of the well-known methods for analytic analysis of the model, however, simulation can then be used to obtain the MTFSB for the system.

6. Application

6.1 Example of an Attacker-System Game Model

A typical example of how an attacker may experience the outcome of his two choices in state i is

$u_i =$

	d_i	ϕ_i
a_i	u_{i1}	u_{i2}
r_i	u_{i3}	u_{i4}

$=$

	detected	undetected
attack	-1	2
give up	0	-1

. (13)

If the attacker chooses to perform the elementary attack action, without being detected, he receives a positive payoff ($u_{i2} = 2$). But if he performs the action and the system will detect this kind of violation, he receives a negative payoff ($u_{i1} = -1$). On the other hand, if the attacker chooses to resign the attack, even though he would not been detected if he had tried, he also receives a negative payoff ($u_{i4} = -1$). However, if he chooses to resign when he would have been detected if he tried, no payoff is received ($u_{i3} = 0$).

The attacker's expected payoff $u_i(a_i)$ as a function of the probability $\alpha_i(a_i)$ of trying the attack action for this game is illustrated in Fig. 4. The dashed line displays the expected payoff when the system always detects the attack action (i.e., $\alpha_i(d_i) = 1$) whereas the dotted line displays the expected payoff if the system never detects the attack action (i.e., $\alpha_i(d_i) = 0$). Hence, the strategy which provides the attacker with the highest expected payoff is the minimax solution of this game

$$p_i(a) = \alpha_i^*(a_i) = 0.25 \tag{14}$$

as indicated the Fig. 4 and verified by the Gambit software tool [11].

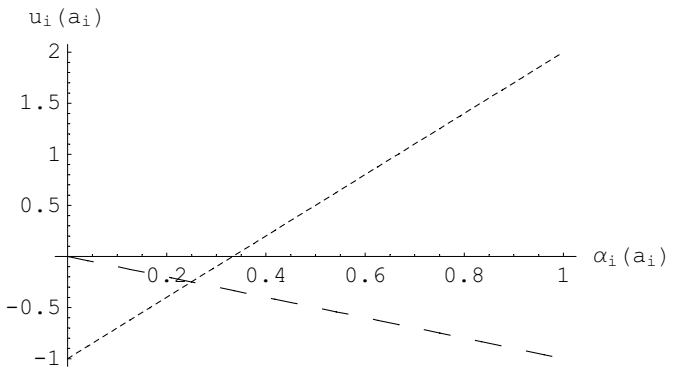


Figure 4. The expected payoff for the game in (13).

6.2 Example of Quantification of an Attack

In a root privileges attack, an attacker tries to obtain root access to a Unix or Linux system connected to a network. Assuming that the attacker is not an insider (a registered user), one common way to gain root access is to

1 crack or sniff passwords to get access to local user account, and

2 trigger a local exploit, e.g., the mremap() exploit on Linux, to get root privileges.

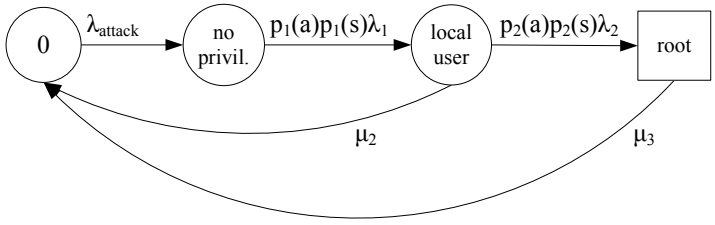


Figure 5. A stochastic model for obtaining root access of a Linux/Unix system.

The stochastic model for this attack scenario is displayed in Fig. 5. For security quantification the model has been parametrized with the values indicated in Table 1. To shorten the example, the attacker's payoffs for attack step 1 and 2 are both assigned values as in the example (13) with the minimax solution $\alpha_i^*(a_i) = 0.25$ (14), which gives $p_1(a) = p_2(a) = \alpha_i^*(a_i) = 0.25$.

Parameter	Value (s)
$1/\lambda_{attack}$	$9.0 \cdot 10^5$
$1/\lambda_1$	$1.8 \cdot 10^4$
$1/\lambda_2$	$2.2 \cdot 10^4$
$p_1(s)$	0.9
$p_2(s)$	0.7
$1/\mu_2$	$8.5 \cdot 10^4$
$1/\mu_3$	$3.6 \cdot 10^3$

Table 1. Numerical values for the parameters of the model in Fig. 5

By inserting the values from Table 1 and solving the equation system

$$\begin{cases} P_{S0} \cdot \lambda_{attack} & = P_{S1} \cdot p_1(a)p_1(s)\lambda_1 \\ P_{S1} \cdot p_1(a)p_1(s)\lambda_1 & = P_{S2} \cdot \big(p_2(a)p_2(s)\lambda_2 + \mu_2\big) \\ P_{S2} \cdot p_2(a)p_2(s)\lambda_2 & = P_{S3} \cdot \mu_3 \\ P_{S0} + P_{S1} + P_{S2} + P_{S3} & = 1 \end{cases} \tag{15}$$

we obtain an expression for P_{S3}, hence, by using (12), MTFSB for this type of attack is calculated as

$$MTFSB = \frac{1 - P_{S3}}{P_{S3} \cdot \mu_3} = ... = 2.555 \cdot 10^6 \text{ (sec)} \approx 30 \text{ (days)}. \tag{16}$$

The MTFSB measure of 30 days reflects how long one can expect that the system will remain secure from illegal root access when attacked by non-registered users under the given assumptions.

7. Conclusions and Further Work

In this paper, game theory is suggested as a method for modeling and computing probabilities of the expected behavior of attackers in quantitative stochastic models of security. One example of an attack against a system is modeled and the operational measure "mean time to first security breach" (MTFSB) is computed for the attack.

The model presented here is very simple. Further work will therefore include extending the game theoretic model with different attacker profiles; not all attackers will experience equal payoffs and some attackers tend to take more risks than others. Models including more than one type of attack and where there are more than one possible way to reach the security breach state, resulting in more complex attack graphs, will be developed. To avoid state space explosion in larger models, the possibilities of using stochastic Petri nets as a modeling and analyzing tool will be considered.

Regarding the game theoretic model, it is interesting to note that if the attacker knows the probability of getting caught at a certain stage of the attack, then there will

always be a pure strategy (either to always attack or to always resign) that maximizes his expected received payoff. Also, in cases where the attacker does not know the exact probability of getting caught there might be other strategies than the minimax solution which gives him a larger payoff. However, as discussed in [15] when leaving the minimax strategy the attacker looses his guarantee of expected payoff and taking such risks seems contrary to human nature!

Furthermore, it can be argued that the players in this game (the attackers) may be unaware of, or ignore, the fact that they are playing a repeated game, hence, statistics of attacker behavior may not always converge to equilibrium in practice. A "one-shot game" with a pure minimax solution may therefore in many cases be more appropriate for modeling expected behavior of attackers. Whether the game model presented in this paper gives a realistic model of real world security related attacks will require further research, including validation of the model against empirical data.

References

[1] T. Alpcan and T. Basar. A game theoretic approach to decision and analysis in network intrusion detection. In *Proc. of 42nd IEEE Conference on Decision and Control*, 2003.

[2] K. B. B. Madan, K. Vaidyanathan Goseva-Popstojanova, and K. S. Trivedi. A method for modeling and quantifying the security attributes of intrusion tolerant systems. In *Performance Evaluation*, volume 56, 2004.

[3] S. Brocklehurst, B. Littlewood, J. Olovsson, and E. Jonsson. On measurement of operational security. *Aerospace and Electronic Systems Magazine, IEEE*, 9, 1994.

[4] R. Gibbons. *Game Theory for Applied Economists*. Princeton University Press, 1992.

[5] B. E. Helvik. *Dependable Computing Systems and Communication, Design and Evaluation*. 2003. Draft lecture notes for the course TTM4120 Dependable Systems, NTNU Norway.

[6] E. Jonsson and T. Olovsson. A quantitative model of the security intrusion process based on attacker behavior. *IEEE Trans. Software Eng.*, 4(25):235, April 1997.

[7] B. Littlewood, S. Brocklehurst, N. Fenton, P. Mellor, S. Page, D. Wright, J. Dobson, McDermid J., and D. Gollmann. Towards operational measures of computer security. *Journal of Computer Security*, 2:211–229, Oct 1993.

[8] Peng Liu and Wanyu Zang. Incentive-based modeling and inference of attacker intent, objectives, and strategies. In *Proceedings of the 10th ACM conference on Computer and communication security*, 2003.

[9] K. Lye and J. M. Wing. Game strategies in network security. In *Proc. of 15th IEEE Computer Security Foundations Workshop*, 2002.

[10] B.B. Madan, K. Vaidyanathan, and K.S. Trivedi. Modeling and quantification of security attributes of software systems. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN'02)*, 2002.

[11] McLennan Andrew M. McKelvey, Richard D. and Theodore L. Turocy. Gambit: Software tools for game theory, version 0.97.0.6, 2004. http://econweb.tamu.edu/gambit/.

[12] R. Ortalo and Y. Deswarte. Experimenting with quantitative evaluation tools for monitoring operational security. *IEEE Transactions on Software Engineering*, 25(5):633–650, Sept/Oct 1999.

[13] A. Puliafito R. A. Sahner, K. S. Trivedi. *Performance and Reliability Analysis of Computer Systems*. Kluwer Academic Publishers, 1996.

[14] S. M. Ross. *Introduction to Probability Models*. Academic Press, 8 edition, 2003.

[15] S. Stahl. *A Gentle Introduction to Game Theory*. American Mathematical Society, 1991.

[16] D. Wang, B.B. Madan, and K.S. Trivedi. Security analysis of sitar intrusion tolerance system. *ACM SSRS'03*, 2003.

Appendix: Game Theory

Formally, a static n-player game in strategic form with complete information is a 3-tuple $(N, \{A_k\}_{k\in N}, \{u_k\}_{k\in N})$. The game consists of

- A set of players: $N = 1, ..., n$.
- The action (strategy) spaces of players: A_k,$k = 1, .., n$ where A_k is the set of all available actions to player k. The outcome space is then defined as $A = \times_{k\in N} A_k = \{(a_1, .., a_n) : a_k \in A_k, k = 1, .., n\}$ and is thus nothing but an action profile.
- The payoff function of players: $u_k : A \to \mathbf{R}, k = 1, .., n\}$.

If N and A_k are finite, the game is called *finite*. In particular, if the game is played by only two players ($N = \{1, 2\}$) exactly four pieces of information are needed to uniquely define the game: (A_1, A_2, u_1, u_2).

A (pure) *strategy* for a player is a choice of action from his set of available actions, whereas

Definition 1 A **mixed strategy** α_k for player k, is a probability distribution over his set of available actions, A_k, i.e., if player k has m actions available, a mixed strategy is an m dimensional vector

$$(\alpha_k^1, \alpha_k^2, .., \alpha_k^m), \text{ such that } \alpha_k^\beta \geq 0 \text{ for all } \beta = 1, 2, .., m, \text{ and } \sum_{\beta=1}^{m} \alpha_k^\beta = 1. \tag{A.1}$$

Hence, $\alpha_k(a_k^\beta)$ is the probability that player k will take action a_k^β.

The *payoff* function maps the action profile of a player to the corresponding consequence (reward or loss) experienced by the player. If each outcome $a \in A$ occurs with probability $p(a)$, then the *expected payoff* of player k is

$$u_k(p) = \sum_{a\in A} p(a)u_k(a). \tag{A.2}$$

In a nonzero-sum game, a *rational* player will always try to maximize her own expected payoff from the game. The choice of a_k that maximize player k's expected payoff over her action space A_k is called the player's *best response* action. The decision making of player k in a game then becomes

$$\max_{a_k\in A_k} u_k(a_k, a_{-k}), \tag{A.3}$$

where a_{-k} is the action choice of the other players, unknown by player k.

Definition 2 The **best response correspondence** of player k is the set of mixed strategies that are optimal, given the other player's mixed strategies. In other words

$$B_k(\alpha_{-k}) = \arg\max_{\alpha_k\in\Delta(A_k)} u_k(\alpha_k, \alpha_{-k}). \tag{A.4}$$

Definition 3 A **mixed strategy equilibrium** (Nash equilibrium) of a game G in strategic form is a mixed strategy profile $(\alpha_1^*, .., \alpha_n^*)$ such that, for all $k = 1, .., n$

$$\alpha_k^* \in \arg\max_{\alpha_k\in\Delta(A_k)} u_k(\alpha_k, \alpha_{-k}^*), \tag{A.5}$$

or

$$\alpha_k^* \in B_k(\alpha_{-k}^*). \tag{A.6}$$

In a zero-sum two-player game where one player's gain is the other player's loss the Nash equilibrium of the game is also called the *minimax* solution of the game.

PAPER B

Using Stochastic Game Theory to Compute the Expected Behavior of Attackers

Karin Sallhammar, Svein J. Knapskog and Bjarne E. Helvik

In Proceedings of the 2005 International Symposium on Applications and the Internet (Saint 2005)

Trento, Italy, January 31 - February 4, 2005

USING STOCHASTIC GAME THEORY TO COMPUTE THE EXPECTED BEHAVIOR OF ATTACKERS

K. Sallhammar, S.J. Knapskog and B.E. Helvik
Centre for Quantifiable Quality of Service in Communication Systems
Norwegian University of Science and Technology,
Trondheim, Norway
{ sallhamm, knapskog, bjarne } @q2s.ntnu.no

Abstract This paper presents ongoing work on using stochastic game theory as a mathematical tool for computing the expected behavior of attackers. The possible use of the Nash equilibrium as a part of the transition probabilities in state transition models is defined and motivated. To demonstrate the approach, a simple example of an attack against a computer network is modeled and analyzed.

1. Introduction

Recently there has been an increased interest in probabilistic methods for quantifying the operational security of networked computer systems. Much of the recent research efforts [10, 1, 3, 6] have focused on using state transition diagrams to model attacks and system restoration, thereby obtaining quantitative measures such as the mean time to security compromise or the mean effort to security failure.

Following the approach of [1], the security of an operating computer system can be modeled as a continuous time Markov chain (CTMC)

$$\{X(t) : t \geq 0\},\ X_s = \{0, 1, 2, \ldots, z\}. \tag{1}$$

If $X(t) = k$ then the system is said to be in state k at time t. The interactions between the states in (1) can be displayed in a state transition diagram. Attacks on the system can then be modeled as a series of intentional state changes of the underlying stochastic process. The state $X_S = 0$ is considered a dormant state, and is hence not included in the analysis in Section 2.

To correctly assess the security of a real world system, any probabilistic model has to incorporate the attacker behavior. Previous work is mainly based on using Markov decision processes [3] or shortest path algorithms [6] to include this aspect. However, none of these methods take into account that the attacker may consider not only the reward of a successful attack but also the potential cost he may experience if the attack is detected and reacted to by the system administrator ("cost" in this paper

should be interpreted as a negative reward). This aspect has previously been ignored in models of attacker behavior. In this paper stochastic game theory is therefore introduced as a tool to model and analyze the expected attacker behavior.

Game theory is not a new concept in security related contexts. In [5], a game theoretic method for analyzing the security of computer networks is presented. The interactions between an attacker and the administrator are modeled as a two-player stochastic game for which best-response strategies (the Nash equilibrium) are computed. In [4] a preliminary framework for modeling attacker intent, objectives and strategies (AIOS) is presented. To infer AIOS a game theoretic approach is used and models for different threat environments are suggested.

This work extends the work of [5, 4] by motivating and defining how to compute and use the Nash equilibrium (NE) of a stochastic game to model the expected attacker behavior for use in state transition models, under the assumption that an attacker will consider the reward versus the possible cost of his actions before he acts.

2. Modeling Attacks

The networked systems of today are very complex and once an attack has been initiated, the attacker often has many atomic attack actions to choose between. He may also choose to interrupt an ongoing attack at a certain stage, or not to start the attack at all. To include these aspects in the transition probabilities between the states of (1), it is necessary to analyze all the options an attacker has in every state of the system. Assuming that for each state k: $k = 1, \ldots, z$, an attacker can take m_k actions

- Attack by choosing one of the possible atomic attack actions a_i^k, where $i = 1, \ldots, m_k - 1$.
  - If the action succeeds the attacker will receive the reward associated with the particular attack.
  - If the action fails no reward will be achieved.
  - If the action is detected, the attacker will receive the cost associated with the attack.
- Resign and interrupt the ongoing attack. This is denoted action $a_{m_k}^k$.
  - The attacker will most likely experience this option as a defeat, hence, by resigning he will receive a cost, which magnitude depends on both how far the attack has proceeded as well as the probability that the attack would have remained undetected if he had chosen to continue.

The probability that the attacker will choose action i in state k will be denoted $p_{attack}(a_i^k)$. Hence, for each state k in the state transition model, the attacker's expected choice of action can be represented by a probability vector

$$\underline{p}_{attack}(a^k) = (p_{attack}(a_1^k), \ldots, p_{attack}(a_{m_k}^k)), \quad \text{where} \sum_{i=1,\ldots,m_k} p_{attack}(a_i^k) = 1. \tag{2}$$

Hence,

$$\mathbf{p_{attack}} = \{\underline{p}_{attack}(a^k)|k = 1, \ldots, z\}, \tag{3}$$

will be the complete set of decision probability vectors for the state transition model.

To continue an attack from state k, the attacker not only has to decide upon an atomic attack action but he also must succeed with the action. Assuming that the reward and cost result from the transitions (entering a new state), we may without loss of generality consider the embedded discrete time Markov chain (DTMC) of (1). The probability that an attacker may cause a transition of the system from state k to state ki can therefore be computed as

$$P(X_{s+1} = ki|X_s = k) = p_{attack}(a_i^k) \cdot p_{success}(a_i^k). \tag{4}$$

If the attacker chooses action $a_{m_k}^k$, or fails, no state change occurs. Hence, to incorporate attacker behavior in the transition probabilities when parameterizing a state transition model, one should

1 Identify all atomic attack actions, i.e., those transitions that can be caused by attackers;

2 Assign success probabilities to the atomic attack actions; and

3 Compute and assign decision probabilities to the atomic attack actions.

The rest of this paper will concentrate on how to perform the third step; i.e., explain how to use stochastic game theory to compute the expected behavior of attackers.

3. Computing Decision Probabilities

Regard each atomic attack action, which may cause a transition of the current system state, as an action in a game where the attacker's choices of action is based on an intelligent consideration of the possible consequences. Then the interactions between the attacker and the system can be modelled as a two-player game, as illustrated in Fig. 1.

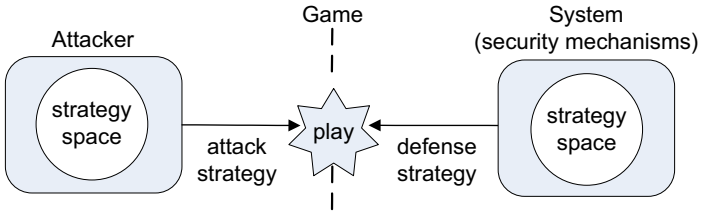


Figure 1. The interactions between an attacker and the system modeled as a two-player game

To obtain the attack matrix (3) of an intelligent and rational attacker *stochastic game theory* can be used. A formal definition of a stochastic game is given in Appendix. In a stochastic game there is a finite number of positions, or states. After each

play in the game, the game either ends or it is transferred to the next state. For the transitions between states there is a randomization involved, which can be related to the success probabilities of atomic attack actions. A general stochastic game model for analyzing atomic attack actions can therefore be defined by

- The two players: $N = \{1, 2\} = \{$*attacker, system*$\}$,
- The game elements: $\Gamma_k, k = 1, \ldots, z,$ and $\forall k$,
- The attacker actions: $A_k = \{a_1^k, \ldots, a_{m_k}^k\},$ and the consequences $C_k = \{d^k, \phi^k\} = \{$*detected*, *undetected*$\}$.

For each possible combination of attack and consequence, there is an entry, as defined in (A.1). In this model, the entry

$$\mu_{a_i,\phi}^k = p_{success}(a_i^k)\left(u(a_i^k, \phi^k) + \Gamma_{ki}\right), \tag{5}$$

means that if the attacker undetected succeeds with the atomic attack action i in state k, he receives the payoff $u(a_i^k, \phi^k)$. Note that the probability of having to play the subsequent game element Γ_{ki}, i.e., to continue the attack, is included in (5). In contrast

$$\mu_{a_i,d}^k = u(a_i^k, d^k), \tag{6}$$

is the entry corresponding to the case where the attacker chooses action i but the action is *detected* and reacted to by the system administrator. Entries similar to (6) can be defined for the cases where the attacker resigns and interrupts the attack. As can be seen, the game is assumed to end if the attacker chooses to resign the ongoing attack, or if the ongoing attack is detected.

Assuming that an attacker does not know the exact probability that his possible actions will remain undetected, game theory says he should assume that his opponent (the system) is a conscious player of the game, who seeks to minimize the attacker's expected payoff [9]. Hence, by using the inductive equations (A.3)-(A.5) the minimax solution, i.e., the NE of the complete stochastic game can be calculated. The set of minimax solution vectors in (A.6) for the game elements Γ_k in the stochastic game model represents a complete attack strategy, which has the property that, by following it, an attacker will know that he has maximized his expected payoff of the attack. This gives him a guarantee of the result from the attack regardless of whether his atomic attack actions are detected by the system or not; the "no regrets property" of game theory. Several experiments indicate that this search for guarantees is a very strong motivator of human behavior and assuming that the attacker population targeting the system will make rational choices relative to their objectives, the situation will in the long run naturally gravitate towards the NE [2, 9]. The minimax strategy will therefore indicate how one can expect *rational* attackers to behave.

In cases where there is a single NE solution of the game (as for zero-sum games), the set of attacker decision probability vectors for all states in the stochastic model can be directly derived from the minimax solution of the underlying stochastic game as

$$\mathbf{p_{attack}} = \boldsymbol{\alpha}^* = \{\underline{\alpha}^k | k = 1, \ldots, z\}^*. \tag{7}$$

	(1,0,0)	(0,1,0)	(0,0,1)	(1,1,0)	(1,0,1)	(0,1,1)	(1,1,1)
(0,0,0)	0,6	0,4	0,1	-	-	-	-
(1,0,0)	-	-	-	0,5	0,2	-	-
(0,1,0)	-	-	-	-	-	0,3	-
(1,1,0)	-	-	-	-	-	-	0,4

Table 1. Success probabilities for the example network state set.

In cases of multiple NE's, the strategy probabilities can be combined (added) and normalized according to (2).

4. Example

This is a simple example (very similar to the one used in [8]) to illustrate the possible use of the theory previously presented. The network illustrated in Fig. 2 consists of a workstation, a public webserver and a private fileserver.

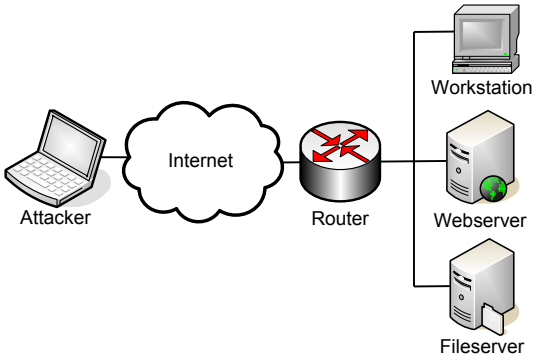


Figure 2. The example network.

For the example network we use the following notation

$$X_s = \{(x, y, z)|x, y, z \in \{0, 1\}\},$$

where e.g., $(1, 0, 1)$ means that the workstation (x) and fileserver (z) are compromised, but the webserver (y) is not. To simplify the example we assume that the attacker priorities are: 1) the fileserver (reward 30); 2) the webserver (reward 20); 3) the workstation (reward 10); and once he has compromised one of them, he will only focus on targets with higher priorities. Also assuming that once an attacker has taken control over any of the possible targets, the probability of successful attacks against the remaining targets will increase, the partial transition matrix with success probabilities in Table 1 can be set up. (The numerical values for the success probabilities are chosen for exemplification.)

By defining the action set

$$\begin{aligned} A_{(0,0,0)} &= \{a_1, a_2, a_3, a_4\}, \\ A_{(1,0,0)} &= \{a_2, a_3, a_4\}, \\ A_{(0,1,0)} &= A_{(1,1,0)} = \{a_3, a_4\}, \\ C_{(0,0,0)} &= C_{(1,0,0)} = C_{(0,1,0)} = C_{(1,1,0)} = \{c_1, c_2\}, \end{aligned} \tag{8}$$

where $a_1 =$ "attack workstation", $a_2 =$ "attack webserver" ,$a_3 =$ "attack fileserver", $a_4 =$ "do nothing", $c_1 =$ "detected" and $c_2 =$ "undetected", and solving the corresponding game elements

$$\begin{aligned} \Gamma_{(0,0,0)} &= \begin{pmatrix} -10 & 0.6 \cdot (10 + \Gamma_{(1,0,0)}) \\ -20 & 0.4 \cdot (20 + \Gamma_{(0,1,0)}) \\ -30 & 0.1 \cdot 30 \\ 0 & -5 \end{pmatrix}, \\ \Gamma_{(1,0,0)} &= \begin{pmatrix} -20 & 0.5 \cdot (20 + \Gamma_{(1,1,0)}) \\ -30 & 0.2 \cdot 30 \\ 0 & -10 \end{pmatrix}, \\ \Gamma_{(0,1,0)} &= \begin{pmatrix} -30 & 0.3 \cdot 30 \\ 0 & -15, \end{pmatrix}, \\ \Gamma_{(1,1,0)} &= \begin{pmatrix} -30 & 0.4 \cdot 30 \\ 0 & -15, \end{pmatrix}, \end{aligned} \tag{9}$$

the decision vector for rational attackers can be computed from the optimal solution for the underlying stochastic game as

$$\begin{aligned} \underline{p}_{attack}(a^{(0,0,0)}) &= (0.29,\ 0,\ 0,\ 0.71), \\ \underline{p}_{attack}(a^{(1,0,0)}) &= (0.28,\ 0,\ 0.72), \\ \underline{p}_{attack}(a^{(0,1,0)}) &= (0.28,\ 0.72), \\ \underline{p}_{attack}(a^{(1,1,0)}) &= (0.26,\ 0.74). \end{aligned} \tag{10}$$

The interpretation of the numerical result displayed in (10) is that a rational attacker will, with probability 0.71, consider it too risky to attack the network at all, however with probability 0.29 he will start to attack the workstation. Once he has gained control of the workstation he might, with probability 0.28, continue the attack by trying to compromise the webserver; etc.

5. Conclusions and Further Work

This model is quite simple. The decision probability vector as defined in (7) should therefore not be taken as an absolute truth of how the attackers will behave, but rather as an indication of their expected behavior. This is especially important in cases where the solution of the game is a pure NE, or when some options have zero probability (as in the example). Further work will therefore try to combine the optimal

strategy vector (A.6) with uncertainty, to derive a more realistic decision vector. The model should also be extended to include different attacker profiles; not all attackers will experience equal rewards or losses, and some attackers tend to take more risks than others.

Regarding the game theoretic model, it is interesting to note that if the attacker knows the probability of getting caught at a certain stage of the attack, there will always be a pure strategy that maximizes his expected received payoff. Also, in cases where the attacker does not know the exact probability of getting caught there might be other strategies than the minimax solution which provide a larger payoff. However, as discussed in [9], when leaving the minimax strategy the attacker looses his guarantee of expected payoff and taking such risks seems contrary to human nature!

References

[1] K. B. B. Madan, K. Vaidyanathan Goseva-Popstojanova, and K. S. Trivedi. A method for modeling and quantifying the security attributes of intrusion tolerant systems. In *Performance Evaluation*, volume 56, 2004.

[2] R. Gibbons. *Game Theory for Applied Economists*. Princeton University Press, 1992.

[3] S. Jha, O. Sheyner, and J. Wing. Two formal analyses of attack graphs. In *Proceedings of the 2002 Computer Security Foundations Workshop*, 2002.

[4] Peng Liu and Wanyu Zang. Incentive-based modeling and inference of attacker intent, objectives, and strategies. In *Proceedings of the 10th ACM conference on Computer and communication security*, pages 179–189, 2003.

[5] K. Lye and J. M. Wing. Game strategies in network security. In *Proceedings of the 15th IEEE Computer Security Foundations Workshop*, 2002.

[6] R. Ortalo and Y. Deswarte. Experimenting with quantitative evaluation tools for monitoring operational security. *IEEE Transactions on Software Engineering*, 25(5):633–650, Sept/Oct 1999.

[7] G. Owen. *Game Theory*. Academic Press, 2 edition, 1982.

[8] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J.M. Wing. Automated generation and analysis of attack graphs. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2002.

[9] S. Stahl. *A Gentle Introduction to Game Theory*. American Mathematical Society, 1991.

[10] F. Stevens, T. Courtney, S. Singh, A. Agbaria, J. F. Meyer, W. H. Sanders, and P Pal. Model-based validation of an intrusion-tolerant information system. In *Proceedings of the 23rd Symposium on Reliable Distributed Systems (SRDS 2004)*, Oct 18-20 2004.

Appendix: Stochastic Games

A two-player zero-sum stochastic game [7] is a set of z "game elements", or states, Γ_k: $k = 1, \ldots, z$. Each game element is represented by an $m_k \times n_k$ matrix, whose entries are of the form

$$\mu_{ij}^k = u_{ij}^k + \sum_{l=1,\ldots,z} q_{ij}^{kl} \Gamma_l, \tag{A.1}$$

with $q_{ij}^{kl} \geq 0$, and $\sum_{l=1,\ldots,p} q_{ij}^{kl} < 1$. Hence, if in state k, player I chooses his ith pure strategy and player II chooses his jth pure strategy, player I will receive a payoff of u_{ij}^k plus a probability of q_{ij}^{kl} for $l = 1, \ldots, z$ of having to play the lth game element next. In contrast to other game models, it is possible for a stochastic game to revert to previous positions.

Definition. A strategy for player I is a set $\underline{\alpha}^{kt}$, $k = 1, \ldots, z$ of m_k vectors satisfying

$$\sum_{i=1,\ldots,m_k} \alpha_i^{kt} = 1,\ \alpha_i^{kt} \geq 0. \tag{A.2}$$

Hence, α_i^{kt} is the probability that player I will choose action i, assuming that he plays the game element Γ_k at the tth stage of the game. The strategy is *stationary* if, for all k, the vectors $\underline{\alpha}^{kt}$ are independent of t. A strategy for player II is a similar set of n_k vectors $\underline{\beta}^{kt}$.

Given a pair of strategies, an expected payoff can be calculated for any $k = 1, \ldots, z$ when the first stage of the game is the game element Γ_k. Thus, the expected payoff for a pair of strategies can be thought of as a z-vector. As with ordinary matrix games, this will lead to the definition of optimal strategies and a value, the value being a z-vector $v = (v_1, v_2, \ldots, v_z)$.

If the value vector is to exist, one must be able to replace the game element Γ_l in (A.1) by the value component v_l, i.e., by defining, inductively,

$$v^0 = (0, 0, \ldots, 0), \tag{A.3}$$

$$x_{ij}^{kr} = u_{ij} + \sum q_{ij}^{kl} v_i^r, r = 1, 2, \ldots, \tag{A.4}$$

$$v_k^{r+1} = \text{val}(x_{ij}^{kr}), \tag{A.5}$$

the sequence of value vectors will eventually converge (for a proof of convergence, see [7]). The optimal strategy α^{kr} for the converged value vector will then converge, in the limit, to the *optimal stationary strategy*

$$\boldsymbol{\alpha}^* = \{\underline{\alpha}^k | k = 1, \ldots, z\}^*, \tag{A.6}$$

for the stochastic game.

PAPER C

Incorporating Attacker Behavior in Stochastic Models of Security

Karin Sallhammar, Bjarne E. Helvik and Svein J. Knapskog

In Proceedings of the 2005 International Conference on Security and Management (SAM'05)

Las Vegas (NV), USA, June 20-23, 2005

INCORPORATING ATTACKER BEHAVIOR IN STOCHASTIC MODELS OF SECURITY

Karin Sallhammar, Bjarne E. Helvik and Svein J. Knapskog
Centre for Quantifiable Quality of Service in Communication Systems
Norwegian University of Science and Technology,
Trondheim, Norway
{ sallhamm, bjarne, knapskog } @q2s.ntnu.no

Abstract We describe a game-theoretic method to compute the probabilities of expected attacker behavior, and we demonstrate how these probabilities can be incorporated in the transition rate matrix of a stochastic model for operational security assessment. The game-theoretic method is based on a reward concept, which considers the effect of successful attacks, as well as the possible cost of detection when computing the expected attacker strategy. Our method aims to fill an important gap in the application of reliability methodology to security evaluation. To demonstrate the usability of the method in different threat environments, an illustrative example is provided.

Keywords: Quantitative security, stochastic modeling, attacker behavior, game theory

1. Introduction

The new paradigms of ubiquitous computing and high capacity data transfer open up the Internet as the main area for information interchange and electronic commerce. Attacks against computer networks used by modern society and economics for communication and finance can therefore threaten the economical and physical well-being of people and organizations. Due to interconnection of systems, such attacks can be carried out not only locally, but also anonymously and from a safe distance. Assessment of the operational security of an ICT system is hence a research area of ever increasing interest.

There are several standards targeting both security assessment and security management available. The ISO 15408 "Common Criteria" standard [4] provides criteria for qualitative evaluations of the security level of a system, while ISO 13355 "Guidelines for the management of IT Security" [3] provides guidelines on risk management of IT security. However, these standards focus on qualitative evaluation of the security level of a system rather than providing quantitative assessment of operational security. Recently, the need for techniques for quantification of security attributes of ICT systems has been raised. This relates both to security requirements in QoS architectures, as well as input to trade-off analysis regarding the design and choice of security mechanisms to comply with an established security policy. One way to

achieve this, is to use probabilistic models for assessing the operational security of an ICT system or service. Specifically, stochastic modeling and analysis techniques, traditionally used in reliability analysis, have been identified as a promising approach for obtaining quantitative measures of the security attributes of systems.

In the groundbreaking paper by Littlewood et. al. [6] a first step towards operational measures of computer security is discussed. The authors point out the lack of quantitative measures for determining operational security and relate security assessment to the reliability domain. Quantitative measures, such as the *mean effort to security breach*, are defined and discussed. Ortalo et.al. [11] present a quantitative model to measure known Unix security vulnerabilities using a privilege graph, which is transformed into a Markov chain. The model allows for the characterization of operational security expressed as the mean effort to security failure as proposed by [6]. Furthermore, Madan et. al. [9, 16, 2] use traditional stochastic modeling techniques to capture attacker behavior and the system's response to attacks and intrusions. A quantitative security analysis is carried out for the steady state behavior of the system. In [15] Singh et. al. describe an approach for probabilistic validation of an intrusion-tolerant replication system. They provide a hierarchical model using stochastic activity nets (SAN), which can be used to validate intrusion tolerant systems, and to evaluate merits of various design choices. Finally, the paper by Nicol et. al [10] provides a survey over the existing model-based system dependability evaluation techniques, and summarizes how they are being extended to evaluate security.

The stochastic model used in this paper is similar to the ones presented in [9, 16, 2, 11, 15]. However, to correctly model intentional attacks posed upon a system, any probabilistic model has to incorporate attacker behavior. We believe this aspect to be one of the remaining main challenges when using stochastic modeling techniques to quantify security. We argue that the attacker behavior should ideally be represented as a probability distribution over the possible attack actions in each state of the system. We therefore define and make use of attacker strategies as a part of the transition probabilities between states. To compute the expected attacker strategies, we use *stochastic game theory*.

Game theory in a security related context has been utilized in previous papers. In [1] a model for attacker and intrusion detection system (IDS) behavior within a two-person, nonzero-sum, non cooperative game framework is suggested. The possible use of game theory for development of decision and control algorithms is investigated. In [8] a game-theoretic method for analyzing the security of computer networks is presented. The interactions between an attacker and the administrator are modeled as a two-player stochastic game for which best-response strategies (Nash equilibrium) are computed. In [7] a preliminary framework for modeling attacker intent, objectives, and strategies (AIOS) is presented. To infer AIOS a game-theoretic approach is used and models for different threat environments are suggested. In the predecessor of this paper [14] stochastic game theory is used to compute the expected attacker behavior. The use of the Nash equilibrium as a part of the transition probabilities in state transition models is introduced. This paper continues where [14]

ends, by elaborating the method for computing expected attacker behavior for use in stochastic models of security. In contrast to the game model in [8], we model the outcome of the game elements as the possible consequences of the attackers' actions being detected or not, and in contrast to [7] we can use the same game model for different threat environments. Furthermore, we include the time aspect in the success probabilities of attack actions; an important aspect that was neglected in both [14] and [8].

The paper is organized as follows. Section 2 presents the stochastic model, and motivates the use of attacker strategies as a part of its transitions between states. In Section 3, two crucial factors, which motivate the attacker's choice of strategy, are discussed. The reward model is defined. Section 4 presents the game-theoretic model used in this paper and explains how the model can be used to compute the expected attacker strategy. In Section 5, the theory is demonstrated by a simple example. Finally, Section 6 concludes the paper by summing up its main contributions and outlining future work.

2. State-based Stochastic Modeling

In the security community one often talks of a *security breach*; a state where the system deviates from its security requirements. A security breach might accidentally be caused by random failures during normal usage operation, by intentional attacks upon the system, or a combination of these. Such attacks on an operating computer system often consist of many successive atomic attack actions, and can hence be modeled as a series of state changes of the system leading from an initially secure state to one of several possible compromised states.

Following our initial approach in [14], we model the security of a system as a continuous-time Markov chain (CTMC) with a finite number of states $i = 1, \ldots N$. Let

$$\mathbf{X}(t) = \{X_1(t), X_2(t), \ldots, X_N(t)\}, \tag{1}$$

where $X_i(t)$ denotes the probability that the system is in state i at time t. The state equation describing the system security behavior is then

$$\frac{d}{dt}\mathbf{X}(t) = \mathbf{X}(t)\mathbf{Q}, \tag{2}$$

where $\mathbf{Q}$ is the $N \times N$ state transition rate matrix of the system. The element $ij (i \neq j)$ of $\mathbf{Q}$, is

$$q_{ij} = \lim_{dt \to 0} \left\{ \frac{\Pr(\text{transition from } i \text{ to } j \text{ in } (t, t+dt))}{dt} \right\}, \tag{3}$$

and

$$q_{ii} = -\sum_{j \neq i} q_{ij}. \tag{4}$$

The effect of attacks and the system's response to intrusions can now be described by the transitions between the states $i = 1, \ldots N$. The state equation can be solved if

the initial state of the system, i.e. the vector $\mathbf{Q}(0)$, is known. To find the steady state probabilities

$$X_i = \lim_{t\to\infty} X_i(t), i = 1, \ldots N, \tag{5}$$

of the system, one can solve the set of N equations given by $N-1$ of the N equations

$$\mathbf{XQ} = \mathbf{0}, \tag{6}$$

and with the Nth equation

$$\sum_{l=1}^{N} X_l = 1. \tag{7}$$

As is common practice in reliability analysis of ICT systems, the CTMC can be used as a basis for obtaining various measures of the operational security of the system. For an example of how to compute confidentiality, integrity or availability measures from the steady-state probabilities in (5), see [2].

It is not straight-forward to apply the "fault-error-failure" reliability pathology to determine the transition rate matrix $\mathbf{Q}$ for security assessment. In quantitative reliability evaluation the stochastic variables are sampled from distributions, which are well-known to correctly represent time to different kind of failures. The faults and their sources can be very diverse. Usually, traditional reliability evaluation does not include deliberately induced faults. Defining similar rates for the transitions describing the expected time to perform a certain atomic attack action may therefore lead to very unrealistic results. One of the main reasons is that faults leading to failures are usually considered to be unintentional and independent of each other. On the other hand, attack attempts causing security compromises are caused by human beings and very often highly intentional with the specific aim of causing maximum benefit to the intruder or damage to the system. Thus, to correctly model intentional attacks posed upon a system, the model has to incorporate attacker behavior. Herein lies the main challenge when using stochastic modeling techniques to quantify security.

2.1 Incorporating Intentional Attacks

To incorporate intentional attacks into stochastic models, the attacker behavior should be represented as a probability distribution over the possible attack actions in each state of the system. These probabilities should then be reflected in the transition rates of the CTMC. As an example, consider a stochastic model for a system, where in state i the system is vulnerable to two certain types of attack actions, denoted a_1 and a_2. We may identify four possible transitions out of this state. With rate φ_{il} the system administrator will detect the vulnerability, patch the system and transfer it into the (secure) state l. Moreover, while in state i the system will experience (accidental) random software failures with rate γ_{im}, which will cause a transition to the (failed) state m. Finally, in state i attackers will be able to exploit the vulnerability, by using any of the two different methods a_1 and a_2. The success rates of these two exploits, given that they are pursued, are assumed to be λ_{ij} and λ_{ik}. The attack action rates must be multiplied with the probability that the attackers will *decide* on the respective

actions, when constructing the transition rate matrix **Q**. This is illustrated in Fig. 1, where $\pi_i(a_1)$ and $\pi_i(a_2)$ denote the probability that an attacker will choose action a_1 and a_2, respectively, when the system is in state i.

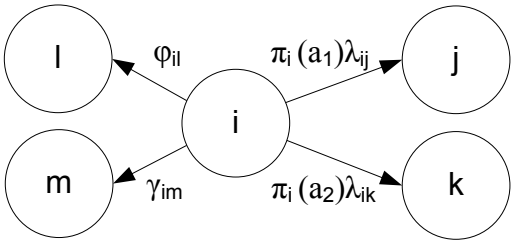


Figure 1. Transitions rates out of state i.

Hence, in this example, the ith row in the transition rate matrix **Q** will be

$$\begin{aligned} Q_{i\cdot} &\supseteq \{q_{ii}, q_{ij}, q_{ik}, q_{il}, q_{im}\} \\ &= \{-\big(\pi_i(a_1)\lambda_{ij} + \pi_i(a_2)\lambda_{ik} + \varphi_{il} + \gamma_{im}\big), \pi_i(a_1)\lambda_{ij}, \pi_i(a_2)\lambda_{ik}, \varphi_{il}, \gamma_{im}\}. \end{aligned} \tag{8}$$

Note that there will always be a possibility that an attacker does not choose any of the possible atomic attack actions a_1 and a_2, i.e.,

$$\pi_i(a_1) + \pi_i(a_2) + \pi_i(\phi) = 1, \tag{9}$$

where $\pi_i(\phi)$ represents the probability that an attacker takes no action, i.e., the attack is resigned in state i. To formalize the idea of attacker decision probabilities, we define and make use of *strategies* to model and compute the expected attacker behavior.

2.2 Attacker Strategies

Denote by A the complete set of all possible atomic attack actions on the system (ϕ included). By a *strategy* is meant a rule for choosing actions. An complete attacker strategy is denoted

$$\Pi = \{\pi_i, i = 1 \ldots, z\}, \tag{10}$$

where

$$\pi_i = \{\pi_i(a), a \in A\}, \tag{11}$$

is the strategy vector for state i. Hence, $\pi_i(a)$ is the probability that the attacker will choose action a in state i. Of course we must have

$$0 \leq \pi_i(a) \leq 1, \forall i, a, \tag{12}$$

and

$$\sum_{\forall a \in A} \pi_i(a) = 1, \forall i. \tag{13}$$

The strategy is stationary if, for all states i, the strategy vector π_i is independent of time. As an initial approach only stationary strategies will be considered in this paper, avoiding the mathematical obstacles of dynamic strategies.

To compute the attacker strategy Π, in a way that realistically models the attackers' expected behavior, one must take into account the driving forces behind the attackers' actions: who are they, what are their purposes, and why are they making the decisions they actually make?

3. Modeling Attacker Behavior

One of the most crucial factors in the analysis of the attacker's choices of action is motivation. In [13] six major factors, which motivate the attacker behavior, are identified. *Money* is the main source of motivation for actions, such as credit card theft, blackmailing or extraction of confidential information. *Entertainment* can be the cause of e.g., hacking websites or rerouting Internet browser requests. The motive of *ego* is the satisfaction and rise in self-esteem, that comes from overcoming technical difficulties, or finding innovative solutions. *Cause*, or ideology, can be based on culture, religion or social issues, and in [13] it is pointed out that it is likely to increase as a motivation factor in the future. For some attackers, *entrance* to a social group of hackers can be the driving force behind writing a particular exploit, or breaking into a particularly strong computer security defense. However, *status* is probably the most powerful motivation factor, and is currently motivating many of today's computer or network system intrusions.

On the other hand, a number of factors may reduce the attackers' motivation and make them refrain from certain attack actions. In our modeling framework, we therefore include the aspect that some attackers may be more risk adverse than others. For example, students with a user account at a university will put their enrollment status at risk if they try to abuse their insider privileges to attack their local computer network. The gain from a successful break-in into, for example, the university fileserver may therefore be smaller than the possible consequences if the intrusion is detected by the system administrators. As another example, the illegal aspect of actions (criminal offense) may prevent even a remote attacker to use available tools to exploit vulnerabilities in corporate networks.

3.1 The Reward Model

To model the attackers' motivation in a situation with a realistic risk awareness, we make use of a *reward* concept. In our model, an attacker accumulates reward during the events of the attack. Whenever an attacker performs an atomic attack action a in state i, he receives an instantaneous reward, denoted $r_i(a)$. Furthermore, if the action succeeds, an additional reward may be gained. This is modeled in terms of expected future rewards, which is due to the ability to continue the attack.

An attack action can be considered successful, if the action causes an undesirable transformation of the current system state. The transition probabilities between states will therefore be an important aspect of the expected reward when an attacker decides

upon an action. If the system is in state i, the next state of the system is determined by the embedded transition probabilities p_{ij} of (1)

$$p_{ij} = \frac{q_{ij}}{\sum\limits_{j \neq i} q_{ij}}, j = 1, \ldots N, j \neq i. \tag{14}$$

In states where there is one or more actions available to the attacker, an alternative transition probability can be computed by conditioning on the chosen action. The conditioned transition probabilities, denoted $p_{ij}(a)$, model the probability that an attacker succeeds with a particular attack action a, assuming that he does not perform two actions simultaneously.

For the example illustrated in Fig 1, we compute $p_{ij}(a_1)$ by inserting $\pi_i(a_1) = 1$ in the embedded transition probabilities in (14)

$$p_{ij}(a_1) = \frac{\lambda_{ij}}{\lambda_{ij} + \varphi_{il} + \gamma_{im}}. \tag{15}$$

Also $p_{ij}(a_2)$ can be computed in a similar manner.

Reward is a generic concept, which can be used to quantify the value of the action in terms of social status, money, satisfaction, etc, as previously discussed. To model the possible consequences experienced by risk adverse attackers, a negative reward, a *cost*, is used to quantify the impact on the attacker, as an attack action is detected and reacted to.

Henceforth, attackers are assumed to be rational: they seek to maximize their own reward from the attack. In the next section, a mathematical framework for computing the expected attacker behavior expressed in terms of the strategy Π, is presented.

4. The Game Model

In order to create a generic and sound framework for computing the expected attacker behavior, we advocate the use of stochastic game theory [12] as the mathematical tool. Regard each atomic attack action, which may cause a transition of the current system state, as an action in a game where the attacker's choices of action are based on considerations of the possible consequences. The interactions between the attacker and the system can then be modeled as a game, as illustrated in Fig. 2.

4.1 Mathematical Framework

The stochastic game, in the context of the operational security of an ICT system, is a two-player, zero-sum, multi-stage game where, at each stage, the parameters of the game depend on the current state of the CTMC introduced in Section 2. Hence, the stochastic game used to compute the attacker strategy Π can be defined as

$$\Gamma = \{\Gamma_i, i = 1, \ldots, z\}, \tag{16}$$

where Γ_i is the game element modeling the game element of state i. It is important to notice that even though the state space of the CTMC may be very large, Γ will in

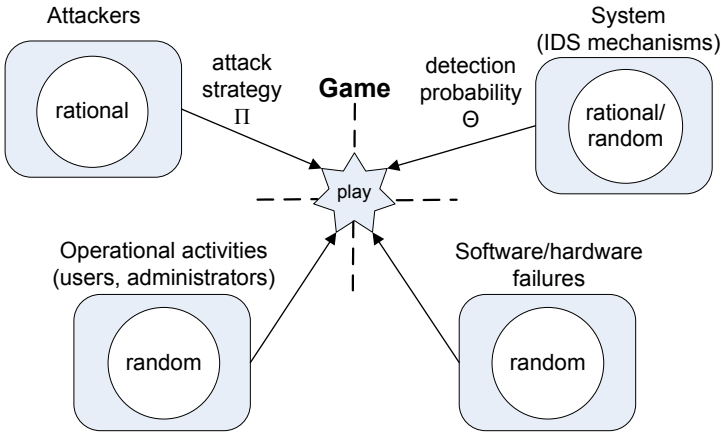


Figure 2. The interactions between an attacker and the system modelled as a game.

general span only a subset of its states, namely those states where an attacker has the possibility of performing atomic attack actions.

Each game element Γ_i can now be represented by the two-column matrix

$$\Gamma_i = \begin{array}{c} \begin{array}{cc} \text{undetected} & \text{detected} \end{array} \\ \begin{pmatrix} \vdots & \vdots \\ \mu_{i1}(a_m) & \mu_{i2}(a_m) \\ \vdots & \vdots \end{pmatrix} \end{array}, \tag{17}$$

where the rows models the possible outcome (in terms of rewards, as previously discussed) of the atomic attack actions available to the attacker in state i. The entries in (17) are of the form

$$\begin{aligned} \mu_{i1}(a_m) &= r_i(a_m|\text{undetected}) + \sum_{j=1,\ldots,z} p_{ij}(a_m)\Gamma_j, \\ \mu_{i2}(a_m) &= r_i(a_m|\text{detected}), \end{aligned} \tag{18}$$

for which $p_{ij}(a_m) \geq 0$ and $\sum_{j=1,\ldots,z} p_{ij}(a_m) < 1$. The conditional transition probabilities $p_{ij}(a_m)$ can now be computed from the embedded discrete process of (1), as explained in Section 3.1. Hence, if the attacker chooses his m'th possible action in state i, and the action remains *undetected*, the attacker will receive the reward given by $r_i(a_m|\text{undetected})$ and if the action succeeds, he will have to play the jth game element next. However, if the attack action is *detected*, the attacker receives the non-positive reward $r_i(a_m|\text{detected})$. As can be seen, the game is assumed to end if the ongoing attack is detected and reacted to, or if the system does not transfer into another game element state.

4.2 Computing the Expected Attacker Strategy

Recall that in Section 2.2 we defined the attacker strategy Π in (10) and (11) as a rule for choosing actions. Let

$$\Theta = \{\theta_i, i = 1, \ldots, z\}, \tag{19}$$

where

$$\theta_i = \{\theta_i(a), a \in A\}, \tag{20}$$

be the *counter-strategy* in the stochastic game, so that $\theta_i(a)$, $0 \leq \theta_i(a) \leq 1$, is the probability that action a will be detected in state i. The expected reward for an attacker in state i when using the strategy vector π_i can then be computed as

$$E(\pi_i, \theta_i) = \sum_{\forall a \in A} \pi_i(a) \Big((1 - \theta_i(a))\mu_{i1}(a) + \theta_i(a)\mu_{i2}(a) \Big). \tag{21}$$

A rational attacker attempting to maximize attack rewards, will choose the strategy vector π_i^*. This is the most an attacker can gain in state i, given the detection probability θ_i for the maximal expected reward

$$\max_{\pi_i} E(\pi_i, \theta_i). \tag{22}$$

The strategy π_i^* maximizing (21) is called the *optimal attack strategy* of the game element Γ_i. Hence the set of optimal strategies for the the complete stochastic game is given by

$$\Pi^* = \{\pi_i^*, i = 1, \ldots, z\}. \tag{23}$$

To compute (23), replace the game element Γ_j in (18) by the attacker's expected reward in state j and then solve the game iteratively. Further details, as well as a proof of the existence of a solution, can be found in [12].

The optimal strategy Π^* of the stochastic game has some interesting properties, which makes it a suitable candidate for modeling the attacker decision probabilities. First of all, the maximization problem in (22) is a generic expression, which can be applied in a variety of threat environments. As will be demonstrated in Section 5, it can be used to compute the expected behavior for different types of attackers: insiders as well as outsiders, risk averse as well as risk ignorant. Second, Π^* represents a complete attack strategy. It has the property that by following it, an attacker will know that he has maximized his expected attack reward. This gives him a guarantee of the result from the attack regardless of whether any of his atomic attack actions fail, or are detected by the system's intrusion detection mechanisms. This phenomenon is designated the "no regrets property" of game theory. It has been frequently applied in the field of economics within the subjects of international trade, and macroeconomics, amongst others. Several experiments indicate that this search for guarantees is a very strong motivator of human behavior. Assuming that the attacker population targeting the system makes rational choices relative to their objectives, the situation will in the long run gravitate towards the optimal strategy.

5. Example

To demonstrate the use of the method presented in this paper, a simple example (similar to the ones in [5, 14]) is provided. Due to space limitation, the state space of the example system is highly abstracted. All numerical values are chosen for exemplification. In contrast to [14], this example is solved for three different cases, hence demonstrating that the same game-theoretic method can be used in different threat environments.

The small network illustrated in Fig. 3 consists of a workstation, a public webserver and a private fileserver. To describe the security of the network, we define the

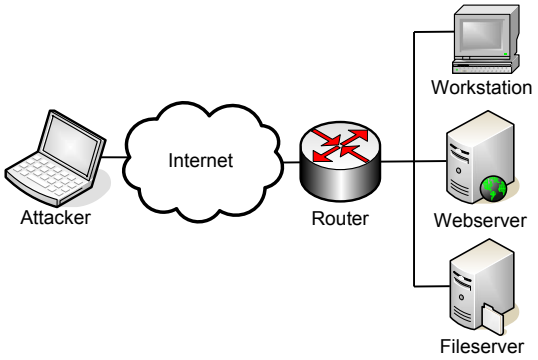


Figure 3. Network topology.

set of steady state probabilities as

$$X_i = \{i = \{x, y, z\} | x, y, z \in \{0, 1\}\}, \tag{24}$$

where e.g., $i = (1, 0, 1)$ represents the state where the workstation (x) and fileserver (z) are compromised, but the webserver (y) is not, etc. The action set can be defined as

$$A = \{a_1, a_2, a_3, \phi\}, \tag{25}$$

where $a_1 =$ "attack workstation", $a_2 =$ "attack webserver", $a_3 =$ "attack fileserver" and $\phi =$ "do nothing".

Assume that the attacker priorities, rewards and costs of actions are as shown in Table 1, and once he has compromised one of the targets, he will only focus on targets with higher priorities. Furthermore, assume there is one system administrator responsible for each workstation and server in the network, hence restoration of two or more compromised network components can take place in parallel. This is illustrated in the state transition diagram in Fig. 4, where the attack action rates and restoration rates are labeled λ and φ, respectively. For clarity, the states have also been numbered 1-8 in this figure. When using the rate values in Table 2 to compute the conditional transition probabilities, the corresponding game elements for the example network

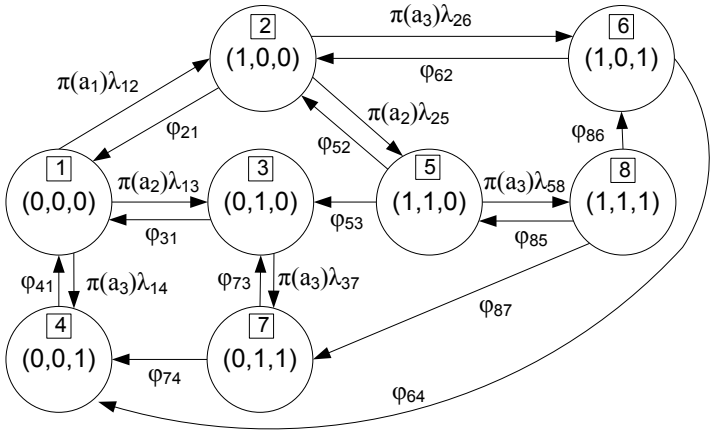


Figure 4. The network security state diagram.

will be

$$
\begin{aligned}
\Gamma_{(0,0,0)} &= \begin{pmatrix} 10 + 1.0 \cdot \Gamma_{(1,0,0)} & -10 \\ 20 + 1.0 \cdot \Gamma_{(0,1,0)} & -20 \\ 30 + 1.0 \cdot \Gamma_{(0,0,1)} & -30 \\ -5 & 0 \end{pmatrix}, \\
\Gamma_{(1,0,0)} &= \begin{pmatrix} 20 + 0.22 \cdot \Gamma_{(1,1,0)} & -20 \\ 30 + 0.30 \cdot \Gamma_{(1,0,1)} & -30 \\ -10 & 0 \end{pmatrix}, \\
\Gamma_{(0,1,0)} &= \begin{pmatrix} 30 + 0.39 \cdot \Gamma_{(0,1,1)} & -30 \\ -15 & 0, \end{pmatrix}, \\
\Gamma_{(1,1,0)} &= \begin{pmatrix} 30 + 0.22 \cdot \Gamma_{(1,1,1)} & -30 \\ -15 & 0, \end{pmatrix}, \\
\Gamma_{(0,0,1)} &= \Gamma_{(0,1,1)} = \Gamma_{(1,0,1)} = \Gamma_{(1,1,1)} = 30.
\end{aligned}
\tag{26}
$$

Priority	Action	$r_i(a$|undetected$)$	$r_i(a$|detected$)$	$\theta_i(a)$
1	a_3	+30	-30	0.6
2	a_2	+20	-20	0.8
3	a_1	+10	-10	0.2
4	ϕ	$r_{(0,0,0)}(\phi)$=-5 $r_{(1,0,0)}(\phi)$=-10 $r_{(0,1,0)}(\phi)$=-15 $r_{(1,1,0)}(\phi)$=-15	0	0

Table 1. Priorities, rewards, costs and detection probabilities.

i,j	(0,0,0)	(1,0,0)	(0,1,0)	(0,0,1)	(1,1,0)	(1,0,1)	(0,1,1)	(1,1,1)
(0,0,0)	-	3.6	2.2	1.8	-	-	-	-
(1,0,0)	8.5	-	-	-	3.7	2.4	-	-
(0,1,0)	6.2	-	-	-	-	-	4.1	-
(0,0,1)	4.1	-	-	-	-	-	-	-
(1,1,0)	-	6.2	8.5	-	-	-	-	4.0
(1,0,1)	-	4.1	-	8.5	-	-	-	-
(0,1,1)	-	-	4.1	6.2	-	-	-	-
(1,1,1)	-	-	-	-	4.1	6.2	8.5	-

Table 2. Attack and restoration rates ($10^4 s^{-1}$).

Solving the stochastic game in accordance to (22) will provide an optimal strategy vector

$$\Pi^* = \{\pi^*_{000}, \pi^*_{100}, \pi^*_{010}, \pi^*_{001}\}, \tag{27}$$

with

$$\begin{aligned}
\pi^*_{000} &= (\pi_{000}(a_1), \pi_{000}(a_2), \pi_{000}(a_3), \pi_{000}(\phi)),\\
\pi^*_{100} &= (\pi_{100}(a_2), \pi_{100}(a_3), \pi_{100}(\phi)),\\
\pi^*_{010} &= (\pi_{010}(a_3), \pi_{010}(\phi)),\\
\pi^*_{110} &= (\pi_{110}(a_3), \pi_{110}(\phi)).
\end{aligned} \tag{28}$$

The stochastic game can now be viewed as three different cases, representing three different threat environments:

Case 1. The attacker knows the detection probabilities of actions and cares about its consequences; he is what we may call a *risk averse insider*. He will try to maximize his expected reward (21) from the attack, i.e., the maximization problem (22) can be directly applied.

Case 2. The attacker does not know the detection probabilities, however he cares about the consequences; he is a *risk averse outsider*. In cases like this, it is common practice in game theory to assume that the opponent (the system) is a conscious player of the game, which seeks to minimize the attacker's expected reward. The optimal attack strategy in this case is equivalent to the Nash equilibrium (NE) solution of the stochastic game (the NE strategy pair (π^*_i, θ^*_i) solves the optimization problem $\max_{\pi_i} \min_{\theta_i} E(\pi_i, \theta_i), i = 1, \ldots, z$).

Case 3. The attacker neither knows nor cares about the consequences of being detected; he is a *risk ignorant outsider*. In a mathematical context this is equivalent of setting $\theta_i(a) = 0$, for all a in (21), which means that the second column of the game matrices in (26) is ignored in the analysis.

Solving the game, once for each case, provides the expected attacker strategy vectors in Table 3. As can be seen, an attacker in Case 2 will act more carefully than an

Π^*	Case 1	Case 2	Case 3
$\pi^*_{(000)}$	(0, 0, 1, 0)	(0, 0, 0.05, 0.95)	(0, 0, 1, 0)
$\pi^*_{(100)}$	(0, 1, 0)	(0, 0.13, 0.87)	(0, 1, 0)
$\pi^*_{(010)}$	(1, 0)	(0.17, 0.83)	(1, 0)
$\pi^*_{(110)}$	(1, 0)	(0.18, 0.82)	(1, 0)

Table 3. Optimal strategies for attackers (three cases).

attacker in Case 3. This is intuitive, since the possible cost of detected actions are included in the computation of the optimal overall strategy Π^* for the attackers in Case 2. An attacker in Case 3 ignores the possible cost of being detected and will therefore always choose to attack the target with highest reward, given that the probability of succeeding is high enough. However, even though the cost is considered in Case 1, the reward of attacks on the fileserver is high enough to make the attacker strategy in this case as aggressive as in Case 3, even though there may be a noticeable cost involved.

6. Conclusions and Further Work

In this paper we demonstrate how the expected attacker behavior can be used as a part of the transitions between states in stochastic models for security assessment. The motivation is to provide more realistic measures of the operational security of ICT systems. Moreover, this paper suggest a method to compute the expected behavior for rational attackers. The method considers the attacker rewards, resulting from successful actions, the possible costs, if the actions are detected, as well as the probabilities of succeeding with particular attack actions.

A natural extension of the model is to include time-dependent success probabilities. This may be important in cases where one can assume that the attackers learn over time or, equivalently, in cases of natural "wear-out" of security countermeasures. In theory, it is possible to solve the stochastic game model to obtain dynamic optimal strategies. Another limitation to address, is the underlying assumptions of game theory; that an attacker has a complete view of all states in the game, which may sometimes be unrealistic.

7. Acknowledgments

The authors would like to thank Tønnes Brekne, André Årnes and Marie E. G. Moe for valuable feedback on earlier versions of this paper.

References

[1] T. Alpcan and T. Basar. A game theoretic approach to decision and analysis in network intrusion detection. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, 2003.

[2] K. B. B. Madan, K. Vaidyanathan Goseva-Popstojanova, and K. S. Trivedi. A method for modeling and quantifying the security attributes of intrusion tolerant systems. In *Performance Evaluation*, volume 56, 2004.

[3] ISO/IEC 13335: Information Technology - Guidelines for the management of IT Security. http://www.iso.ch.

[4] ISO 15408: Common Criteria for Information Technology Security Evaluation, 1999. http://www.commoncriteria.org/.

[5] S. Jha, O. Sheyner, and J. Wing. Two formal analyses of attack graphs. In *Proceedings of the 2002 Computer Security Foundations Workshop*, 2002.

[6] B. Littlewood, S. Brocklehurst, N. Fenton, P. Mellor, S. Page, D. Wright, J. Dobson, McDermid J., and D. Gollmann. Towards operational measures of computer security. *Journal of Computer Security*, 2:211–229, Oct 1993.

[7] Peng Liu and Wanyu Zang. Incentive-based modeling and inference of attacker intent, objectives, and strategies. In *Proceedings of the 10th ACM conference on Computer and communication security*, pages 179–189, 2003.

[8] K. Lye and J. M. Wing. Game strategies in network security. In *Proceedings of the 15th IEEE Computer Security Foundations Workshop*, 2002.

[9] B.B. Madan, K. Vaidyanathan, and K.S. Trivedi. Modeling and quantification of security attributes of software systems. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN'02)*, 2002.

[10] David M. Nicol, William H. Sanders, and Kishor S. Trivedi. Model-based evaluation: From dependability to security. *IEEE Transactions on Dependable and Secure Computing*, 1:48–65, January-March 2004.

[11] R. Ortalo, Y. Deswarte, and M. Kaaniche. Experimenting with quantitative evaluation tools for monitoring operational security. *IEEE Transactions on Software Engineering*, 25(5):633–650, Sept/Oct 1999.

[12] G. Owen. *Game Theory*. Academic Press, 2 edition, 1982.

[13] The Honeynet Project. *Know Your Enemy*. Addison-Wesley, 2 edition, 2004.

[14] K. Sallhammar, S.J. Knapskog, and B.E. Helvik. Using stochastic game theory to compute the expected behavior of attackers. In *Proceedings of the 2005 International Symposium on Applications and the Internet Workshops (Saint2005).*, 2005.

[15] S. Singh, M. Cukier, and W.H. Sanders. Probabilistic validation of an intrusion-tolerant replication system. In de Bakker, J.W., de Roever, W.-P., and Rozenberg, G., editors, *International Conference on Dependable Systems and Networks (DSN'03)*, June 2003.

[16] D. Wang, B.B. Madan, and K.S. Trivedi. Security analysis of sitar intrusion tolerance system. *ACM SSRS'03*, 2003.

PAPER D

On Stochastic Modeling for Integrated Security and Dependability Evaluation

Karin Sallhammar, Bjarne E. Helvik and Svein J. Knapskog

The Journal of Networks (JNW, ISSN 1796-2056), Vol. 1, No. 5, September/October 2006

ON STOCHASTIC MODELING FOR INTEGRATED SECURITY AND DEPENDABILITY EVALUATION

Karin Sallhammar, Bjarne E. Helvik and Svein J. Knapskog
Centre for Quantifiable Quality of Service in Communication Systems
Norwegian University of Science and Technology,
Trondheim, Norway
{ sallhamm, bjarne, knapskog } @q2s.ntnu.no

Abstract This paper presents a new approach to integrated security and dependability evaluation, which is based on stochastic modeling techniques. Our proposal aims to provide operational measures of the trustworthiness of a system, regardless if the underlying failure cause is intentional or not. By viewing system states as elements in a stochastic game, we can compute the probabilities of expected attacker behavior, and thereby be able to model attacks as transitions between system states. The proposed game model is based on a reward- and cost concept. A section of the paper is devoted to the demonstration of how the expected attacker behavior is affected by the parameters of the game. Our model opens up for use of traditional Markov analysis to make new types of probabilistic predictions for a system, such as its expected time to security failure.

1. Introduction

Security is a concept addressing the attributes confidentiality, integrity and availability [1]. Today it is widely accepted that, due to the unavoidable presence of vulnerabilities, design faults and administrative errors, an ICT system will never be totally secure. Connecting a system to a network will necessarily introduce a risk of inappropriate access resulting in disclosure, corruption and/or loss of information. Therefore, the security of a system should ideally be interpreted in a probabilistic manner. More specifically, there is an urgent need for modeling methods that provide operational *measures* of the security. Dependability, on the other hand, is the ability of a computer system to deliver service that can justifiably be trusted. It is a generic concept, which includes the attributes reliability, availability, safety, integrity and maintainability [2]. In a dependability context one distinguishes between accidental faults, which are modeled as random processes, and intentional faults, i.e. attacks, which in most cases are not considered at all. A major drawback of this approach is that attacks may in many cases be the dominating failure cause for today's networked systems. The classical way of dependability evaluation can therefore be very

deceptive; highly dependable systems may in reality fail much more frequently than expected, due to the exploitation from attackers.

To be considered trustworthy, a system must be both dependable *and* secure. However, these two aspects have so far tended to be treated separately. A unified modeling framework for security and dependability evaluation would be advantageous from both points of view. The security community can benefit from the mature dependability modeling techniques, which can provide the operational measures that are so desirable today. On the other hand, by adding hostile actions to the set of possible fault sources, the dependability community will be able to make more realistic models than the ones that are currently in use. In this paper we review a methodology that traditionally has been used for system dependability analysis only, and motivate the application of a similar approach in the security domain. By modeling intrusions alongside with accidental failures, both the security and dependability properties of a system can be considered during the evaluation process.

Modeling and analysis of a system for predictive purposes can be performed by static or dynamic methods. This paper focuses on the dynamic method of using stochastic models (Markov chains), which is commonly used to obtain availability (the fraction of time the system is operational during an observation period) or reliability (the probability that the system remains operational over an observation period) predictions by the dependability community. The paper is organized as follows. Section 2 presents the stochastic model and explains how intrusions can be modeled as transition between states in the model. Section 3 explains how the model can be used to predict measures for the system. In Section 4, we show that the states can be viewed as elements in a stochastic game, and explain how game theory can be used to compute the expected attacker behavior. Then, in Section 5, we demonstrate how the expected attacker behavior is affected by the parameters of the game. To illustrate the approach, Section 6 includes a small case study. In Section 7 we compare our work with related research. Section 8 includes some concluding remarks and points to future work.

2. Stochastic Modeling

At the highest level of a system description is the specification of the system's functionality. The security policy is normally a part of this specification. This high level description can be used to perform qualitative assessment of system properties, such as the security levels obtained by Common Criteria evaluation [3]. Even though a qualitative evaluation can be used to rank a particular security design, its main focus is on the safeguards introduced during the development and design of the system. Moreover, such methods only evaluate static behavior of the system and do not consider dependencies of events or time aspects of failures. As a consequence, the achieved security level cannot be used to predict the system's actual behavior, i.e. its ability to withstand attacks when running in a certain threat environment. To create a model suitable for quantitative analysis and assessment of operational security and dependability, one needs to use a fine-granular system description, which is capable of incorporating the dynamic behavior of the system. This is the main strength of

state transition models where, at a low level, the system is modeled as a finite state machine. By a *state* in this context is meant an operational mode of the system characterized by which units of the system that are operational or failed, whether there are ongoing attacks, active countermeasures, operational and maintenance activities, whether parts of the system compromised or not, etc. Most systems consist of a set of interacting components and the system state is therefore the set of its component states. In a state transition model, one usually discriminates between good states and failed states, depending on whether the required service is delivered or not. Normally, a system will be subject to multiple failure cases, so that the model will have multiple failure modes. During its operational lifetime, a system will alternate between its different states. This may be due to normal usage as well as misuse, administrative measures and maintenance, as well as software- and hardware failures and repairs. The behavior of the system is therefore characterized by the *transitions* between the states, each transition triggered by an event. The event that will occur next, as well as the time until next event, is random. Hence, the behavior of the system is a stochastic process.

2.1 The Failure Process

It has been shown in [2, 4, 5] that the "fault-error-failure" pathology, which is commonly used for modeling the failure process in a dependability context, can be applied in the security domain as well. Based on the results from this research we demonstrate how a stochastic process can be used to model security failures in a similar way as the dependability community usually treats accidental and unintentional failures.

By definition, the fault-error-failure process is a sequence of events. A *fault* is an atomic phenomenon, that can be either internal or external, which causes an *error* in the system. An error is a deviation from the correct operation of the system. An error is always internal and will not be visible from outside the system. Even though a system is erroneous it still manages to deliver its intended services. An error may lead to a *failure* of the system. In a dependability context, a failure is an event that causes the delivered service to deviate from the correct service, as described in the system's functional specification. Similarly, a security failure causes a system service to deviate from its security requirements, as specified in the security policy. For each failure state, which conflicts with the system's intended functionality, we can therefore assign a corresponding property that is violated, e.g. confidentiality-failed or availability-failed. Both security- and dependability failures can be caused by a number of accidental fault sources, such as erroneous user input, administrative misconfiguration, software bugs, hardware deterioration, etc. The failures originating from most of these faults can be modeled as randomly distributed in time, as is common practice in dependability modeling and analysis. However, the ones hardest to predict are the external malicious human-made faults, which are introduced with the objective of altering the functioning of the system during use [2]. In a security context, the result of such a fault is generally referred to as an *intrusion*. Because they are intentional in nature, intrusions cannot be modeled as truly random processes. Even

though the time, or effort, to perform an intrusion may be randomly distributed, the *decision* to perform the action is not. As pointed out in [6], security analysis must assume that an attacker's choice of action will depend on the system state, may change over time, and will result in security failures that are highly correlated.

2.2 Modeling Intrusion as Transitions

To be able to model the effect of an intrusion as a transition between a good system state and a failed system state, one needs to take a closer look at the intrusion process itself. According to [4], there are two underlying causes of any intrusion:

- At least one *vulnerability*, i.e. weakness, in the system. The vulnerability is possible to exploit, however, it will require a certain amount of time from an attacker.
- A *malicious action* that tries to exploit the vulnerability. Since the action is intentional, a decision is implicitly made by the attacker. All attackers will not choose the same course of action. Hence, there will be a probability that an attacker decides to perform a particular action.

An intrusion will therefore result from an action which has been successful in exploiting a vulnerability. Assume that i is a good (but vulnerable) system state and that j is a failed system state. To formalize the idea of an attacker's decision, we define $\pi_i(a)$ as the probability that an attacker will choose action a when the system is in state i. In a low level system abstraction model, the successful intrusion will cause a transition of the system state, from the good state i to the failed state j. In this paper we model all the expected failure times as negatively exponentially distributed. This is primarily to simplify mathematical analysis of the system. In reality, other types of distributions may be more suitable. Define $\lambda_{ij}(a)$ as the accumulated failure intensity if all potential attackers always take action a. Hence, the *failure rate* between state i and j may be computed as $q_{ij} = \pi_i(a)\lambda_{ij}(a)$. This is illustrated in Fig. 1 where the good state $i = 1$ is depicted as a circle and the failed state $j = 2$ as a square.

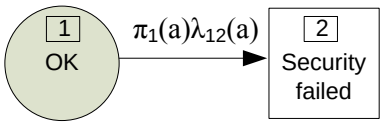


Figure 1. A two-state Markov model with assigned failure rate.

By introducing the attack probability $\pi_i(a)$, the result from a successful intrusion can be modeled as one or more *intentional state changes* of the underlying stochastic process, which represents the dynamic behavior of the system. The adopted method for computing the attack probabilities will be explained in Section 4.

In contrast to attack graphs, as used in e.g. [7], where each state transition corresponds to a single atomic step of a penetration, our model aims to be more high-level

and focus on the *impact* of the intrusions on the system rather than on the specific attack procedures themselves. This facilitates the modeling of unknown attacks in terms of generic state transitions. For example, in the stochastic model depicted in Fig. 1 the attack a can simply be explained as "the action that seeks to transfer the system from the good state 1 to the failed state 2".

During the modeling process, the granulation of the state space needs to be carefully considered. Too simple models (as the one in Fig. 1) will not provide any valuable insight into the system behavior, whereas too complex models may quickly lead to state space explosion. The choice of what to include in the states definition will therefore be a trade-off between model representativeness and complexity. An example, primary for illustration purposes, will be provided in Section 6.

3. Obtaining System Measures

This section formalizes the ideas discussed in the precious section, and explains how the stochastic model can be used to predict system security and dependability measures.

3.1 System Equations

In mathematical terms, the stochastic process describing the dynamic system behavior is a continuous time Markov chain (CTMC) with discrete state space $\mathbf{S} = \{S_1, \ldots, S_N\}$. Let

$$\mathbf{X}(t) = \{X_1(t), \ldots, X_N(t)\}, \tag{1}$$

where $X_i(t)$ denotes the probability that the system is in state i at time t. Formally, the interactions between the states $i = 1, \ldots, N$ are described in the $N \times N$ state-transition rate matrix $\mathbf{Q}$, whose elements are

$$q_{ij} = \begin{cases} \lim_{dt \to 0} \left\{ \frac{\Pr(\text{transition from } i \text{ to } j \text{ in}(t,t+dt))}{dt} \right\}, & i \neq j \\ -\sum_{j \neq i} q_{ij}, & i = j \end{cases}. \tag{2}$$

The element $q_{ij} \in \mathbf{Q}$, $(i \neq j)$, represents the transition rate between state i and j in the model and is, if the transition is caused by an intrusion, constructed from an attack probability and intensity, as explained in Section 2.2. If the initial state of the system, i.e. $\mathbf{X}(0)$, is known, the state equation can be solved. Then

$$\mathbf{X}(t) = \mathbf{X}(0)\exp(\mathbf{Q}t). \tag{3}$$

The solution to this equation provides the transient state probabilities for a system. However, it is common to assume that the system is in steady state when analyzed. The probability that a CTMC will be in state i at time t often converges to a limiting value, which is independent of the initial state. The steady state probabilities

$$\mathbf{X} = \{X_1, \ldots, X_N\}, \tag{4}$$

where $X_i = \lim_{t\to\infty} X_i(t), i = 1, \ldots, N$, can then be obtained by solving the set of N equations given by $N-1$ of the N equations

$$\mathbf{XQ} = \mathbf{0}, \tag{5}$$

and with the N'th equation

$$\sum_{l=1}^{N} X_l = 1. \tag{6}$$

The steady state probabilities provide us with the possibility of obtaining operational measures of the system, such as the mean between failures ($MTBF$) or the mean time spent in the good states (MUT). See e.g. [8] for details. To compute the mean time to failure ($MTTF$) and the mean time to first failure ($MTFF$) for a system we adopt the approach described in [9]. Assume that the state set can be partitioned as $\mathbf{S} = \{\mathbf{S}_G, \mathbf{S}_F\}$, where $\mathbf{S}_G = \{S_1, \ldots, S_K\}$ and $\mathbf{S}_F = \{S_{K+1}, \ldots, S_N\}$, so that the states $1, \ldots, K$ are good states and the states $K+1, \ldots, N$ are failed states. Since the state set $\mathbf{S}$ is ordered, the $\mathbf{Q}$ matrix can be written i partitioned form as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \\ \mathbf{Q}_3 & \mathbf{Q}_4 \end{bmatrix}, \tag{7}$$

where the size of $\mathbf{Q}_1$ is $K \times K$, the size of $\mathbf{Q}_2$ is $K \times (N-K)$ and so forth. To compute the $MTFF$ one assumes that the system is new at $t = 0$, i.e. the initial state is known by certainty, and it is known to be good. Define $\mathbf{T} = \{T_1, \ldots T_K\}$. By solving

$$-\mathbf{TQ}_1 = \{1, 0, \ldots, 0\} \tag{8}$$

the mean time to first failure for the system can be computed as

$$MTFF = \sum_{i=1}^{K} T_i. \tag{9}$$

To compute the $MTTF$ the steady state probabilities in (4) must be known. Since $\mathbf{S}$ is partitioned, also $\mathbf{X}$ can be partitioned as $\mathbf{X} = \{\mathbf{X}_G, \mathbf{X}_F\}$, where $\mathbf{X}_G = \{X_1, \ldots, X_K\}$ and $\mathbf{X}_F = \{X_{K+1}, \ldots, X_N\}$. Now the system can be in any of the good states at $t = 0$, i.e. $\mathbf{X}_G(0) = \frac{\mathbf{X}_G}{\mathbf{X}_G \mathbf{h}_K}$. Hence,

$$MTTF = \mathbf{X}_G(0)(-\mathbf{Q}_1)^{-1}\mathbf{h}_K = \frac{\mathbf{X}_G(-\mathbf{Q}_1)^{-1}\mathbf{h}_K}{\mathbf{X}_G\mathbf{h}_K}, \tag{10}$$

where $\mathbf{h}_K$ is a column vector of K ones.

3.2 Model Parametrization

In order to obtain measures the stochastic model has to be parametrized, i.e. the elements $q_{ij} \in \mathbf{Q}$ need to be evaluated. The procedure of obtaining accidental failure-

and repair rates has been practiced for many years in traditional dependability analysis, and will therefore not be discussed in this paper. However, choosing the accumulated attack intensities $\lambda_{ij}(a)$'s remains a challenge. One solution is to let security experts assess the intensities based on subjective expert opinion, empirical data or a combination of both. An example of empirical data is historical attack data collected from honeypots. The data can also be based on intrusion experiments performed by, for example, students in a controlled environment. Empirical data from such an experiment conducted at Chalmers University of Technology in Sweden [10] indicates that the time between successful intrusions during the standard attack phase is exponentially distributed. Another ongoing project at the Carnegie Mellon CyLab [11] aims to collect information from a number of different sources in order to predict attacks. Even though the process of assessing the attack intensities is crucial, and an important research topic in itself, it is not the primary focus of this paper.

Obtaining realistic $\pi_i(a)$'s, i.e. the probabilities that an attacker chooses particular attack actions in certain system states, may be more difficult. In this paper we use *game theory* as a means for computing the expected attacker behavior. The procedure is summarized in the next section.

4. Predicting Attacker Behavior

Game theory is an approach frequently used for human behavior prediction in e.g. economics, political science and sociology. This section demonstrates how a two-player zero-sum stochastic game [12] can be used to compute the expected attacker behavior, in terms of a set of attack probability vectors $\pi = \{\pi_i\}$. The procedure contains five main steps:

1 Identify the game elements.

2 Construct the action sets.

3 Assign the outcome values.

4 Compute the transition probabilities.

5 Solve the game.

Formally, the game we use is a tuple $(\Gamma, A, D, \gamma, p)$, where $\Gamma = \{\Gamma_i\}$ is a state set, $A = \{a\}$ and $D = \{d\}$ are action sets, $\gamma : \Gamma \times A \times D \rightarrow \mathcal{R}$ is an outcome function and $p : \Gamma \times A \times D \times \Gamma \rightarrow [0, 1]$ is a state transition probability function.

In the context of attack prediction for security and dependability assessment, the game is played by an attacker versus the system. (In fact, the attacker's real counterplayer in the game is the system's IDS mechanisms, for simplicity referred to as "the system" hereafter.) Even though in real-life there may be numerous attackers attacking the system, simultaneously and independent of each other, a two-player game model is sufficient to predict their individual behavior, provided that they possess similar motives and skills. In contrast to previous research in the field of network security and game theory (see Section 7) we view the game entirely from an attacker's

perspective. The purpose of our game model is to predict the behavior of attackers and not to perform any cost-benefit optimization of system defense strategies. We therefore assume that the set of system IDS mechanisms are fixed and do not change over time. Since the game is zero-sum, one player's gain will be the other player's loss. Hence, we do not need to specify separate outcome values for the system itself, as was done in [13] and [14], it is sufficient to assign the attackers' outcome values. The main benefit of our approach is that it does not assume that the attackers know the system outcome values. Moreover, it reduces the number of parameters in the system evaluation model that has to be assessed.

To compute the expected attacker behavior by means of a stochastic game, the five-step procedure is as follows.

4.1 Step 1: Identify the Game Elements.

The first step is to identify the game elements. From the stochastic model, pick all states in **S** where the system is vulnerable to intrusions. Each of these states can be viewed as a game element Γ_i in a two-player zero-sum stochastic game with state set Γ. For example, in Fig. 2 the shaded states V, L and IS represent states where the system is vulnerable. Hence, the set of game elements for this model is $\Gamma = \{\Gamma_V, \Gamma_L, \Gamma_{IS}\}$.

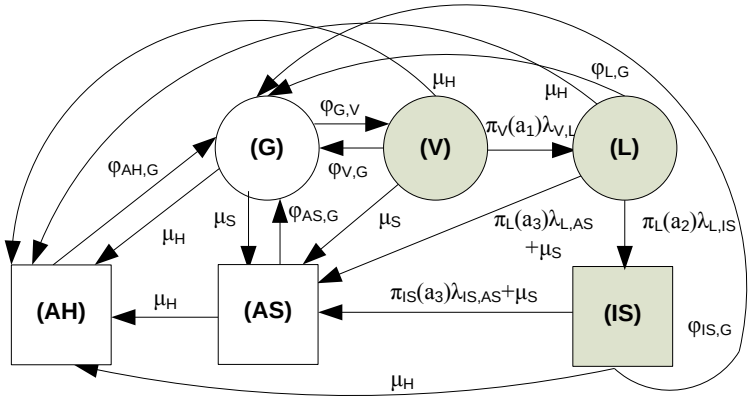


Figure 2. State transition model of DNS server (cf. Section 6) with game elements identified.

Note that even though the system state space **S** may be very large, the corresponding set with game elements Γ will (in most cases) contain only a subset of all the states in **S**, as the example indicates.

4.2 Step 2: Construct the Action Sets.

The next step is to construct the action sets A and D. The set A consists of all possible attack actions. For all transitions out of the game element states, which represent intrusions, identify the corresponding attack actions. Note that A must also contain an "inaction", which we will denote by ϕ, to represent that an attacker may

not take any action at all. We use $A_i = \{a_1, \ldots, a_m\}$ to refer to the set of actions available in game state i. All actions will not necessarily be available in all states, i.e. $A_i \subseteq A$, however $A_i \cap \phi = \phi$. For instance, in Fig. 2 the complete action set is $A = \{a_1, a_2, a_3, \phi\}$, whereof $A_V = \{a_1, \phi\}$, $A_L = \{a_2, a_3, \phi\}$ and $A_{IS} = \{a_3, \phi\}$.

Let π_i be the probability distribution over the action set A_i. In a game theoretic context $\pi_i = (\pi_i(a_1), \ldots, \pi_i(a_m))$ is called the *attack strategy* of Γ_i. Hence, $\pi_i(a_k) \in \pi_i$ will be the probability that an attacker chooses action a_k when the system is in state i, as previously discussed. One must have $\sum_{a_k} \pi_i(a_k) = 1, \forall \Gamma_i \in \Gamma$. The attack probability vectors π_i will represent the degree of hostility in the network environment, or equivalently, the aggressiveness of the attackers targeting the system. The smaller $\pi_i(a_k)$, the less the probability of the particular attack a_k in system state i and, hence, the smaller the corresponding failure rate will be.

The set D consists of all possible defense actions, whereof $D_i = \{d_1, \ldots, d_m\}$ is the set of actions available in state i. The system *defense strategy* of Γ_i is $\theta_i = (\theta_i(d_1), \ldots, \theta_i(d_m))$. Hence, $\theta_i(d_k) \in \theta_i$ is the probability that an IDS alarm indicating action a_k will be triggered in system state i. As for A_i, also D_i must contain an ϕ element, since there may not be any reaction at all from the system. Also $\sum_{d_k} \theta_i(d_k) = 1, \forall \Gamma_i \in \Gamma$.

4.3 Step 3: Assign the Outcome Values.

To model the attackers' motivation we make use of a reward- and cost concept. The term cost is used to refer to a negative reward. By an outcome of a game element is meant a possible consequence of a play of the game, as experienced by an attacker. For each game element Γ_i, we assign an outcome value to each attack action and response pair (a_k, d_l). These values will be denoted r_{kl}, or c_{kl}, depending on whether the outcome represents a reward or cost.

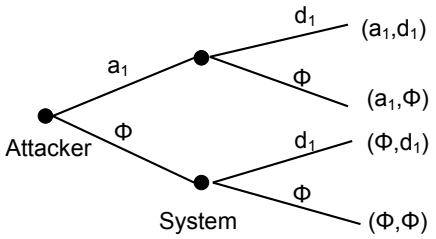


Figure 3. The possible outcomes from game element Γ_V.

The possible outcomes from game element Γ_V in Fig. 2 is depicted in Fig. 3. Since an attacker has two actions to choose between, $A_V = \{a_1, \phi\}$, and there are two possible response actions, $D_V = \{d_1, \phi\}$, there are four possible outcomes from that particular play. It could be argued that since nothing happens if the attacker does not take any action, the outcomes from the action pairs (ϕ, d_1) and (ϕ, ϕ) do not make any sense from an attacker's point of view. We counter this by pointing out that what we aim to compute from Γ_V is the expected attacker behavior in terms of strategy

$\pi_V = (\pi_V(a_1), \pi_V(\phi))$, which an attacker decides to adopt before the attack. So, if we assign a reward to the action pair (ϕ, d_1) it implies that if the attacker decides not to attack the system, no matter what, and that all the attacks will always be detected, then the attacker will experience this outcome as a gain: "It's good that I didn't try to attack the system, since I would have been detected if I did". This will be the case even though the system never gets a chance to actually detect any attack. The same line of reasoning is valid for the (ϕ, ϕ) outcome.

Reward and cost are generic concepts, which can be used to quantify the payoff of the actions both in terms of abstract values, such as social status and satisfaction versus disrespect and disappointment, as well as real values such as financial gain and loss. For instance, in [13] the reward of a successful attack action is the expected amount of recovery effort required from the system administrator and in [15] the reward is the degree of bandwidth occupied by a DDoS attack. In contrast to [13, 15], we use the cost values in the game model to represent the fact that risk averse attackers may sometimes refrain from certain attack actions due to the possible consequences of detection. Note that the outcome values themselves are not important, it is their size relatively to each other that will affect the expected attacker behavior. This topic will be further discussed in Section 5.

4.4 Step 4: Compute the Transition Probabilities.

Given that an attack action is chosen in state i, and that the intrusion is successful and remains undetected, the system may transfer to another state j where the game can continue. The transition probability between game element Γ_i and Γ_j, denoted $p_{ij}(a_k, d_l)$, can be computed by conditioning on the chosen action a_k and the system response d_l. For example, if the system in Fig. 2 is in state V and an attacker decides to attack the system, and the action remains undetected, then $\pi_V\big(a_1|(a_1, \phi)\big) = 1$. It is obtained from the Markov properties of the system that the probability of going from state V to L becomes

$$p_{VL}(a_1, \phi) = \frac{\lambda_{VL}}{\lambda_{VL} + \varphi_{VG} + \mu_S + \mu_H}. \tag{11}$$

Here, φ_{VG}, μ_S and μ_H are the rates of the competing events, which may disturb the attack. Hence, (11) is the probability that the game will continue in state L. Note that one must have $p_{ij}(a_k, d_l) \geq 0$ and $\sum_j p_{ij}(a_k, d_l) < 1, \forall \Gamma_i \in \Gamma$.

Recall that A_i and D_i are the action sets associated with state i. The possible outcomes of each game element Γ_i can now be represented by a $|A_i| \times |D_i|$ matrix, which has the form

$$\Gamma_i = \begin{array}{c|c|c|c} & d_1 & \dots & d_m \\ \hline a_1 & \gamma_{11} & \dots & \gamma_{1m} \\ \hline \vdots & \vdots & & \vdots \\ \hline a_m & \gamma_{m1} & \dots & \gamma_{mm} \end{array}, \tag{12}$$

where γ_{kl} is the total outcome associated with the action pair (a_k, d_l). The entries in (12), representing state i, are of the form

$$\gamma_{kl} = \begin{cases} r_{kl} + \sum_j p_{ij}(a_k, d_l)\Gamma_j & \text{for successful attacks,} \\ c_{kl} & \text{otherwise,} \end{cases} \tag{13}$$

for which $r_{kl} \geq 0$ and $c_{kl} \leq 0$. When solving the game, the Γ_j element in (13) will be replaced by a value, as explained in the next subsection. The first case in (13) applies if the outcome represents a successful and undetected attack action. The attacker receives an immediate reward r_{kl} and there is also a possibility of future rewards, since the system may move to another game state. The second case normally applies if an attack action is detected, but can also apply if an attacker resigns even though any of the possible attacks would have been undetected. The attacker receives a cost c_{kl}. Implicitly, the formulation in (13) means that the game will end if an attack is detected and reacted to, if the attacker resigns, or if the system does not transfer to another game state, which will happen with probability $1 - \sum_j p_{ij}(a_k, d_l)$.

4.5 Step 5: Solve the Game.

The last step is to solve the game. By solving is meant to compute the best strategies for the players who participate in the game. Our model relies on the basic assumption of game theory, which states that a rational player will always try to maximize his own reward. For each system state i, which is modeled as a game element Γ_i, we can therefore expect an attacker to behave in accordance with the probability distribution $\pi_i = (\pi_i(a_1), \ldots, \pi_i(a_m))$ that maximizes $E(\pi_i, \theta_i)$, where

$$E(\pi_i, \theta_i) = \sum_{\forall a_k \in A_i} \sum_{\forall d_l \in D_i} \pi_i(a_k)\theta_i(d_l)\gamma_{kl}. \tag{14}$$

Recall that we use zero-sum game elements to model the interactions between an attacker and the system. An attacker who does not know the defense strategy θ_i will therefore think of the system as a counter-player in the game who tries to minimize the attacker's reward. Hence, the optimal attack strategy of Γ_i, and its corresponding defense strategy, are obtained by solving

$$\max_{\pi_i} \min_{\theta_i} E(\pi_i, \theta_i). \tag{15}$$

These strategies will be denoted π_i^* and θ_i^*, respectively. The *value* of game element Γ_i, denoted $V(i)$, is defined as the expected outcome when π_i^* and θ_i^* are used, i.e.

$$V(i) = E(\pi_i^*, \theta_i^*). \tag{16}$$

The purpose of the stochastic game model is to predict the complete set of attack probability vectors $\pi^* = \{\pi_i^*\}$ to be used in the system rate matrix $\mathbf{Q}$. To find the π_i^* strategies for all game elements in the stochastic game, one can use Alg. 1, which is based on the Shapley algorithm [16]. The functions Value$[\Gamma_i]$ and Solve$[\Gamma_i]$ refer to

standard algorithms for solving zero-sum matrix games by linear programming. The former returns the expected value in (16) when the attacker and the system use their optimal strategies, whereas the latter returns the attacker's optimal strategy itself as resulting from (15). Note that Alg. 1 replaces the game element Γ_j in (13) with its value component $V(j)$ iteratively when solving the stochastic game.

Algorithm 1 Compute expected attacker strategy

IN: $(\Gamma, A, D, \gamma, p)$ {a stochastic game}
OUT: π^* {the optimal attack strategy}
  Initialize the value vector $V = \{V(i)\}$ arbitrarily
  repeat
    for each game element $\Gamma_i \in \Gamma$ **do**
      for all γ_{kl} **do**
        replace all Γ_j in (13) with $V(j)$
      end for
      compute the matrix $\Gamma_i(V) = [\gamma_{kl}]$,
    end for
    for each game element $\Gamma_i \in \Gamma$ **do**
      update the value vector $V(i) \leftarrow \text{Value}[\Gamma_i(V)]$
    end for
  until $V(i) = \text{Value}[\Gamma_i(V)], \forall \Gamma_i \in \Gamma$
  for each game element $\Gamma_i \in \Gamma$ **do**
    $\pi_i^* \leftarrow \text{Solve}[\Gamma_i(V)]$
  end for
  return the set of equilibrium vectors $\pi^* = \{\pi_i^*\}$

We believe that the optimal attack strategy set $\pi^* = \{\pi_i^*\}$ will be a good indication of the expected attack probabilities for the vulnerable system states. This is because π^* gives a lower-bound on the attacker outcome, regardless of the system defense strategy. When following π^* the attacker has no reason to change strategy; the *no-regrets property* of game theory. This property means that the attacker has maximized his expected outcome from the attack, regardless if his actions are successful or not. Several experienced indicates that this search for guarantees is a very strong motivation of human behavior. Assuming that the attacker population targeting the system will make rational choices relative to their objectives, their collected behavior will, in the long run, gravitate towards the optimal attack strategy [17]. For further details on the underlying assumptions and solution of the stochastic game model, the reader is referred to [12, pp. 96–101].

5. Attacker Profiling

To distinguish between different types of attackers, it is common practice to make use of attacker profiles. A number of fine-granular classifications of attackers exist in the literature. In [18] Rogers summarizes earlier research on attacker categorization and provides a new taxonomy based on a two-dimensional circumflex classification

model. *Skill* and *motivation* are identified as the primary classification criteria, which fit well into our mathematical framework where the attacker skill is represented by attack intensities and the motivation by the reward- and cost concept. The advantage of Roger's circumflex approach is that it does not rely on any hard categorization model, but can rather serve as a basis when defining attacker profiles that share similar characteristics. Hence, to comply with the model in [18] we suggest *tuning*, of both the reward- and cost values of the game elements as well as the attack intensities in the stochastic model, to model the motivation and skill of the particular kind of attackers that are considered in the system's threat environment. The effect of tuning the attack intensities is straight-forward to explain; by raising the attack intensity values, the corresponding failure rates will increase. However, the influence of the outcome values in the game model is not as obvious. This section will therefore demonstrate the tuning of the game parameters.

5.1 Tuning the Game Parameters

The stochastic game model presented in the previous section is based on a reward- and cost concept. These values will represent the attackers' motivation when deciding on attack actions. Whenever an attacker performs an attack action, he immediately receives a reward. Furthermore, if the action succeeds, additional rewards may be gained. We use negative rewards, i.e. costs, to make room for the possibility that some attackers may be more risk averse than others. The cost of a detected action will be an important demotivating factor when modeling, for example, insiders; legitimate users who override their current privileges. Similarly, commercial adversaries would lose reputation and market share if it is exposed that illegal means are used.

Since we have chosen to model the interactions between an attacker and the system as a zero-sum game rather than a general-sum one, an increasing cost value will play a deterrent role for an attacker. However, due to the inherent properties of the minimax solution in (15), also an increasing reward value will indirectly play a deterrent role for an attacker. One must therefore vary the cost parameters rather than the reward parameters in order to get an intuitive corresponding attack strategy. This process will be further illustrated in the upcoming examples.

In (13) we set $r_{kl} = 1$ and $p_{ij}(a_k, d_l) = 0$, $\forall j, k, l$, and then let the cost value vary between $-10 \leq c_{kl} \leq 0$. This provides us with the possibility of analyzing how the cost of a detected attack versus the reward of an undetected one will affect the expected attacker behavior for a particular system state i.

5.2 One Possible Attack Action

As a first example, assume that a system is vulnerable to a single attack action in state i. An attacker can choose either to perform the attack (action a), or to resign (action ϕ). The system's response actions are to either set an alarm (action d) or no reaction (action ϕ). Hence, $A_i = \{a, \phi\}$ and $D_i = \{\phi, d\}$. To model this scenario

we use the 2×2 game element

$$\Gamma_i = \begin{array}{c|c|c} & \phi & d \\ \hline a & \gamma_{a\phi} & \gamma_{ad} \\ \hline \phi & \gamma_{\phi\phi} & \gamma_{\phi d} \end{array} = \begin{array}{c|c|c} & \phi & d \\ \hline a & 1 & b \\ \hline \phi & c & 0 \end{array}, \tag{17}$$

where the cost value b represents an attacker's cost of a detected action and c the cost of resigning, even though an attempted attack would have been undetected. By varying b and c we can now demonstrate how the relation $\gamma_{ad}/\gamma_{a\phi}$ (i.e. the cost of a detected attack versus the reward of an undetected attack) and $\gamma_{\phi\phi}/\gamma_{a\phi}$ (i.e. the cost associated with resigning versus the reward of an undetected attack) will affect the attackers' expected behavior, in terms of the attack probability $\pi_i^*(a)$. To compute $\pi_i^* = (\pi_i^*(a), \pi_i^*(\phi))$ we solve (15), as previously discussed.

Reducing b

If $b = -2$ and $c = -3$ in (17), then the expected probability of attacking will be $\pi_i^*(a) = 0.50$. However, if the cost of a detected action is increased to $b = -10$, then $\pi_i^*(a) = 0.21$. Hence, an increasing cost of a detected action will decrease the attackers' motivation.

Reducing c

Again, if $b = -2$ and $c = -3$ in (17), then $\pi_i^*(a) = 0.50$. However, if $c = -10$, then $\pi_i^*(a) = 0.77$. As the cost of resigning increases, the attackers' motivation will increase.

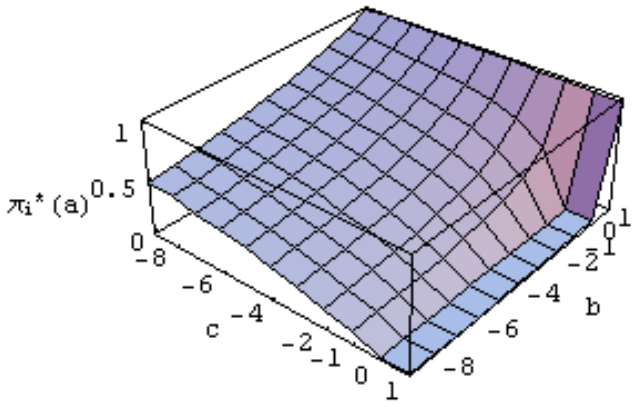


Figure 4. The expected attacker behavior $\pi_i^*(a)$ w.r.t. b and c.

Fig. 4 depicts a more complete graph of risk averse attackers' expected behavior. In the graph we let $-9 \leq b, c \leq 1$. One can see that the expected probability of attacking is highest, $\pi_i^*(a) = 1.0$, when $b = 1$. This is intuitive since an attacker who receives the same reward whether he is detected or not will always choose to attack. On the other hand, the expected probability of attacking is lowest, $\pi_i^*(a) = 0.0$, when

$c > 0$ and $b < 0$. This can be interpreted as if the reward of an attack is small enough, so that it is not significantly greater than the cost of resigning, an attacker may not even bother to try (of course this is an ideal situation unlikely to occur in real life). In general, as the examples indicate and the graph illustrates, as the cost values increase we can expect attackers to act more carefully.

It is interesting to note that even though measures are taken to increase the cost of detected actions, legal proceedings for instance, a rapidly decreasing b will only have marginal effect on the behavior of an attacker who has a strong reluctance of resigning. This is shown in the graph as a slowly decreasing $\pi_i^*(a)$ along the "$c = -9$"-axis. In fact, the parameter that has the strongest influence on the expected attacker behavior w.r.t. (17) is c. Unfortunately, since c represents a mental factor in this game (the attackers' reluctance to resign) it will be difficult for a system administrator to take preventive measures influencing c in a way that will reduce $\pi_i^*(a)$.

5.3 More Attack Actions

The same methodology can also be used to compute the expected attacker behavior for states where a system is vulnerable to a large number of attack actions. An $(n + 1) \times (n + 1)$ game element (n possible attack actions) might look like

$$\Gamma_i = \begin{array}{c|c|c|c|c} & \phi & d_1 & \dots & d_n \\ \hline a_1 & 1 & b1 & \dots & 1 \\ \hline \vdots & \vdots & & & \vdots \\ \hline a_n & 1 & 1 & \dots & bn \\ \hline \phi & c & 0 & \dots & 0 \end{array}, \tag{18}$$

For example, if n = 4 and we choose the cost values as $b1$=-3, $b2$=-4, $b3$=-8, $b4$=-6 and c=-1, then

$$\pi_i^* = (\pi_i^*(a_1), \pi_i^*(a_2), \pi_i^*(a_3), \pi_i^*(a_4), \pi_i^*(\phi)) = (0.14, 0.12, 0.07, 0.08, 0.54). \tag{19}$$

However, if $b1$ and $b2$ is increased to -4 and -8 respectively, then

$$\pi_i^* = (0.13, 0.07, 0.07, 0.09, 0.64). \tag{20}$$

One can see that also for larger games, an increasing cost of a detected action will lead to a smaller probability of an attacker choosing that particular action.

A more detailed look of how the expected attack probabilities depend on the cost values is depicted in Fig. 5-6. In Fig. 5 one can see how a particular attack probability is affected when varying the different cost values. For example, the upper left graph indicates that $\pi_i^*(a_1)$ is at its highest when $b1 \to 0$. Fig. 6 depicts how the elements in the attack probability vector π_i^* depend on a particular cost value. For example, the lower right graph shows that $\pi_i^*(\phi)$ is high when $b4 < -2$ and that $\pi_i^*(a_4)$ rises as $b4 \to 0$. Note that $\sum_{a_k \in A_i} \pi_i^*(a_k) = 1$.

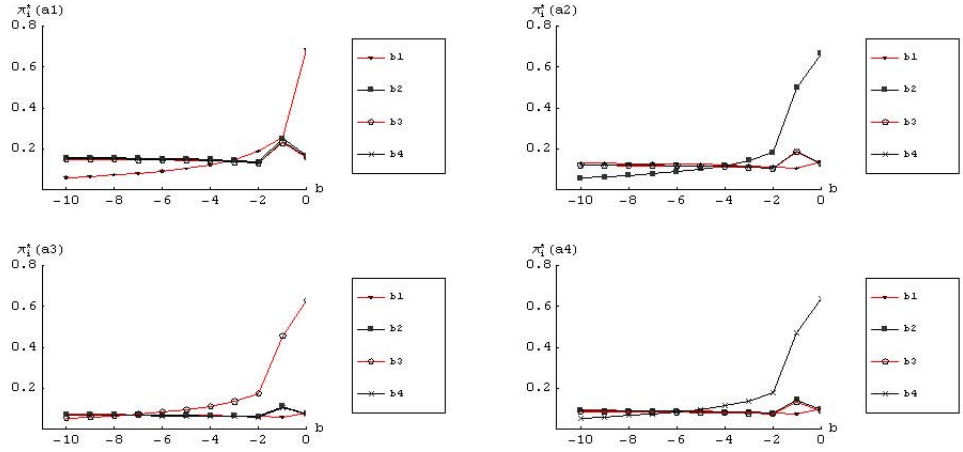


Figure 5. A particular attack probability as a function of the cost values.

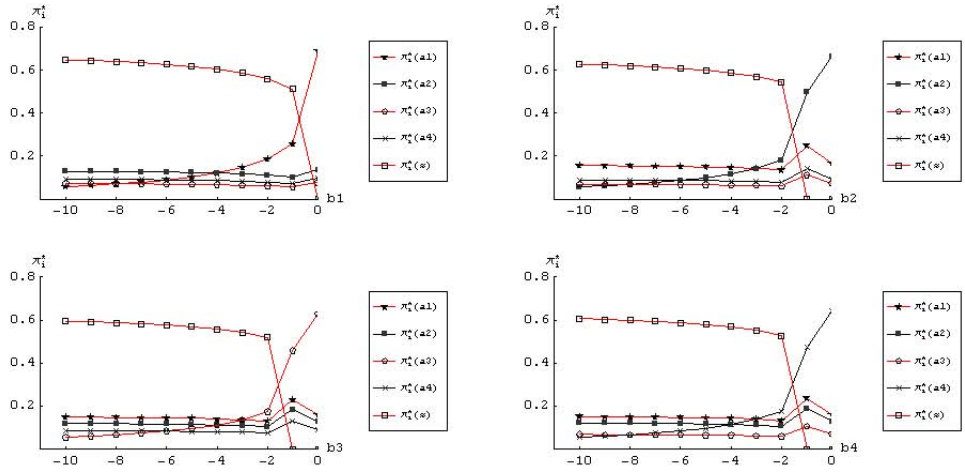


Figure 6. Attack probability vectors as a function of a particular cost value.

6. Case Study: The DNS Service

To further illustrate the approach, we model and analyze the security and dependability of a DNS service. The Domain Name System (DNS) provides a critical service to the Internet - the mapping between names and addresses. The most important attributes of this service are availability and integrity; the service should be there when the clients need it, and it must provide correct replies to DNS request. We distinguish between two different types of accidental failure modes; hardware availability failures (AH), which require a manual repair, and software availability failures (AS), which only require a system reconfiguration and/or reboot. Unfortunately, buffer

overflow vulnerabilities are common in multiple implementations of DNS resolver libraries. During its operational lifetime, the server will be subject to manual maintenance, upgrades and reconfigurations. Humans frequently make mistakes. It is therefore realistic to assume that the system will naturally alternate between a good state (G) where it is secure against these types of attacks and another good, but vulnerable, state (V) where buffer overflow attacks are possible. When the system is in the vulnerable state, an attacker who can send malicious DNS requests might exploit such a vulnerability to gain access to the server. This may transfer the system into a third state (L), from where it is possible to insert false entries in the server cache; software integrity failure (IS), or to shut the server down; software availability failure (AS). In this case, all the three states G, V and L are considered to be good states. Even though the system is erroneous in states V and L, it still manages to deliver the intended service, i.e. to provide clients with correct replies to DNS requests. Hence, the system state set is $\mathbf{S} = \{G, V, L, IS, AS, AH\}$, whereof $\mathbf{S}_G = \{G, V, L\}$ and $\mathbf{S}_F = \{IS, AS, AH\}$.

The state transition model in Fig. 2 in Section 4 depicts the security and dependability behavior of a single DNS server under the given assumptions. The transitions labeled with the μ_S and μ_H rates represent the accidental software- and hardware failures, the φ rates represent the system administrator's possible actions and the λ rates represent the intensities of the possible attack actions. The game elements are the shaded states in the figure: $\Gamma = \{\Gamma_V, \Gamma_L, \Gamma_{IS}\}$. The attack action set in the stochastic game is $A = \{a_1, a_2, a_3, \phi\} = \{$*"illegal login"*, *"cache poisoning"*, *"server shut down"*, *"do nothing"*$\}$ and the defense action set is the corresponding $D = \{d_1, d_2, d_3, \phi\}$. Using the rate values $\lambda_{V,L} = 1/3$, $\lambda_{L,IS} = \lambda_{L,AS} = \lambda_{IS,AS} = 3$, $\varphi_{G,V} = 1/480$, $\varphi_{V,G} = 1/120$, $\varphi_{L,G} = \varphi_{IS,G} = 1$, $\varphi_{AS,G} = 3$, $\varphi_{AH,G} = 1/24$, $\mu_H = 1/3600$ and $\mu_S = 1/120$ (per hour) the game elements become

$$\Gamma_V = \begin{array}{c|cc} & \phi & d_1 \\ \hline a_1 & r_{a_1,\phi} + 0.952\Gamma_3 & c_{a_1,d_1} \\ \phi & c_{\phi,\phi} & 0 \end{array},$$

$$\Gamma_L = \begin{array}{c|ccc} & \phi & d_2 & d_3 \\ \hline a_2 & r_{a_2,\phi} + 0.748\Gamma_4 & c_{a_2,d_2} & 0 \\ a_3 & r_{a_3,\phi} & 0 & c_{a_3,d_3}s \\ \phi & c_{\phi,\phi} & 0 & 0 \end{array}, \tag{21}$$

$$\Gamma_{IS} = \begin{array}{c|cc} & \phi & d_3 \\ \hline a_3 & r_{a_3,\phi} & c_{a_3,d_3} \\ \phi & c_{\phi,\phi} & 0 \end{array}.$$

By using Alg. 1, the stochastic game can be solved. The optimal attack strategy vectors $\pi^* = \{\pi^*_V, \pi^*_L, \pi^*_{IS}\}$ will then be used in the state transition rate matrix for the DNS server when predicting system measures. The rate matrix $\mathbf{Q}$ is displayed in Table 1. To illustrate the effect of the reward and cost values on the predicted system measures, we perform the computations for four different scenarios. Note that all numerical values (reward- and costs as well as failure rates) are chosen for illustration purposes only.

$$\mathbf{Q} = \begin{pmatrix} -(\varphi_{G,V} + \mu_S + \mu_H) & \varphi_{G,V} & 0 & 0 & \mu_S & \mu_H \\ \varphi_{V,G} & q_{V,V} & \pi_V(a_1)\lambda_{V,L} & 0 & \mu_S & \mu_H \\ \varphi_{L,G} & 0 & q_{L,L} & \pi_L(a_2)\lambda_{L,IS} & \mu_S + \pi_L(a_3)\lambda_{L,AS} & \mu_H \\ \varphi_{IS,G} & 0 & 0 & q_{IS,IS} & \mu_S + \pi_{IS}(a_3)\lambda_{IS,AS} & \mu_H \\ \varphi_{AS,G} & 0 & 0 & 0 & -(\varphi_{AS,G} + \mu_H) & \mu_H \\ \varphi_{AH,G} & 0 & 0 & 0 & 0 & -\varphi_{AH,G} \end{pmatrix}$$

$$= \begin{pmatrix} -1.07 \cdot 10^{-2} & 2.08 \cdot 10^{-3} & 0 & 0 & 8.33 \cdot 10^{-3} & 2.78 \cdot 10^{-4} \\ 8.33 \cdot 10^{-3} & q_{V,V} & 0.33 \cdot \pi_V(a_1) & 0 & 8.33 \cdot 10^{-3} & 2.78 \cdot 10^{-4} \\ 1 & 0 & q_{L,L} & 3 \cdot \pi_L(a_2) & 8.33 \cdot 10^{-3} + 3 \cdot \pi_L(a_3) & 2.78 \cdot 10^{-4} \\ 1 & 0 & 0 & q_{IS,IS} & 8.33 \cdot 10^{-3} + 3 \cdot \pi_{IS}(a_3) & 2.78 \cdot 10^{-4} \\ 3 & 0 & 0 & 0 & -3.89 & 2.78 \cdot 10^{-4} \\ 4.17 \cdot 10^{-2} & 0 & 0 & 0 & 0 & -4.17 \cdot 10^{-2} \end{pmatrix}$$

Table 1. The state transition rate matrix for the DNS server (rate values in matrix reduced to three significant numbers). To increase readability, $q_{V,V} = -(\varphi_{V,G} + \pi_V(a_1)\lambda_{V,L} + \mu_S + \mu_H) = -1.69 \cdot 10^{-2} - 0.33 \cdot \pi_V(a_1)$, $q_{L,L} = -(\varphi_{L,G} + \pi_L(a_2)\lambda_{L,IS} + \mu_S + \pi_L(a_3)\lambda_{L,AS} + \mu_H) = -1.0086 - 3 \cdot \pi_L(a_2) - 3 \cdot \pi_L(a_3)$ and $q_{IS,IS} = -(\varphi_{IS,G} + \mu_S + \pi_{IS}(a_3)\lambda_{IS,AS} + \mu_H) = -1.0086 - 3 \cdot \pi_{IS}(a_3)$ have been suppressed in the matrix.

6.1 Case 1: The Worst-case Scenario

First we look at the "worst-case" scenario when all attackers always try all possible attacks, i.e. $\pi_i(a_k) = 1, \forall i, k$ in $\mathbf{Q}$. In this case we do not use the game model to compute the expected attacker behavior. Using (5) and (6) in Section 3 we compute the steady state probabilities for the DNS server as $\mathbf{X} = \{X_G, X_V, X_L, X_{IS}, X_{AS}, X_{AH}\} = \{0.984, 5.85 \cdot 10^{-3}, 2, 78 \cdot 10^{-3}, 2.08 \cdot 10^{-4}, 3.24 \cdot 10^{-3}, 6.62 \cdot 10^{-3}\}$. Hence, by using (9) and (10) we obtain the mean time to first failure $MTFF = 97.11$ (h) and the mean time to failure $MTTF = 96.62$ (h) for the DNS server.

6.2 Case 2: Risk Averse Attackers

Now assume that the attackers will take into account the possible consequences of their actions. We use the set of reward- and cost values $r_{a_1,\phi}$=$r_{a_2,\phi}$=$r_{a_3,\phi}$=1, c_{a_1,d_1}=-4, c_{a_2,d_2}=-3, c_{a_3,d_3}=-2, $c_{\phi,\phi}$=-5. Solving the stochastic game in accordance to Alg. 1 provides the optimal attack strategy vectors $\pi_V^* = (0.568, 0.432)$, $\pi_L^* = (0, 0.625, 0.375)$ and $\pi_{IS}^* = (0.625, 0.375)$. The corresponding steady state probabilities for this case become $\mathbf{X} = \{0.980, 9.89 \cdot 10^{-3}, 6.50 \cdot 10^{-4}, 0, 3.16 \cdot 10^{-3}, 6.62 \cdot 10^{-3}\}$, hence $MTFF = 101.61$ (h) and $MTTF = 100.97$ (h). Since this scenario assumes risk averse attackers, both the $MTFF$ and $MTTF$ will be slightly higher than in the worst-case scenario.

6.3 Case 3: Implementing Countermeasures

Assume that we want to evaluate the benefit of setting up a new logging and tracing mechanism for the DNS server, with the purpose of reducing the probability of illegal login attempts (action a_1). As in the previous scenario we consider risk averse attackers. All detected illegal login attempts will be recorded and prosecuted, which are modeled as an increasing cost value $c_{a_1,d_1} = -7$ in game element Γ_V. Hence,

the new expected attack strategy for state V will be $\pi_V^* = (0.394, 0.606)$. The corresponding measures are $\mathbf{X} = \{0.976, 1.37 \cdot 10^{-2}, 6.25 \cdot 10^{-4}, 0, 3.14 \cdot 10^{-3}, 6.62 \cdot 10^{-3}\}$, $MTFF = 102.11$ (h) and $MTTF = 101.26$ (h). The results show that even though $\pi_V^*(a_1)$ is decreased with 23%, the $MTFF$ and $MTTF$ are only marginally increased. As a conclusion, the DNS service will not benefit much from the new logging and tracing mechanism.

6.4 Case 4: Accidental Failures Only

Finally assume that we do not consider attacks at all, but rather model accidental failure only, i.e. $\pi_i(a_k) = 0, \forall i, k$ in $\mathbf{Q}$. The corresponding measures are $\mathbf{X} = \{0.882, 0.108, 0, 0, 2.75 \cdot 10^{-3}, 6.62 \cdot 10^{-3}\}$ and $MTFF = MTTF = 116.13$ (h). As can be seen, the system's expected time to failure will increase noticeably when attacks are not included as possible fault sources. Hence, for the actual parameterization the random failures will dominate the trustworthiness of the service.

A comparison of the $MTTF$ for the four cases are provided in Fig. 7. It should

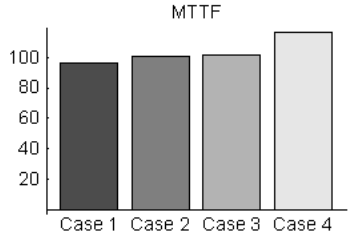


Figure 7. $MTTF$ for the four cases.

be noted that since the rate values in this example are chosen for illustration purposes only, the predicted system measures may not conform with failure times for a real-life server implementation.

7. Related Work

Security and dependability. In [2], Laprie et.al. provide a taxonomy for dependability and security, and a thorough definition of its concepts. A deliverable produced by the MAFTIA project [4] refines these concepts in the context of malicious faults and discusses how fault prevention, removal, tolerance and forecasting can be re-interpreted in a security context. Jonson et.al. [19] suggest a unified framework for integrated security and dependability assessment. The objective is to create a basis for system failure analysis, regardless if the failure is caused by an intrusion or a hardware fault. Nicol et.al. [6] provide a survey over existing dependability analysis techniques and summarizes how these are being extended to evaluate security. The terminology and concepts in this paper are built on these papers.

Stochastic models of security. Ortalo et.al. [20] present a quantitative model to measure known Unix security vulnerabilities using a privilege graph, which is transformed into a Markov chain. The model allows for the characterization of operational security expressed as the mean effort to security failure, as originally proposed by Littlewood et.al. in [21]. Further, Madan et. al. [22] use traditional stochastic modeling techniques to capture attacker behavior and the system's response to attacks and intrusions. A quantitative security analysis is carried out for the steady state behavior of the system. In [23] Stevens et. al. describe an approach for probabilistic validation of an intrusion-tolerant replication system. They provide a hierarchical model using stochastic activity nets (SAN), which can be used to validate intrusion tolerant systems and to evaluate merits of various design choices. Our modeling approach is inspired by all these paper, especially [22]. The main difference is the use of attack probabilities to integrate attacker behavior in the transition rates of our model. Moreover, we model accidental hardware and software failures, alongside with intrusions.

Game Theory. Game theory in a security related context has also been utilized in previous papers. Lye and Wing [13] use a game theoretic method to analyze the security of computer networks. The interactions between an attacker and the administrator are modeled as a two-player general-sum stochastic game for which optimal strategies (Nash Equilibrium) are computed. In [15] a preliminary framework for modeling attacker intent, objectives and strategies (AIOS) is presented. To infer AIOS a game theoretic approach is used and models for different threat environments are suggested. The game theoretic method used in this paper is heavily influenced by these models. However, in contrast to [13], we model the outcome of the game elements as the possible consequences of the attackers' actions being detected or not by the system's IDS mechanisms, and in contrast to [15] we use the same game model for different threat environments.

This paper is based on the results previously published in [24]. This extended version contains expansions of key ideas, discussions, examples, elaborations and applications.

8. Concluding Remarks

This paper presents a stochastic model for integrated security and dependability evaluation. Our modeling approach aim to consider most aspects that will affect the trustworthiness of a system, such as normal user behavior, administrative activities, random software- and hardware failures, and intentional attacks. By using stochastic game theory we can compute the expected attacker behavior for different types of attackers. The reward- and cost concept makes it possible to use the stochastic model to predict security- and dependability measures for a particular threat environment. Having solved the game, the expected attacker behavior is reflected in the transitions between states in the system model, by weighting the transition rates according to probability distributions. In the final step, the corresponding stochastic process is used to compute operational measures of the system.

The game theoretic approach deserves a few more comments. The optimal strategies have frequently been used to derive predictions of what players in a game will do [25]. As pointed out in Section 4, π^* will be an indication of the best strategy for attackers who do not know the probabilities that their actions will be detected. If the detection probabilities are known, maximizing (14) will be straightforward, hence, (15) is not applicable. Moreover, the approach is based on the underlying assumption that the attackers have a complete overview of the system states, the possible transitions between states and the existing vulnerabilities. This may not always be the case in real life. Other types of models, e.g. games with incomplete information, may therefore be more appropriate in some cases. Finally we would like to point out that modeling the attackers' interactions with the system as a zero-sum stochastic game will always provide us with a single unique solution to the game.

As indicated in the case study, there are additional features of our model than just probabilistic predictions of a system. For instance, system administrators can use our approach to answer questions such as "What is the effect of hardening security?" and "Should we perform additional monitoring?". The effect of these two countermeasures can be evaluated in our modeling and analysis framework before implementation, by changing the corresponding transition rates in the model and then comparing the results.

Currently, our model is being integrated into a framework for dynamic security and dependability assessment. The framework is based on a method for real-time risk assessment using a distributed networked agent-sensor architecture, published in [26]. By using live data from network sensors, the current state and the future behavior of the system can be predicted, which makes it possible to compute system security and dependability measures, in real time.

In the future we plan to verify the model's ability to predict real-life attacks. This will require further research, including validation of the model against empirical data.

References

[1] "ISO/IEC 13335: Information Technology - Guidelines for the management of IT Security," http://www.iso.org.

[2] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, pp. 11–33, January-March 2004.

[3] "ISO 15408: Common Criteria for Information Technology Security Evaluation," 1999, http://www.commoncriteria.org.

[4] D. Powell and R. Stroud (eds.), "Malicious- and accidental-fault tolerance for internet applications - Conceptual model and architecture," 2001. [Online]. Available: citeseer.ist.psu.edu/474556.html

[5] E. Jonsson, "Towards an integrated conceptual model of security and dependability," in *Proceedings of the First International Comference on Availability, Reliability and Security (AReS)*, 2006.

[6] D. M. Nicol, W. H. Sanders, and K. S. Trivedi, "Model-based evaluation: From dependability to security," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, pp. 48–65, January-March 2004.

[7] S. Jha, O. Sheyner, and J. Wing, "Two formal analyses of attack graphs," in *Proceedings of the 2002 Computer Security Foundations Workshop*, 2002.

[8] B. E. Helvik, *Dependable Computing Systems and Communication Networks, Design and Evaluation*, draft lecture notes (256 p.), Department of Telematics, NTNU, Nov. 2003.

[9] J. A. Buzacott, "Markov approach to finding failure times of repairable systems," *IEEE Transactions on Reliability*, vol. R-19, pp. 128–134, November 1970.

[10] E. Jonsson and T. Olovsson, "A quantitative model of the security intrusion process based on attacker behavior," *IEEE Transactions of Software Engineering*, vol. 23, no. 4, pp. 235–245, April 1997.

[11] A. Arora, "A statistical analysis of computer attacks using data from the honeynet project," Carnegie Mellon CyLab. http://www.cylab.cmu.edu/.

[12] G. Owen, *Game Theory*, 3rd ed. Academic Press, 2001.

[13] K. Lye and J. M. Wing, "Game strategies in network security," *International Journal of Information Security*, vol. 4, no. 1-2, pp. 71–86, 2005.

[14] T. Alpcan and T. Basar, "A game theoretic approach to decision and analysis in network intrusion detection," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, December 2003.

[15] P. Liu and W. Zang, "Incentive-based modeling and inference of attacker intent, objectives, and strategies," in *Proceedings of the 10th ACM conference on computer and communication security*, 2003, pp. 179–189.

[16] L. S. Shapley, "Stochastic games," *Proceedings of the National Academy of Science USA*, vol. 39, pp. 1095–1100, 1953.

[17] S. Stahl, *A Gentle Introduction to Game Theory*. American Mathematical Society, 1991.

[18] "The development of a meaningful hacker taxonomy: A two dimensional approach," CERIAS Tech Report 2005-43, Tech. Rep., 2005.

[19] E. Jonsson, L. Stromberg, and S. Lindskog, "On the functional relation between security and dependability impairments," in *Proceedings of the New Security Paradigms Workshop 1999*, Sep. 22–24 1999.

[20] R. Ortalo, Y. Deswarte, and M. Kaaniche, "Experimenting with quantitative evaluation tools for monitoring operational security," *IEEE Transactions on Software Engineering*, vol. 25, no. 5, pp. 633–650, Sept/Oct 1999.

[21] B. Littlewood, S. Brocklehurst, N. Fenton, P. Mellor, S. Page, D. Wright, J. Dobson, McDermid J., and D. Gollmann, "Towards operational measures of computer security," *Journal of Computer Security*, vol. 2, pp. 211–229, Oct 1993.

[22] K. B. B. Madan, K. V. Goseva-Popstojanova, and K. S. Trivedi, "A method for modeling and quantifying the security attributes of intrusion tolerant systems," in *Performance Evaluation*, vol. 56, 2004.

[23] F. Stevens, T. Courtney, S. Singh, A. Agbaria, J. F. Meyer, W. H. Sanders, and P. Pal, "Model-based validation of an intrusion-tolerant information system," in *Proceedings of the 23rd Symposium on Reliable Distributed Systems (SRDS 2004)*, Oct 18-20 2004.

[24] K. Sallhammar, B. E. Helvik, and S. J. Knapskog, "Towards a stochastic model for integrated security and dependability evaluation," in *Proceedings of the First International Comference on Availability, Reliability and Security (AReS)*, 2006.

[25] C. A. Holt and A. E. Roth, "The Nash equilibrium: A perspective," in *Proceedings of the National Academy of Sciences*, vol. 101, no. 12, March 23 2004.

[26] A. Årnes, K. Sallhammar, K. Haslum, T. Brekne, M. E. G. Moe, and S. J. Knapskog, "Real-time Risk Assessment with Network Sensors and Intrusion Detection Systems," in *International Conference on Computational Intelligence and Security (CIS)*, Dec 2005.

PAPER E

Real-time Risk Assessment with Network Sensors and Intrusion Detection Systems

André Årnes, Karin Sallhammar, Kjetil Haslum, Tønnes Brekne,
Marie Elisabeth Gaup Moe and Svein J. Knapskog

In Proceedings of the 2005 International Conference on Computational Intelligence and Security (CIS'05)

Xian, China, December 15-19, 2005

REAL-TIME RISK ASSESSMENT WITH NETWORK SENSORS AND INTRUSION DETECTION SYSTEMS

André Årnes, Karin Sallhammar, Kjetil Haslum, Tonnes Brekne,
Marie Elisabeth Gaup Moe, Svein Johan Knapskog
Centre for Quantifiable Quality of Service in Communication Systems
Norwegian University of Science and Technology,
Trondheim, Norway
{ andrearn,sallhamm,haslum,tonnes,marieeli,knapskog } @q2s.ntnu.no

Abstract This paper considers a real-time risk assessment method for information systems and networks based on observations from networks sensors, such as intrusion detection systems. The system risk is dynamically evaluated using hidden Markov models, providing a mechanism for handling data from sensors with different trustworthiness in terms of false positives and negatives. The method provides a higher level of abstraction for monitoring network security, suitable for risk management and intrusion response applications.

1. Introduction

Risk assessment is a central issue in management of large-scale networks. However, current risk assessment methodologies focus on manual risk analysis of networks during system design or through periodic reviews. Techniques for real-time risk assessment are scarce, and network monitoring systems and intrusion detection systems (IDS) are the typical approaches. In this paper, we present a real-time risk assessment method for large scale networks that build upon existing network monitoring and intrusion detection systems. An additional level of abstraction is added to the network monitoring process, focusing on risk rather than individual warnings and alerts. The method enables the assessment of risk both on a system-wide level, as well as for individual objects.

The main benefit of our approach is the ability to aggregate data from different sensors with different weighting according to the trustworthiness of the sensors. This focus on an aggregate risk level is deemed more suitable for network management and automated response than individual intrusion detection alerts. By using hidden Markov models (HMM), we can find the most likely state probability distribution of monitored objects, considering the trustworthiness of the IDS. We do not make any assumptions on the types of sensors used in our monitoring architecture, other

than that they are capable of providing standardized output as required by the model parameters presented in this paper.

1.1 Target Network Architecture

The target of the risk assessment described in this paper is a generic network consisting of computers, network components, services, users, etc. The network can be arbitrarily complex, with wireless ad-hoc devices as well as ubiquitous services. The network consists of entities that are either *subjects* or *objects*. Subjects are capable of performing actions on the objects. A subject can be either users or programs, whereas objects are the targets of the risk assessment. An asset may be considered an object. The unknown factors in such a network may represent vulnerabilities that can be exploited by a malicious attacker or computer program and result in unwanted incidents. The potential exploitation of a vulnerability is described as threats to assets. The *risk* of a system can be identified through the evaluation of the probability and consequence of unwanted incidents.

1.2 Monitoring and Assessment Architecture

We assume a multiagent system architecture consisting of agents that observe objects in a network using sensors. The architecture of a multiagent risk assessment system per se is not the focus of this paper, but a description is included as a context.

An *agent* is a computer program capable of a certain degree of autonomous actions. In a multiagent system, agents are capable of communicating and cooperating with other agents. In this paper, an agent is responsible for collecting and aggregating sensor data from a set of sensors that monitor a set of objects. The main task of the agent is to perform real-time risk assessment based on these data. A multiagent architecture has been chosen for its flexibility and scalability, and in order to support distributed automated response.

A *sensor* can be any information-gathering program or device, including network sniffers (using sampling or filtering), different types of intrusion detection systems (IDS), logging systems, virus detectors, honeypots, etc. The main task of the sensors is to gather information regarding the security state of objects. The assumed monitoring architecture is hybrid in the sense that it supports any type of sensor. However, it is assumed that the sensors are able to classify and send standardized observations according to the risk assessment model described in this paper.

1.3 Related Work

Risk assessment has traditionally been a manual analysis process based on a standardized framework, such as [1]. A notable example of real-time risk assessment is presented in [2], which introduces a formal model for the real time characterization of risk faced by a host. *Distributed intrusion detection systems* have been demonstrated in several prototypes and research papers, such as [3, 4]. Multiagent systems for intrusion detection, as proposed in [5] and demonstrated in e.g. [6] (an IDS prototype based on lightweight mobile agents) are of particular relevance for this paper. An im-

portant development in distributed intrusion detection is the recent IDMEF (Intrusion Detection Message Exchange Format) IETF Internet draft [7]. *Hidden Markov models* have recently been used in IDS architectures to detect multi-stage attacks [8], and as a tool to detect misuse based on operating system calls [9]. *Intrusion tolerance* is a recent research field in information security related to the field of fault tolerance in networks. The research project SITAR [10] presents a generic state transition model, similar to the model used in this paper, to describe the dynamics of intrusion tolerant systems. Probabilistic validation of intrusion tolerant systems is presented in [11].

2. Risk Assessment Model

In order to be able to perform dynamic risk assessment of a system, we formalize the distributed network sensor architecture described in the previous section. Let $O = \{o_1, o_2, \ldots\}$ be the set of objects that are monitored by an agent. This set of objects represents the part of the network that the agent is responsible for. To describe the security state of each object, we use discrete-time Markov chains. Assume that each object consisting of N states, denoted $S = \{s_1, s_2, \ldots, s_N\}$.

As the security state of an object changes over time, it will move between the states in S. The sequence of states that an object visits is denoted $X = x_1, x_2, \ldots, x_T$, where $x_t \in S$ is the state visited at time t. For the purpose of this paper, we assume that the state space can be represented by a general model consisting of three states: Good (G), Attacked (A) and Compromised (C), i.e., $S = \{G, A, C\}$. State G means that the object is up and running securely and that it is not subject to any kind of attack actions. In contrast to [10], we assume that objects always are vulnerable to attacks, even in state G. As an attack against an object is initiated, it will move to security state A. An object in state A is subject to an ongoing attack, possibly affecting its behavior with regard to security. Finally, an object enters state C if it has been successfully compromised by an attacker. An object in state C is assumed to be completely at the mercy of an attacker and subject to any kind of confidentiality, integrity and/or availability breaches.

The security observations are provided by the sensors that monitor the objects. These *observation messages* are processed by agents, and it is assumed that the messages are received or collected at *discrete time intervals*. An observation message can consist of any of the symbols $V = \{v_1, v_2, \ldots, v_M\}$. These symbols may be used to represent different types of alarms, suspect traffic patterns, entries in log data files, input from network administrators, and so on. The *sequence* of observed messages that an agent receives is denoted $Y = y_1, y_2, \ldots, y_T$, where $y_t \in V$ is the observation message received at time t. Based on the sequence of observation messages, the agent performs dynamic risk assessment. The agent will often receive observation messages from more than one sensor, and these sensors may provide different types of data, or even inconsistent data. All sensors will not be able to register all kinds of attacks, so we cannot assume that an agent is able to resolve the correct state of the monitored objects at all times. The observation symbols are therefore probabilistic functions of the object's Markov chain, the object's true security state will be *hidden* from the agent. This is consistent with the basic idea of HMM [12].

2.1 Modeling Objects as Hidden Markov Models

Each monitored object can be represented by a HMM, defined by $\lambda = \{\mathbf{P}, \mathbf{Q}, \pi\}$.

$\mathbf{P} = \{p_{ij}\}$ is the state transition probability distribution matrix for object o, where $p_{ij} = P(x_{t+1} = s_j | x_t = s_i), 1 \leq i, j \leq N$. Hence, p_{ij} represents the probability that object o will transfer into state s_j next, given that its current state is s_i. To be able to estimate **P** for real-life objects, one may use either statistical attack data from production or experimental systems or the subjective opinion of experts. Learning algorithms may be employed in order to provide a better estimate of **P** over time.

$\mathbf{Q} = \{q_j(l)\}$ is the observation symbol probability distribution matrix for object o in state s_j, whose elements are $q_j(l) = P(y_t = v_l | x_t = s_j), 1 \leq j \leq N, 1 \leq l \leq M$. In our model, the element $q_j(l)$ in **Q** represents the probability that a sensor will send the observation symbol v_l at time t, given that the object is in state s_j at time t. **Q** therefore indicates the sensor's false-positive and false-negative effect on the agents risk assessments.

$\pi = \{\pi_i\}$ is the initial state distribution for the object. Hence, $\pi_i = P(x_1 = s_i)$ is the probability that s_i was the initial state of the object.

2.2 Quantitative Risk Assessment

Following the terminology in [1], risk is measured in terms of *consequences* and *likelihood*. A consequence is the (qualitative or quantitative) outcome of an event and the likelihood is a description of the probability of the event. To perform dynamic risk assessment, we need a mapping: $\mathcal{C} : S \rightarrow \mathbb{R}$, describing the expected cost (due to loss of confidentiality, integrity and availability) for each object. The total risk $\mathcal{R}_t$ for an object at time t is

$$\mathcal{R}_t = \sum_{i=1}^{N} \mathcal{R}_t(i) = \sum_{i=1}^{N} \gamma_t(i) \mathcal{C}(i) \tag{1}$$

where $\gamma_t(i)$ is the probability that the object is in security state s_i at time t, and $\mathcal{C}(i)$ is the cost value associated with state s_i.

In order to perform real-time risk assessment for an object, an agent has to dynamically update the object's state probability $\gamma_t = \{\gamma_t(i)\}$. Given an observation y_t, and the HMM λ, the agent can update the state probability γ_t of an object using Algorithm 1. The complexity of the algorithm is $O(N^2)$. For further details, see the Appendix.

3. Case – Real-time Risk Assessment for a Home Office

To illustrate the theory, we perform real-time risk assessment of a typical home office network, consisting of an Internet router/WLAN access point, a stationary computer with disk and printer sharing, a laptop using WLAN, and a cell phone connected to the laptop using Bluetooth. Each of the objects (hosts) in the home office network has a sensor that processes log files and checks system integrity (a host IDS). In addi-

Algorithm 2 Update state probability distribution

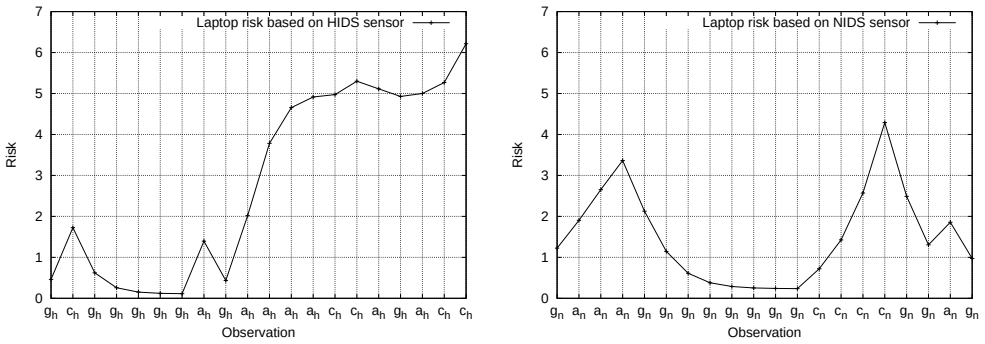
IN: y_t, λ {the observation at time t, the hidden Markov model}
OUT: γ_t {the security state probability at time t}
  if $t = 1$ **then**
    for $i = 1$ to N **do**
      $\alpha_1(i) \leftarrow q_i(y_1)\pi_i$
      $\gamma_1(i) \leftarrow \frac{q_i(y_1)\pi_i}{\sum_{j=1}^{N} q_j(y_1)\pi_j}$
    end for
  else
    for $i = 1$ to N **do**
      $\alpha_t(i) \leftarrow q_i(y_t) \sum_{j=1}^{N} \alpha_{t-1}(j)p_{ji}$
      $\gamma_t(i) \leftarrow \frac{\alpha_t(i)}{\sum_{j=1}^{N} \alpha_t(j)}$
    end for
  end if
  return γ_t

tion, the access point has a network monitoring sensor that is capable of monitoring traffic between the outside network and the internal hosts (a network IDS).

For all objects, we use the state set $S = \{G, A, C\}$. The sensors provide observations in a standardized message format, such as IDMEF, and they are capable of classifying observations as indications of the object state. Each sensor is equipped with a database of signatures of potential attacks. For the purpose of this example, each signature is associated with a particular state in S. We define the observation symbols set as $V = \{g, a, c\}$, where the symbol g is an indication of state G and so forth. Note that we have to preserve the discrete-time property of the HMM by sampling sensor data periodically. If there are multiple observations during a period, we sample one at random. If there are no observations, we assume the observation symbol to be g. In order to use multiple sensors for a single object, a round-robin sampling is used to process only one observation for each period. This is demonstrated in example 3.

The home network is monitored by an agent that regularly receives observation symbols from the sensors. For each new symbol, the agent uses Algorithm 1 to update the objects' security state probability, and (1) to compute its corresponding risk value. Estimating the matrices $\mathbf{P}$ and $\mathbf{Q}$, as well as the cost $\mathcal{C}$ associated with the different states, for the objects in this network is a non-trivial task that is out of scope for this paper.

The parameter values in these examples are therefore chosen for illustration purposes only. Also, we only demonstrate how to perform dynamic risk assessment of the laptop.



(a) HIDS sensor (b) NIDS sensor

Figure 1. Laptop risk assessment

3.1 Example 1: Laptop Risk Assessment by HIDS Observations

First, we assess the risk of the laptop, based on an observation sequence Y_{HIDS-L}, containing 20 samples collected from the laptop HIDS. We use the HMM $\lambda_L = \{\mathbf{P}_L, \mathbf{Q}_{HIDS-L}, \pi_L\}$, where

$$\mathbf{P}_L = \begin{pmatrix} p_{GG} & p_{GA} & p_{GC} \\ p_{AG} & p_{AA} & p_{AC} \\ p_{CG} & p_{CA} & p_{CC} \end{pmatrix} = \begin{pmatrix} 0.995 & 0.004 & 0.001 \\ 0.060 & 0.900 & 0.040 \\ 0.008 & 0.002 & 0.990 \end{pmatrix}, \tag{2}$$

$$\mathbf{Q}_{HIDS-L} = \begin{pmatrix} q_G(g) & q_G(a) & q_G(c) \\ q_A(g) & q_A(a) & q_A(c) \\ q_C(g) & q_C(a) & q_C(c) \end{pmatrix} = \begin{pmatrix} 0.70 & 0.15 & 0.15 \\ 0.15 & 0.70 & 0.15 \\ 0.20 & 0.20 & 0.60 \end{pmatrix}, \tag{3}$$

$$\pi_L = (\pi_G, \pi_A, \pi_C) = (0.8, 0.1, 0.1). \tag{4}$$

Since the HIDS is assumed to have low false-positive and false-negative rates, both $q_G(a), q_G(c), q_A(c) \ll 1$ and $q_A(g), q_C(g), q_C(a) \ll 1$ in $\mathbf{Q}_{HIDS-L}$. The dynamic risk in Figure 1(a) is computed based on the observation sequence Y (as shown on the x-axis of the figure) and a security state cost estimate measured as $\mathcal{C}_L = (0, 5, 10)$.

3.2 Example 2: Laptop Risk Assessment by NIDS Observations

Now, we let the risk assessment process of the laptop be based on another observation sequence, Y_{NIDS-L}, collected from the NIDS. A new observation symbol probability distribution is created for the NIDS

$$\mathbf{Q}_{NIDS-L} = \begin{pmatrix} 0.5 & 0.3 & 0.2 \\ 0.2 & 0.6 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{pmatrix}. \tag{5}$$

One can see that the NIDS has higher false-positive and false-negative rates, compared to the HIDS. Figure 1(b) shows the laptop risk when using the HMM $\lambda_L = \{\mathbf{P}_L, \mathbf{Q}_{NIDS-L}, \pi_L\}$. Note that the observation sequence is not identical to the one in example 1, as the two sensors are not necessarily consistent.

3.3 Example 3: Aggregating HIDS and NIDS Observations

The agent now aggregates the observations from the HIDS and NIDS sensors by sampling from the observation sequences Y_{HIDS-L} and Y_{NIDS-L} in a round-robin fashion. To update the current state probability γ_t, the agent therefore chooses the observation symbol probability distribution corresponding to the sampled sensor, i.e., the HMM will be

$$\lambda_L = \{\mathbf{P}_L, \mathbf{Q}^*, \pi_L\}, \text{where } \mathbf{Q}^* = \begin{cases} \mathbf{Q}_{HIDS-L} \text{ if } y_t \in Y_{HIDS} \\ \mathbf{Q}_{NIDS-L} \text{ if } y_t \in Y_{NIDS} \end{cases} . \quad (6)$$

The calculated risk is illustrated in Figure 2. The graph shows that some properties of the individual observation sequences are retained.

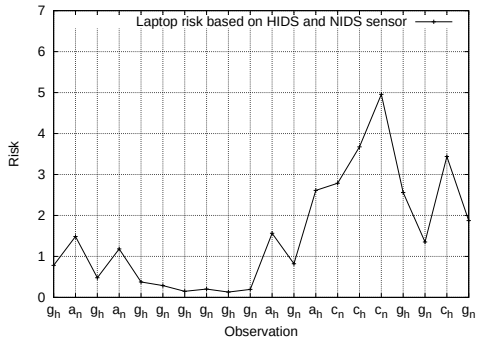


Figure 2. Laptop risk assessment based on two sensors (HIDS and NIDS)

4. Managing Risk with Automated Response

In order to achieve effective incident response, it must be possible to effectively initiate defensive measures, for example by reconfiguring the security services and mechanisms in order to mitigate risk. Such measures may be manual or automatic. An information system or network can be automatically reconfigured in order to reduce an identified risk, or the system can act as a support system for system and network administrators by providing relevant information and recommending specific actions. To facilitate such an approach, it is necessary to provide a mechanism that relates a detected security incidence to an appropriate response, based on the underlying risk model. Such a mechanism should include a policy for what reactions should be taken in the case of a particular incident, as well as information on who has

the authority to initiate or authorize the response. Examples of distributed intrusion detection and response systems have been published in [13, 14].

The dynamic risk-assessment method described in this paper can provide a basis for automated response. If the risk reaches a certain level, an agent may initiate an automated response in order to control the risk level. Such a response may be performed both for individual objects (e.g. a compromised host) or on a network-wide level (if the network risk level is too high). Examples of a local response may be firewall reconfigurations for a host, changing logging granularity, or shutting down a system. Examples of a global response may be the revocation of a user certificate, the reconfiguration of central access control configurations, or firewall reconfigurations. Other examples include traffic rerouting or manipulation, and honeypot technologies. Note that such adaptive measures has to be supervised by human intelligence, as they necessarily introduce a risk in their own right. A firewall reconfiguration mechanism can, for example, be exploited as part of a denial-of-service attack.

5. Conclusion

We present a real-time risk-assessment method using HMM. The method provides a mechanism for aggregating data from multiple sensors, with different weightings according to sensor trustworthiness. The proposed discrete-time model relies on periodic messages from sensors, which implies the use of sampling of alert data. For the purpose of real-life applications, we propose further development using continuous-time models in order to be able to handle highly variable alert rates from multiple sensors. We also give an indication as to how this work can be extended into a multi-agent system with automated response, where agents are responsible for assessing and responding to the risk for a number of objects.

References

[1] Standards Australia and Standards New Zealand: AS/NZS 4360: 2004 risk management (2004)

[2] Gehani, A., Kedem, G.: Rheostat: Real-time risk management. In: Recent Advances in Intrusion Detection: 7th International Symposium, RAID 2004, Sophia Antipolis, France, September 15-17, 2004. Proceedings, Springer (2004) 296–314

[3] Staniford-Chen, S., Cheung, S., Crawford, R., Dilger, M., Frank, J., Hoagland, J., Levitt, K., Wee, C., Yip, R., Zerkle, D.: GrIDS – A graph-based intrusion detection system for large networks. In: Proceedings of the 19th National Information Systems Security Conference. (1996)

[4] Snapp, S.R., Brentano, J., Dias, G.V., Goan, T.L., Heberlein, L.T., lin Ho, C., Levitt, K.N., Mukherjee, B., Smaha, S.E., Grance, T., Teal, D.M., Mansur, D.: DIDS (distributed intrusion detection system) - motivation, architecture, and an early prototype. In: Proceedings of the 14th National Computer Security Conference, Washington, DC (1991) 167–176

[5] Balasubramaniyan, J.S., Garcia-Fernandez, J.O., Isacoff, D., Spafford, E., Zamboni, D.: An architecture for intrusion detection using autonomous agents. In: Proceedings of the 14th Annual Computer Security Applications Conference, IEEE Computer Society (1998) 13

[6] Helmer, G., Wong, J.S.K., Honavar, V.G., Miller, L., Wang, Y.: Lightweight agents for intrusion detection. J. Syst. Softw. **67** (2003) 109–122

[7] Debar, H., Curry, D., Feinstein, B.: Intrusion detection message exchange format (IDMEF) – Internet-Draft (2005)

[8] Ourston, D., Matzner, S., Stump, W., Hopkins, B.: Applications of hidden markov models to detecting multi-stage network attacks. In: Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS). (2003)

[9] Warrender, C., Forrest, S., Pearlmutter, B.: Detecting intrusions using system calls: Alternative data models. In: Proceedings of the 1999 IEEE Symposium on Security and Privacy. (1999)

[10] Gong, F., Goseva-Popstojanova, K., Wang, F., Wang, R., Vaidyanathan, K., Trivedi, K., Muthusamy, B.: Characterizing intrusion tolerant systems using a state transition model. In: DARPA Information Survivability Conference and Exposition (DISCEX II). Volume 2. (2001)

[11] Singh, S., Cukier, M., Sanders, W.: Probabilistic validation of an intrusion-tolerant replication system. In de Bakker, J.W., de Roever, W.-P., Rozenberg, G., eds.: International Conference on Dependable Systems and Networks (DSN'03). (2003)

[12] Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. Readings in speech recognition (1990) 267–296

[13] Carver Jr., C.A., Hill, J.M., Surdu, J.R., Pooch, U.W.: A methodology for using intelligent agents to provide automated intrusion response. In: Proceedings of the IEEE Workshop on Information Assurance and Security. (2000)

[14] Porras, P.A., Neumann, P.G.: EMERALD: Event monitoring enabling responses to anomalous live disturbances. In: Proc. 20th NIST-NCSC National Information Systems Security Conference. (1997) 353–365

Appendix: On Algorithm 1

Given the first observation y_1 and the hidden Markov model λ, the initial state distribution $\gamma_1(i)$ can be calculated as

$$\gamma_1(i) = P(x_1 = s_i|y_1, \lambda) = \frac{P(y_1, x_1 = s_i|\lambda)}{P(y_1|\lambda)} = \frac{P(y_1|x_1 = s_i, \lambda)P(x_1 = s_i|\lambda)}{P(y_1|\lambda)}. \tag{A.1}$$

To find the denominator, one can condition on the first visited state and sum over all possible states

$$P(y_1|\lambda) = \sum_{j=1}^{N} P(y_1|x_1 = s_j, \lambda)P(x_1 = s_j|\lambda) = \sum_{j=1}^{N} q_j(y_1)\pi_j. \tag{A.2}$$

Hence, by combining (A.1) and (A.2)

$$\gamma_1(i) \quad = \quad \frac{q_i(y_1)\pi_i}{\sum_{j=1}^{N} q_j(y_1)\pi_j}, \tag{A.3}$$

where $q_j(y_1)$ is the probability of observing symbol y_1 in state s_j, and π is the initial state probability. To simplify the calculation of the state distribution after t observations we use the *forward-variable* $\alpha_t(i) = P(y_1 y_2 \cdots y_t, x_t = s_i|\lambda)$, as defined in [12]. By using recursion, this variable can be calculated in an efficient way as

$$\alpha_t(i) = q_i(y_t) \sum_{j=1}^{N} \alpha_{t-1}(j)p_{ji}, \qquad t > 1. \tag{A.4}$$

From (A.1) and (A.3) we find the initial forward variable

$$\alpha_1(i) = q_i(y_1)\pi_i, \qquad t = 1. \tag{A.5}$$

In the derivation of $\alpha_t(i)$ we assumed that y_t only depend on x_t and that the Markov property holds.

Now we can use the forward variable $\alpha_t(i)$ to update the state probability distribution by new observations. This is done by

$$\begin{aligned}\gamma_t(i) &= P(x_t = s_i | y_1 y_2 \cdots y_t, \lambda) = \frac{P(y_1 y_2 \cdots y_t, x_t = s_i | \lambda)}{P(y_1 y_2 \cdots y_t | \lambda)} \\ &= \frac{P(y_1 y_2 \cdots y_t, x_t = s_i | \lambda)}{\sum_{j=1}^{N} P(y_1 y_2 \cdots y_t, x_t = s_j | \lambda)} = \frac{\alpha_t(i)}{\sum_{j=1}^{N} \alpha_t(j)}.\end{aligned} \tag{A.6}$$

Note that (A.6) is similar to Eq. 27 in [12], with the exception that we do not account for observations that occur after t, as our main interest is to calculate the object's state distribution after a number of observations.

PAPER F

A Framework for Predicting Security and Dependability Measures in Real-time

Karin Sallhammar, Bjarne E. Helvik and Svein J. Knapskog

International Journal of Computer Science and Network Security (IJCSNS), Vol. 7, No. 3, pp. 169-183, March 2007.

A FRAMEWORK FOR PREDICTING SECURITY AND DEPENDABILITY MEASURES IN REAL-TIME

Karin Sallhammar, Bjarne E. Helvik and Sven J. Knapskog
Centre for Quantifiable Quality of Service in Communication Systems
Norwegian University of Science and Technology,
Trondheim, Norway
{ sallhamm, bjarne, knapskog } @q2s.ntnu.no

Abstract The complex networked system of today that our technological and social society rely on are vulnerable to a large number of failures, accidental as well as intentional. Ideally, the service delivered by such a system should be both dependable and secure. This paper presents a framework for integrated security and dependability assessment. The proposed model is based on traditional stochastic analysis techniques, supported by live data from network sensors, which is used to estimate the current state and predict the future behavior of a system in real-time. The method is demonstrated by a small case study.

1. Introduction

The new paradigms of ubiquitous computing and high capacity data transfer has led to an explosive growth in the number and complexity of computing systems used for critical applications, such as electronic commerce, health-care and urgent information interchange. Since the modern society of today is highly dependent on these services, the computing systems need to function properly despite not only accidental failures but also malicious attacks. To increase the trustworthiness of the implemented services, it should be possible to monitor the systems' current robustness towards these impairments, as well as assess and predict their current and near future behavior. This paper deals with a method for such monitoring and prediction.

A system's ability to provide a correct and timely service can be described in terms of its dependability and security. Dependability is the ability to deliver service that can justifiably be trusted, and can be stated as an integrative concept that encompasses the attributes availability, reliability, safety, integrity and maintainability [2]. Security, on the other hand, is defined as a concept addressing the attributes confidentiality, integrity and availability [8]. To function properly, the critical applications and systems our society relies upon need to be both dependable *and* secure. However, despite the fact that a system cannot be considered trustworthy without a rigorous analysis comprising a joint consideration of these two concepts, dependabil-

ity and security have tended to be treated separately. To allow continuous estimation of the trustworthiness of the services provided by today's computing systems, there is an urgent need of new modeling methods that treats both security and dependability.

To model, analyze and evaluate systems that are yet to be built or systems whose specific vulnerabilities remain unknown, stochastic assumptions are needed [11]. During the last decade, probabilistic modeling of security has gained a lot of interest [10, 12, 9, 3, 17, 11]. Such models, which are inspired by the traditional dependability analysis techniques, can be used to provide quantitative measures of a system's operational security. However, most of the recent research efforts have focus on either security or dependability analysis, rather than aiming for a unified evaluation framework. In [15] we described a method for integrated security and dependability evaluation, which uses stochastic analysis techniques to model and compute expected failure times for a computing system, regardless of whether the failure cause is intentional or not. To incorporate malicious behavior in the stochastic model, a game theoretic approach is applied. The model can be used for both predicting a system's future security and dependability behavior, as well as for a trade-off analysis of possible countermeasures. This paper extends our previously published results by integrating the proposed model in a distributed network monitoring environment. By using observations provided by network sensors, the probability of the current system state can be estimated, which makes it possible to use the stochastic model to predict the future behavior of the monitored computer system in real-time. The overall concepts used in our framework is depicted in Fig. 1. The system that is to

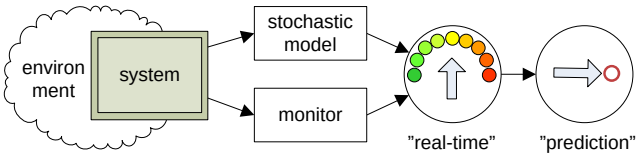


Figure 1. The overall concepts.

be assessed, together with its operational environment, is described by a stochastic model. There is a monitor that continuously surveys the system and gives warnings of possible disturbances. By using the stochastic model together with real-time data from the monitor our model can estimate the current state of the system, and predict its future behavior.

This paper is organized as follows. Section 2 starts by reviewing the basis of stochastic modeling, and introduces the proposed security and dependability measures. Section 3 discusses how security analysis differs from traditional dependability evaluation, and explains how our integrated framework treat these issues. In Section 4 the proposed security and dependability assessment architecture is presented. In Section 5 we explain how the monitoring architecture is used to collect sensor data and how this information is interpreted to estimate the current system state. Section 6 provides the methodology for computing real-time measures of the system. In Sec-

tion 7 the approach is demonstrated by an illustrative example. Finally, Section 8 concludes the paper and points to future work.

2. Predicting Security and Dependability

When operating a system with security and dependability requirements it is of interest to be able answer questions like: "Is the system already compromised?", "What is the probability that the system is currently under attack?", "What is the probability that the system will operate without security breaches or failures for the next 24 hours?", or "What is the expected time until such an event occurs?". The objective of this paper is to provide a methodology that can be used to provide answers to such questions. For the simplicity of the presentation we denote any undesired event in or state of the system as a *failure*, in accordance with [2], without discrimination with respect to kind of failure. A refinement is straight forward. Hence, we denote the time from now and until the next failure by T_F, and seek to obtain:

$P_F(t) = P(T_F > t)$ the probability that the time until the next failure is greater than t.[1]

$MTNF = \frac{1}{P_F(0)} \int_0^\infty P_F(t)\, dt$ the mean time to next failure[2], assuming that the system will sooner or later fail (i.e., $\lim_{t\to\infty} P_F(t) = 0$).

Dealing with security issues, it is not necessarily evident when the system is failed (compromised), so we will in general have $P_F(0) \leq 1$, where $1 - P_F(0)$ represents the probability that the system is already failed (compromised).

To obtain the above measures, two important elements are needed:

- the ability to predict the current state of the system, and
- for a given state, the ability to predict the future behavior of the system.

We will return to these elements later on in this paper. First, we introduce the overall modeling approach applying Markov models and discuss how security issues may be included in these.

2.1 The Stochastic Modeling Approach

Above we have informally introduced the concept of state. By a *state* in this context is meant an operational mode of the system characterized by which units of the system that are operational or failed, whether there are ongoing attacks, active countermeasures, operational and maintenance activities, whether parts of the system compromised or not, etc. The decision of what to include or not in the state definition

[1] This expression corresponds to the reliability function in traditional dependability analysis (where only random failures are regarded), but this term is not used to avoid misinterpretation.

[2] The $MTNF$ measure differs from the $MTTF$ (mean time to failure) and $MTFF$ (mean time to first failure) traditionally used in dependability analysis. In contrast to $MTTF$ and $MTFF$, the $MTNF$ measure is conditioned on $P_F(0)$. The measure will be further explained in Section 6.

is a trade-off between model representativeness and complexity, and is a salient part of every modeling and analysis effort. The more system states that are taken into consideration, the more fine-granular the model becomes. Since the model needs to be parametrized, the granulation of the state space must be carefully considered. Too simple models will not provide any valuable insight into the system behavior, whereas too complex models will quickly lead to state space explosion. An example primarily for illustration will be presented in Section 7.

Let say the system has N disjoint states and that at any time it is in one of these. The behavior of the system is characterized by the *transitions* between the states, each transition triggered by an event. The event that will occur next, as well as the time until next event, is random. Hence, the behavior of the system is a stochastic process. For the sake of simplicity, assume that the occurrence rates of events depend only on the state the system is in, i.e. the system is modeled by a continuous time Markov chain (CTMC) (for an introduction to Markov modeling for dependability analysis, see for instance [6]). A CTMC is characterized by its rate matrix, whose elements represents the transition rates between the system states.

2.2 System Equations

Assume that the system has a finite number of states, denoted $\mathbf{S} = \{S_1, \ldots, S_N\}$. This state set can be split into two subsets: $\mathbf{S}_G$, which contains the good system states, and $\mathbf{S}_F$, which contains the failed system states. Let

$$\mathbf{X}(t) = \{X_1(t), \ldots, X_N(t)\}, \tag{1}$$

where $X_i(t)$ denotes the probability that the system is in state i at time t. The state equation describing the system behavior is then

$$\frac{d}{dt}\mathbf{X}(t) = \mathbf{X}(t)\mathbf{Q}, \tag{2}$$

where $\mathbf{Q} = \{q_{ij}\}$ is the $N \times N$ state transition rate matrix of the system. The element q_{ij} represents the transition rate from state i to state j. Note that $q_{ii} = -\sum_{i \neq j} q_{ij}$. The state equation can be solved if the initial state of the system $\mathbf{X}(0)$ is known. Then

$$\mathbf{X}(t) = \mathbf{X}(0)\text{exp}(\mathbf{Q}t). \tag{3}$$

The solution to (3) provides the transient state probabilities for a system. See for instance [14]. However, the probability that a CTMC will be in state i at time t often converges to a limiting value, which is independent of the initial state. The steady state probabilities

$$\mathbf{X} = \{X_1, \ldots, X_N\}, \tag{4}$$

whose elements $X_i = \lim_{t \to \infty} X_i(t), i = 1, \ldots, N$, can then be obtained by solving the set of N equations given by $N - 1$ of the N equations

$$\mathbf{XQ} = \mathbf{0}, \tag{5}$$

and with the N'th equation

$$\sum_{l=1}^{N} X_l = 1. \tag{6}$$

As is common practice in dependability analysis, the CTMC can be used as a basis for obtaining various measures of the system, such as the $MTNF$ or $P_F(t)$ previously discussed. For computation of measures relating to T_F, the failure states may be made absorbing, i.e. $q^*_{ij} = 0$ when $i \in \mathbf{S}_F, j \in \mathbf{S}_G$, otherwise $q^*_{ij} = q_{ij}$. Let $\mathbf{Q}^* = \{q^*_{ij}\}$ be the modified state transition rate matrix with absorbing failure states and denote by $\mathbf{X}^*(t)$ the corresponding state probabilities. Hence,

$$P_F(t) = \sum_{i \in \mathbf{S}_G} X^*_i(t), \tag{7}$$

from which the $MTNF$ can be computed. Rather than integrating $\int_0^\infty P_F(t)\,dt$ to obtain $MTNF$ we adopt a computationally more efficient approach, based on [4]. The details together with computational issues will be further explained in Section 6.

3. The Challenges with Security Modeling

In dependability analysis it is very common to use the stochastic modeling approach described in Section 2 to quantify the reliability or availability of systems. In that case, the states are classified as either "up" states (the good states in $\mathbf{S}_G$) or "down" states (the failed states in $\mathbf{S}_F$), depending on whether the required service is delivered or not. In theory, by associating down states with failures of confidentiality or integrity one can use these methods also for evaluating the security properties of a software system. A simple example is depicted in Fig. 2, where $\mathbf{S} = \{G, C, F\} = \{$"good", "compromised", "failed"$\}$. Here, it is assumed that a

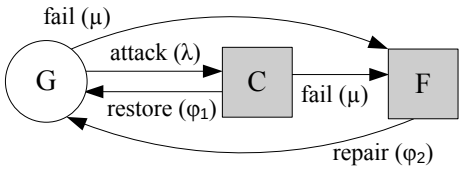


Figure 2. A simple Markov model, including a compromised system state.

large number of attackers are targeting the system in state G, with accumulated attack intensity λ. In contrast to attack graphs (as used in e.g. [9]) where each state transition corresponds to a single atomic step of a penetration, our model aim to be more high-level and focus on the *impact* of the attacks on the system rather than on the specific attack procedures themselves. This facilitates the modeling of unknown attacks in terms of generic state transitions. For example, in Fig. 2 the attack is merely defined as "the action that seeks to transfer the system from a good state to a compromised state" and nothing more. For real-world cases, where more complex models are needed, two problems quickly arise:

1 The attacks are *intentional*, rather than purely random.

2 Regarding security, the current system state may be *unobservable*.

The remainder of this section will discuss these two problems and explain our proposed solutions.

3.1 Incorporating Attacker Behavior.

When using the traditional Markov approach it is (in most cases) straightforward to model accidental failures as state transitions. However, since attacks are intentional they may not always be well characterized by models of random nature. Hence, a more sophisticated approach than the simple model depicted in Fig. 2 is needed. To be able to model the effect of a successful attack as a transition between system states one needs to consider the two underlying causes of any attack. As pointed out in [15], there must be (at least) one vulnerability in the system, and a malicious action that tries to exploit that vulnerability, for an attack to be successful. Even though the time an attacker needs to perform an attack action may be modeled as randomly distributed, the *decision* to perform the action will also influence the system failure rate. Therefore, attacker behavior must be represented in the state transitions. In this paper we follow the approach in [15] and define $\pi_i(a)$ as the probability that an attacker will choose action a when the system is in (the vulnerable) state i. The failure rate between state i and j when incorporating malicious behavior can therefore be computed as

$$q_{ij} = \pi_i(a)\lambda_{ij}(a), \tag{8}$$

where $\lambda_{ij}(a)$ is the accumulated intensity if all potential attackers always take action a. By introducing the attack probability $\pi_i(a)$ as an element in the rate value q_{ij}, the result from a successful attack can be modeled as one or more *intentional state changes* of the underlying stochastic process, which represents the dynamic behavior of the system. To compute the attack probabilities we use the game model published in [15]. The game model is based on a reward- and cost concept, which makes it possible to predict the expected attacker behavior, in terms of attack probabilities, for a number of different attacker profiles. The game theoretic approach will not be further explained in this paper; the reader is referred to [15] for the exact details.

3.2 Observing the System Security State.

In dependability analysis, the system state set $\mathbf{S}$ is usually considered known. Moreover, all states are assumed to be deterministically observable, in that the current system state is well defined and perceptible, at all times. However, in a security context the degree of unobservability may be quite high. A system might very well seem to be in a good state even though it is compromised, e.g. due to a stealthy attack. How can one compute measures such as $P_F(t)$ or $MTNF$ if one does not know the initial state of the system with certainty? Our solution is to use information from network sensors monitoring the system to estimate its current state probability. We then replace $\mathbf{X}(0)$ in (3) with the most likely state probability at that particular time

instant, which provides us with the possibility of re-computing the system measures in real-time. This procedure will be further explained in the subsequent sections of this paper.

4. The Prediction Framework

The proposed real-time security and dependability prediction framework is illustrated in Fig. 3. As depicted in the figure, the system is described by a three-part stochastic model, which consists of the state space **S**, the game model Γ and the corresponding state transition rate matrix **Q**. Naturally, the system behavior will depend on its operating environment, such as user behavior, administrative activities, the possible intrusions and exploits, and random software and hardware failures. Note that we include attacker profile data as a separate part of the system environment. As mentioned in the previous section, by using cost- and reward values from the attacker profile, the game model is used to compute the attack probabilities that are to be incorporated in the rate matrix. As Fig. 3 indicates, the purpose of the stochastic model is to provide the system rate values needed to predict the $P_F(t)$ and $MTNF$ measures.

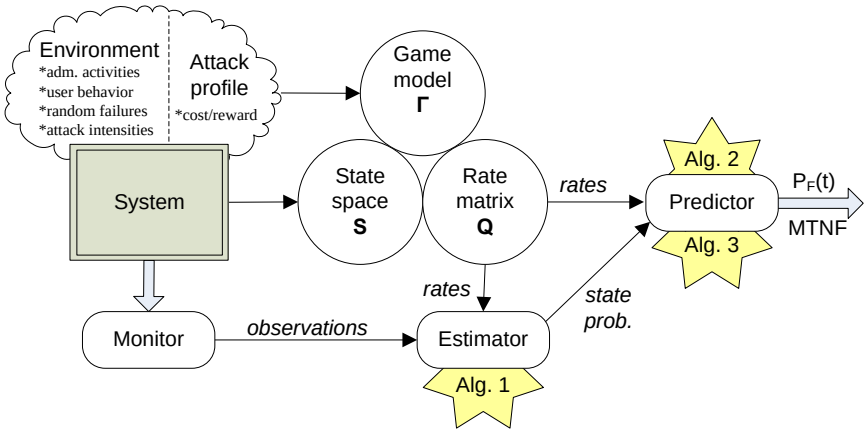


Figure 3. The security and dependability assessment architecture

From Fig. 3 it is clear that to perform real-time security and dependability assessment for the system, there are three main tasks that has to be performed; *monitoring*, *estimation* and *predicting*. The main task of the monitor is to provide the estimator with observations regarding the system security and dependability behavior. The monitor is implemented as a distributed architecture consisting of agents that observe the system by means of network sensors. The estimator then uses the observations provided by the monitor to estimate the current state probability of the system. To be able to obtain the goal of real-time security and dependability measurements, the current state probability is updated and forwarded to the predictor as soon as a new observation has been received and interpreted. Finally, the predictor computes mea-

sures for the observed system, based on the transition rates from the stochastic model together with the estimated state probability. As previously discussed, the predictor uses well-known Markov analysis techniques to compute the $P_F(t)$ and $MTNF$ for the system. The implementation details of the monitoring and estimation architecture (Alg. 3) will be further described in Section 5 and the exact procedure for computing the predicted measures (Alg. 4-5) will be explained in Section 6.

5. The Monitoring and Estimation Architecture

The proposed monitor and estimator in Fig. 3 are both based on the results published in [1]. In this paper we restrict ourselves to an overall description of the architecture. The reader is referred to [1] for more details.

5.1 The Monitor

In [1], the monitor is implemented as a distributed monitoring architecture consisting of agents that observe one or more systems using network sensors. A *sensor* can be any information-gathering program or device, including network sniffers using sampling or filtering, different types of intrusion detection systems (IDS), logging systems, virus detectors, etc. The main task of a sensor is to gather information regarding the current state of one or more systems. The assumed monitoring architecture is hybrid in the sense that it supports any type of sensor. However, it is assumed that the sensors are able to classify and send standardized observations according to the state estimation model described in this paper. An *agent* is a computer program capable of a certain degree of autonomous action. An agent is responsible for collecting and aggregating sensor data from a set of sensors that monitor one or more systems and to forward these data to the estimator. In a multi-agent system, agents are capable of communicating and cooperating with other agents. A multi-agent architecture is preferable over a single agent implementation, due to its flexibility and scalability. The case study presented later on in this paper make use of a single agent only.

In real-life distributed agent-sensor implementations, observations often arrive in bursts, and there will also be silent periods without any activity at all. In this paper we let the agent adopt a sampling process of the sensor(s) monitoring a particular system, similarly to the approach in [18]. By providing the estimator with observations at regular time intervals, the predicted system security and dependability measures can be updated at a pre-defined frequency. The sampling process will be further explained in the next subsection.

5.2 The Discrete Sampling Process

Recall that we use a CTMC to model the security and dependability behavior of a system. Due to its stochastic behavior, the system may be in any of the states in $\mathbf{S}$ when sampled. As Fig. 3 indicates, the purpose of the estimator is to use the sampled observations to estimate the current system state. To formalize, let z_τ be the (possibly unobservable) system state at sampling instant τ. The *sequence* of states that a system

is in during the sampling instants will then be denoted $Z = (z_1, z_2, \ldots)$. Let

$$\mathbf{X}^\tau = \{X_1^\tau, \ldots, X_N^\tau\}, \tag{9}$$

where X_i^τ denotes the probability that the system is in state i at the τ'th sample. Since one cannot assume that Z is known (recall the unobservability problem discussed in Section 3.2), it is this state probability that will be estimated and used to predict new system measures, at each sampling instant τ.

Recall the system rate matrix $\mathbf{Q}$. Assume that the interval between two different adjacent samples is fixed to Δ. Now, let $\mathbf{P}(\Delta)$ denote the one-step transition probability matrix with elements $p_{ij}(\Delta)$, such that $p_{ij}(\Delta) = P(z_{\tau+1} = j | z_\tau = i), 1 \leq i, j \leq N$. Hence, $p_{ij}(\Delta)$ represents the probability that the system will, given that its current state at sampling instant τ is i, be in state j after an additional time Δ, i.e. at the next sample $\tau + 1$. By using (3), $\mathbf{P}(\Delta)$ can be derived from $\mathbf{Q}$ as

$$\mathbf{P}(\Delta) = \mathbf{I} \cdot \exp(\mathbf{Q}\Delta), \tag{10}$$

where $\mathbf{I}$ is the identity matrix. For simplicity, we let $\mathbf{P}$ represents $\mathbf{P}(\Delta)$ in the rest of this paper. It is important to notice that even though we used a fixed sampling interval Δ in this paper, this is not a requirement for the model to work. One can easily imagine scenarios where it is desirable to sample the sensors and predict new system measures at irregular time intervals. In that case $\mathbf{P}(\Delta)$ needs to be recomputed at each sampling instant τ.

5.3 Interpreting Observations

Due to the inhomogeneity of sensor types, the observations can consist of a variety of information; different types of alarms, suspect traffic patterns, entries in log data files, input from network administrators, indications of system elements up and down, ongoing operational and maintenance activities, and so on. To formalize, we assume that any observation can be classified as one of the symbols in the finite symbol set $V = \{v_1, \ldots, v_M\}$. The *sequence* of observations that the monitor forwards to the estimator is denoted $Y = (y_1, y_2, \ldots)$, where $y_\tau \in V$ is the observation received at sampling instant τ. Based on Y, the estimator will estimate the system's current state, in terms of the state probability $\mathbf{X}^\tau$ in (9). The estimator will receive observations originating from more than one sensor, and these sensors may provide different types of data or even inconsistent data. All sensors will not be able to register all kinds of activities, so one cannot assume that the estimator is able to resolve the correct state of the monitored system at all times. The observation symbols are therefore probabilistic functions of the system's Markov chain, i.e. the system's true state will be *hidden* from the estimator. This is consistent with the basic idea of hidden Markov models (HMM), as described in [13].

5.4 State Probability Estimation

Each monitored system can be represented by a HMM, defined by the three tuple $\Lambda = (\mathbf{P}, \mathbf{X}^1, \mathbf{O})$. As previously discussed, $\mathbf{P} = \{p_{ij}\}$ is the one-step transition

probability matrix for the system. $\mathbf{X}^1 = \{X_1^1, \cdots, X_N^1\}$ is the state probability distribution of the system when the sampling starts, i.e. at sample instant $\tau = 1$. If one does not know the initial state probability of the system, the elements in $\mathbf{X}^1$ have to be estimated, for instance by using the system steady state probabilities in (4). $\mathbf{O} = \{o_j(l)\}$ is the observation symbol probability matrix for a system during sampling. Its elements are $o_j(l) = P(y_\tau = v_l | z_\tau = j), 1 \leq j \leq N, 1 \leq l \leq M$, i.e. $o_j(l)$ represents the probability that a sensor will provide the observation symbol v_l when sampled, given that the system is in state j. The elements of $\mathbf{O}$ will therefore give an indication of the sensor's false-positive and false-negative effect on the security and dependability prediction process. Note that if there are more than one sensor monitoring a particular system, one should define a separate observation symbol probability vector $\mathbf{O}_k$ for each sensor k.

By using an observation y_τ and the HMM Λ, the estimator will compute and replace $\mathbf{X}^\tau$ in (9) with $\hat{\mathbf{X}}^\tau$, where $\hat{\mathbf{X}}^\tau$ is the system's *most likely state probability* at sampling instant τ. This is done by means of Alg. 3. The complexity of the algorithm is $O(N^2)$, where N is the number of system states.

Algorithm 3 Estimate the current state probability

IN: y_τ, Λ {an observation at sampling instant τ, the HMM}
OUT: $\hat{\mathbf{X}}^\tau$ {the estimated state probability at sampling instant τ}

```
if τ = 1 then
  for i = 1 to N do
    α_i^τ ← o_i(y_1) X_i^1
    X̂_i^τ ← α_i^τ / Σ_{j=1}^N α_j^τ
  end for
else
  for i = 1 to N do
    α_i^τ ← o_i(y_τ) Σ_{j=1}^N α_j^{τ-1} p_ji
    X̂_i^τ ← α_i^τ / Σ_{j=1}^N α_j^τ
  end for
end if
return X̂^τ = {X̂_1^τ, ..., X̂_N^τ}
```

To see why Alg. 3 works, note that, given the first observation y_1 at $\tau = 1$, and the HMM $\Lambda = (\mathbf{P}, \mathbf{X}^1, \mathbf{O})$, the elements in a new initial state probability $\hat{\mathbf{X}}^1$ can be estimated as

$$\hat{X}_i^1 = P(z_1 = i | y_1, \Lambda) = \frac{P(y_1, z_1 = i|\Lambda)}{P(y_1|\Lambda)} = \frac{P(y_1|z_1 = i, \Lambda)P(z_1 = i|\Lambda)}{P(y_1|\Lambda)}. \quad (11)$$

To find the denominator, one can condition on the first visited state and sum over all possible states

$$P(y_1|\Lambda) = \sum_{j=1}^{N} P(y_1|z_1 = j, \Lambda)P(z_1 = j|\Lambda) = \sum_{j=1}^{N} o_j(y_1)X_j^1. \tag{12}$$

Hence, by combining (11) and (12)

$$\hat{X}_i^1 \quad = \quad \frac{o_i(y_1)X_i^1}{\sum_{j=1}^{N} o_j(y_1)X_j^1}. \tag{13}$$

To simplify the computation of the estimated state probability at the τ't observation we use the *forward-variable* $\alpha_i^\tau = P(y_1 \cdots y_\tau, z_\tau = i|\Lambda)$, as defined in [13]. By using recursion, this variable can be calculated in an efficient way as

$$\alpha_i^\tau = o_i(y_\tau) \sum_{j=1}^{N} \alpha_j^{\tau-1} p_{ji}, \qquad \tau > 1. \tag{14}$$

In the derivation of α_i^τ we assumed that y_τ depends on z_τ only, and that the Markov property holds. From (11) and (13) we find the initial forward variable

$$\alpha_i^1 = o_i(y_1)X_i^1, \qquad \tau = 1. \tag{15}$$

Now we can use the forward variable α_i^τ to update the estimated state probability distribution by new observations. This is done by

$$\begin{aligned} \hat{X}_i^\tau &= P(z_\tau = i|y_1 \cdots y_\tau, \Lambda) = \frac{P(y_1 \cdots y_\tau, z_\tau = i|\Lambda)}{P(y_1 \cdots y_\tau|\Lambda)} \\ &= \frac{P(y_1 \cdots y_\tau, z_\tau = i|\Lambda)}{\sum_{j=1}^{N} P(y_1 \cdots y_\tau, z_\tau = j|\Lambda)} = \frac{\alpha_i^\tau}{\sum_{j=1}^{N} \alpha_j^\tau}. \end{aligned} \tag{16}$$

6. Making the System Predictions

The final step in the security and dependability assessment process illustrated in Fig 3 is that the predictor uses the estimated state probability distribution together with the state transition rate matrix to compute system measures. This section provides the algorithms (Alg. 4-5) together with as a detailed explanation of the mathematical equations that are used to compute the $P_F(t)$ and $MTNF$ measures.

6.1 Computing $P_F^\tau(t)$

Recall the definition of $P_F(t)$ provided in (7). To use the estimated state probabilities to compute the function at sample instant τ, Alg. 4 can be used.

As can be seen from the algorithm, at each sampling instant τ there are three main steps to perform. First, the algorithm sets the initial state probability equal to the estimated, i.e. $\mathbf{X}^*(0) \leftarrow \hat{\mathbf{X}}^\tau$. The $\mathbf{X}^*(0)$ vector is then used when solving the system

Algorithm 4 Predict the $P_F^\tau(t)$ measures

IN: $\mathbf{Q}^*, \hat{\mathbf{X}}^\tau$ {the modified rate matrix, the estimated state prob. at τ}
OUT: $P_F^\tau(t)$ {the predicted $P_F(t)$ at sample τ}
  for $i = 1$ to N **do**
    $X_i^*(0) \leftarrow \hat{X}_i^\tau$
    $X_i^*(t) \leftarrow X_i^*(0)\text{exp}(\mathbf{Q}^* t)$
    if $i \leq K$ **then**
      $P_F^\tau(t) += X_i^*(t)$
    end if
  end for
  return $P_F^\tau(t)$

state equation defined in (2), i.e. $\mathbf{X}^*(t) = \mathbf{X}^*(0)\text{exp}(\mathbf{Q}^* t)$. To solve the system state equation, the algorithm uses the Mathematica package "StateDiagrams.m" [7], which implements the theory in [4]. Then, in accordance to (7), the $P_F^\tau(t)$ function is computed as

$$P_F^\tau(t) = \sum_{i \in \mathbf{S}_G} X_i^*(t). \tag{17}$$

Even though this definition of $P_F^\tau(t)$ is very similar to the traditional definition of the system reliability function (see e.g. [6]), there is a crucial difference. As a consequence of the estimation process, we cannot make the usual assumption that the system state is good when computing (17). Because $\hat{\mathbf{X}}^\tau$ is used to determine $\mathbf{X}^*(0)$ in step 1 it might be that $\sum_{i \in S_G} X_i^*(0) \neq 1$. Hence, to validate the predicted system measures, one should also use (17) to compute $P_F^\tau(0) = \sum_{i \in \mathbf{S}_G} X_i^*(0)$, i.e. the probability that the system actually is in a good state at sampling instant τ.

6.2 Computing $MTNF^\tau$

To compute the mean time to next failure ($MTNF$) measure at sampling instant τ, Alg. 5 is used.

Algorithm 5 Predict the $MTNF^\tau$ measures

IN: $\mathbf{Q}_1, \hat{\mathbf{X}}^\tau$ {(a part of) the rate matrix, the estimated state prob. at τ}
OUT: $MTNF^\tau$ {the predicted $MTNF$ at sample τ}
  for $j = 1$ to K **do**
    $\hat{\hat{X}}_j^\tau \leftarrow \frac{\hat{X}_j^\tau}{\sum_{j=1}^K \hat{X}_j^\tau}$
    define $-\sum_{i=1}^K T_i q_{ij} = \hat{\hat{X}}_j^\tau$
  end for
  solve for all T_i
  return $MTNF^\tau = \sum_{i=1}^K T_i$

The algorithm has been implemented in accordance to methodology in [4], slightly modified to fit into the context of the proposed security and dependability assessment architecture. Suppose the states are ordered, such that $\mathbf{S}_G = \{S_1, \ldots, S_K\}$ and $\mathbf{S}_F = \{S_{K+1}, \ldots, S_N\}$. Then $\mathbf{Q}$ can be written in partitioned form as

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \\ \mathbf{Q}_3 & \mathbf{Q}_4 \end{pmatrix}, \tag{18}$$

where the size of $\mathbf{Q}_1$ is $K \times K$, the size of $\mathbf{Q}_2$ is $K \times (N-K)$ and so forth. Now also the estimated state probability vector can be partitioned as $\hat{\mathbf{X}}^\tau = \{\hat{\mathbf{X}}_G^\tau, \hat{\mathbf{X}}_F^\tau\}$, where $\hat{\mathbf{X}}_G^\tau = \{\hat{X}_1^\tau, \ldots, \hat{X}_K^\tau\}$ and $\hat{\mathbf{X}}_F^\tau = \{\hat{X}_{K+1}^\tau, \ldots, \hat{X}_N^\tau\}$.

To compute the system's expected time to next failure, one has to assume that the system is in one of the good states in $\mathbf{S}_G$ at sampling instant τ (otherwise $MTNF = 0$, since the system already has failed). Therefore, the estimated state probabilities in $\hat{\mathbf{X}}_G^\tau$ must be renormalized such that

$$\hat{\hat{\mathbf{X}}}_G^\tau = \frac{\hat{\mathbf{X}}_G^\tau}{\hat{\mathbf{X}}_G^\tau \mathbf{h}_K} \tag{19}$$

where $\mathbf{h}_K$ is a column vector of K ones. Define $\mathbf{T} = \{T_1, \ldots T_K\}$. By solving

$$-\mathbf{T}\mathbf{Q}_1 = \hat{\hat{\mathbf{X}}}_G^\tau \tag{20}$$

the mean time to next failure for the system at the particular instant τ can be computed as

$$MTNF^\tau = \sum_{i=1}^{K} T_i, \tag{21}$$

provided that the system is in any of the good states in $\mathbf{S}_G$ when sampled. The main difference between the $MTNF^\tau$ measure used in this paper and the $MTFF$ (mean time to first failure) measure used in traditional dependability analysis, is that when computing $MTFF$ the system is considered *new* when it starts to deliver its intended service, i.e. $\mathbf{X}(0) = \{1, 0, \ldots, 0\}$. In contrast, $MTNF^\tau$ is computed from the estimated state probability rather than the initial system state probability. The advantage with our approach is that by computing $MTNF^\tau$ as proposed in (21) one can use the real-time observations provided by the monitoring architecture to make a better prediction of the system's expected time to next failure, and update this prediction whenever new information arrives. Hence, in contrast to the static $MTFF$, the $MTNF^\tau$ will be a *dynamic* system measure, more suitable for a real-time system assessment architecture. However, as previously discussed, since the $MTNF^\tau$ is conditioned on a good system state at sampling instant τ (i.e. that $z_\tau \in \mathbf{S}_G$) the measure should always be evaluated together with the corresponding $P_F^\tau(0)$ to make sense. This will be illustrated in the case study in the next section.

7. Case Study: A Database Server

To illustrate the proposed approach, we model and simulate the security and dependability assessment process for a typical network service configuration consisting

of a database service for a local area network (LAN). In this paper we consider a single server implementation only, however, the example model can easily be extended to the more commonly used distributed server implementation (see [16] for an example). Fig. 4 illustrates the database server that is to be assessed in this study.

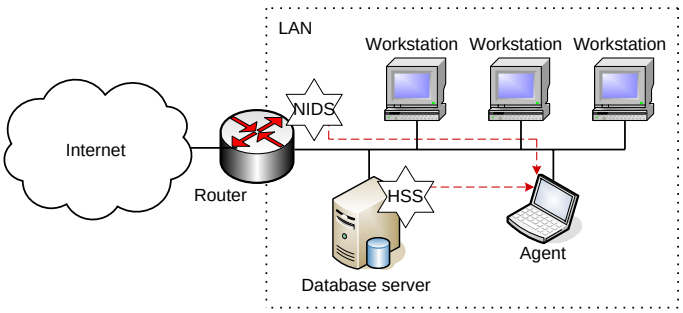


Figure 4. The database server in its network environment.

In this example, the database server is assumed to be subject to accidental software- and hardware failures, as well as packet flooding Denial of Service (DoS) attacks originating from outside the LAN. As can be seen, the database server is monitored by a distributed agent-sensor system consisting of one agent that samples and interprets information from two different kind of sensors; a network intrusion detection system (NIDS) and a host-based sensor system (HSS). The NIDS monitors traffic between the outside network and the internal LAN, and the HSS processes log files and checks system status locally at the database server.

7.1 The Stochastic Model

The database server can be modeled by a four-state CTMC. State G means that the server is fully operational, i.e. it is a system "up" state. In state A the server is subject to an ongoing DoS attack, which means that its performance is degraded so that the service is only partially available. Still, the A state is considered a system "up" state. If the DoS attack is detected and reacted to before the server crashes, the system will return to state G. In the "down" states SF and HF the server is subject to software and hardware failures, respectively. A hardware failure requires a manual repair. To recover from a software failure, only a server reboot is needed. Note that since also the effect of a successful DoS attack is a software failure requiring a reboot, we do not need to distinguish between accidental and malicious software failure modes in the stochastic model. Hence, the complete state set is $\mathbf{S} = \{G, A, SF, HF\}$ whereof $\mathbf{S}_G = \{G, A\}$ and $\mathbf{S}_F = \{SF, HF\}$, as illustrated in Fig. 5.

The time to failure, attack and repair are assumed to follow the exponential distributions $\lambda e^{-\lambda t}$, $\varphi e^{-\varphi t}$ and $\mu e^{-\mu t}$, respectively. The specific rates used in this example are $\lambda_S = 0.005$, $\lambda_H = 0.0003$, $\varphi_1 = 0.002$, $\varphi_2 = 60$, $\mu_A = 15$, $\mu_S = 0.25$ and

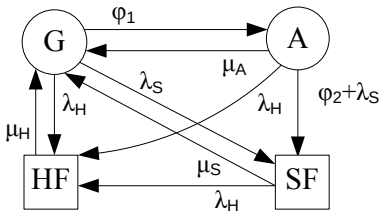


Figure 5. The state transition diagram for the database server.

$\mu_H = 0.04\ (h^{-1})$. The rate transition matrix for the server is

$$\mathbf{Q}_{\text{server}} = \begin{pmatrix} -(\varphi_1 + \lambda_S + \lambda_H) & \varphi_1 & \lambda_S & \lambda_H \\ \mu_A & -(\mu_A + \varphi_2 + \lambda_S + \lambda_H) & \varphi_2 + \lambda_S & \lambda_H \\ \mu_S & 0 & -(\mu_S + \lambda_H) & \lambda_H \\ \mu_H & 0 & 0 & -\mu_H \end{pmatrix}$$
$$= \begin{pmatrix} -0.0073 & 0.002 & 0.005 & 0.0003 \\ 15 & -75.0053 & 60.005 & 0.0003 \\ 0.25 & 0 & -0.2503 & 0.0003 \\ 0.04 & 0 & 0 & -0.04 \end{pmatrix}.$$

In this example we chose not to demonstrate how to incorporate attacker behavior in $\mathbf{Q}_{\text{server}}$, but assume that a game model, Γ_{server}, has already been applied to obtain the attack probability parts of φ_1 and φ_2. The result from the game model affects the numerical values of the predicted system measures in the final step of the system assessment model but is otherwise not substantially important for understanding the functionality of the prediction architecture. The reader is therefore referred to the previously published paper [15] for a more illustrative case-study on this particular topic.

7.2 The Monitoring System

As illustrated in Fig. 4, the agent collects and interprets data from both the NIDS and the HSS. The observations are then forwarded to the estimator (not illustrated in the figure). In this example the observation symbol set is $V = \{g, a, sf, hf\}$ where symbol g is an indication of system state G, symbol a an indication of state A, and so forth. In this paper we do not focus on *how* the NIDS and HSS data is interpreted; we simple assume that the agent is able to map sensor data into symbols representing states.

The HMM representing this monitoring system is defined by the three-tuple $\Lambda = (\mathbf{P}_{\text{server}}, \mathbf{X}^1_{\text{server}}, \mathbf{O}_{\text{server}})$. The sampling interval is fixed to $\Delta = 15$ min. By using (10)

we compute the one-step transition probability matrix as

$$\mathbf{P}_{\text{server}} = \begin{pmatrix} p_{G,G} & p_{G,A} & p_{G,SF} & p_{G,HF} \\ p_{A,G} & p_{A,A} & p_{A,SF} & p_{A,HF} \\ p_{SF,G} & p_{SF,A} & p_{SF,SF} & p_{SF,HF} \\ p_{HF,G} & p_{HF,A} & p_{HF,SF} & p_{HF,HF} \end{pmatrix}$$
$$= \begin{pmatrix} 0.998 & 2.66 \cdot 10^{-5} & 1.58 \cdot 10^{-3} & 7.46 \cdot 10^{-5} \\ 0.246 & 6.49 \cdot 10^{-6} & 0.754 & 7.46 \cdot 10^{-5} \\ 0.061 & 1.53 \cdot 10^{-6} & 0.939 & 7.46 \cdot 10^{-5} \\ 9.94 \cdot 10^{-3} & 2.51 \cdot 10^{-7} & 7.85 \cdot 10^{-6} & 0.990 \end{pmatrix}.$$

As the initial sampling state distribution we use the steady state probabilities of the system

$$\mathbf{X}^1_{\text{server}} = \{X^1_G, X^1_A, X^1_{SF}, X^1_{HF}\} = \{0.967, 2.58 \cdot 10^{-5}, 2.55 \cdot 10^{-2}, 7.44 \cdot 10^{-3}\},$$

found by solving (5)-(6). One can see that the database is very likely to be in state G when the sampling process starts. The observation symbol probability matrix for the database server is

$$\mathbf{O}_{\text{server}} = \begin{pmatrix} o_G(g) & o_G(a) & o_G(sf) & o_G(hf) \\ o_A(g) & o_A(a) & o_A(sf) & o_A(hf) \\ o_{SF}(g) & o_{SF}(a) & o_{SF}(sf) & o_{SF}(hf) \\ o_{HF}(g) & o_{HF}(a) & o_{HF}(sf) & o_{HF}(hf) \end{pmatrix} = \begin{pmatrix} 0.80 & 0.10 & 0.06 & 0.04 \\ 0.30 & 0.55 & 0.10 & 0.05 \\ 0.08 & 0.02 & 0.70 & 0.20 \\ 0.01 & 0.01 & 0.10 & 0.88 \end{pmatrix}.$$

Since both $o_G(sf), o_G(hf), o_A(sf), o_A(hf) \neq 0$ and $o_{SF}(g), o_{SF}(a), o_{HF}(g), o_{HF}(a) \neq 0$ in $\mathbf{O}_{\text{server}}$, one can see that even though the sensors in this case study have relatively low false-positive and false-negative rates there is still room for the possibility of misleading observations. Note that we use a single observation symbol probability matrix to represent the trustworthiness of the (merged) data from both the NIDS and the HSS sensor. See e.g. [1] for an example of round robin sampling of sensors, or [5] for an algorithm for optimal selection of data from multiple sensors.

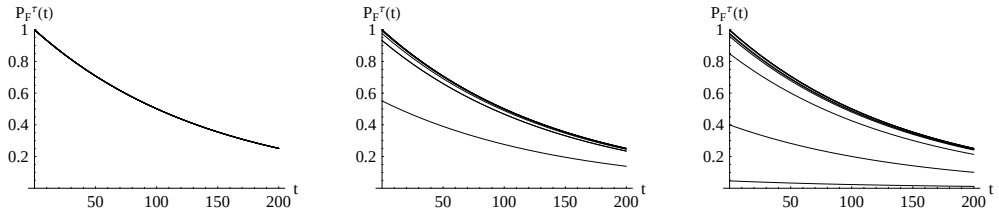
7.3 Simulation Results

To evaluate the effectiveness of the proposed prediction method, we simulate the following three different observation sequences

$$\begin{aligned} Y_1 &= (g, g, g, a, g, g, g, g, g, g), \\ Y_2 &= (g, g, a, sf, a, sf, sf, g, g, g), \\ Y_3 &= (g, g, sf, g, sf, g, hf, hf, hf, hf). \end{aligned}$$

The purpose of the first simulated sequence (Y_1) is to demonstrate how the prediction process reacts to a single "attack" warning observation (a) that are preceded and followed by a number of "good" observations (g). The second simulation (Y_2) demonstrates how the prediction algorithm reacts to alternate a and sf observations. The third sequence (Y_3) simulates a number of software failure observations that are indicated to be repaired, and finally followed by a protracted hardware failure.

The $P_F^\tau(t)$ functions First, we discuss the predicted $P_F^\tau(t)$ functions. The results from the three simulations are depicted in Fig. 6.



(a) Simulating Y_1. (b) Simulating Y_2. (c) Simulating Y_3.

Figure 6. An overview of the $P_F^\tau(t)$ graphs from the three simulations. Each subfigure depicts the predicted $P_F(t)$ graphs at the sampling instants $\tau = 1, \ldots, 10$.

From Fig. 6(a) it appears that $P_F^\tau(0)$ is very close to 1 for most samples $\tau = 1, \ldots, 10$. Fig. 7 shows a more detailed view of the simulation results from Y_1. One can see that the $P_F^\tau(t)$ graph is lower for the first sample, i.e. when $\tau = 1$. This is because, in accordance to $\mathbf{X}^1_{\text{server}}$, the server is assumed to be in state G with only 96.7% certainty when the sampling process starts. As the estimator receives more g symbols, the estimated probability of state G will rise, and hence, the corresponding $P_F^\tau(t)$ graph will rise. Note that since the fourth observation $y_4 = a$ in the first simulation, the $P_F^4(t)$ graph will be slightly lower than the subsequent predicted graphs.

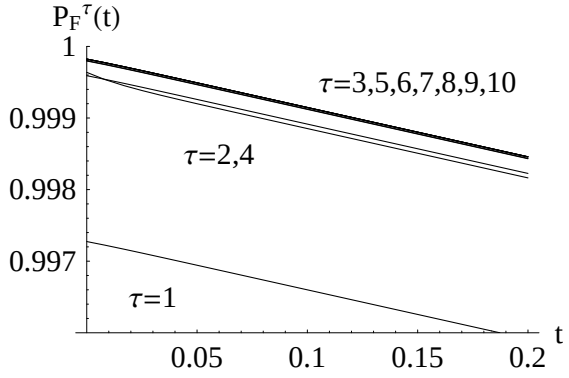


Figure 7. A closer look at the $P_F^\tau(t)$ graphs when simulating $Y_1 = (g, g, g, a, g, g, g, g, g, g)$.

As can be seen from Fig. 6(b) and Fig. 8, $P_F^\tau(t)$ for the second simulation will be quite high until the estimator receives the first *sf* symbol at sampling instant $\tau = 4$. Even though the next observation ($y_5 = a$) will rise the predicted graph, the next observation after that ($y_6 = sf$) will lower it even more. The lowest graph of them all will appear at sampling instant $\tau = 7$, which is due to the two successive *sf* observations. Note that for the same reason $P_F^7(0) \approx 0.55$, since the system with a high probability already has failed.

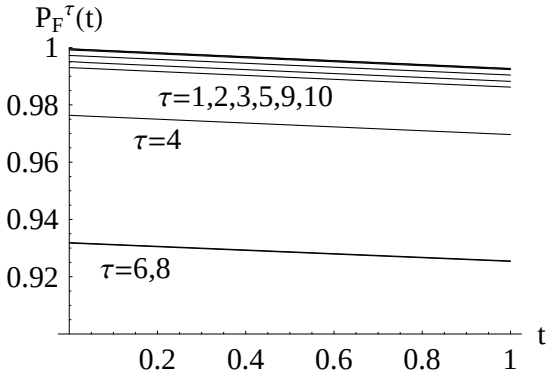
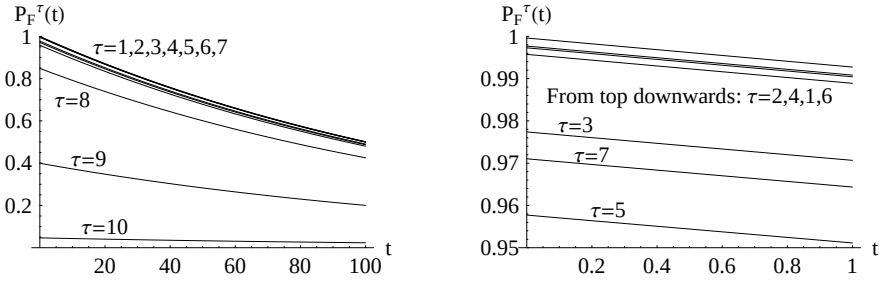


Figure 8. A closer look at the $P_F^\tau(t)$ graphs when simulating $Y_2 = (g, g, a, sf, a, sf, sf, g, g, g)$.

The result from the third simulation (Fig. 6(c) and Fig. 9) shows that the alternating g and sf observations will give rise to corresponding $P_F^\tau(t)$ graphs. As the agent starts to receive hf (hardware failure) symbols, the predicted $P_F^\tau(t)$ graphs will decrease even more. Also $P_F^\tau(0) \to 0$ as $\tau \to 10$.



(a) An overview of the predicted graphs.

(b) A closer look a at the top seven graphs.

Figure 9. A closer look at the $P_F^\tau(t)$ graphs when simulating $Y_3 = (g, g, sf, g, sf, g, hf, hf, hf, hf)$.

As indicated in Fig. 6, $P_F^\tau(t) \to 0$ as $t \to \infty$ for all simulated graphs, i.e. even though the estimated state during sampling is likely to be good, the system will sooner of later fail.

The $MTNF$ measures. The predicted $MTNF^\tau$ measures, together with the corresponding $P_F^\tau(0)$'s, are depicted in Fig. 10. During the first simulation (Y_1), the predicted $MTNF$ measure drops as the a symbol is received (at sampling instant $\tau = 4$), but returns to the same level as more g symbols are received. The corresponding $P_F^\tau(0)$ graph indicates that the predicted $MTNF$ measures are reliable ($P_F^\tau(0) \approx 1$), with an exception for the first sample ($P_F^1(0) = 0.998$).

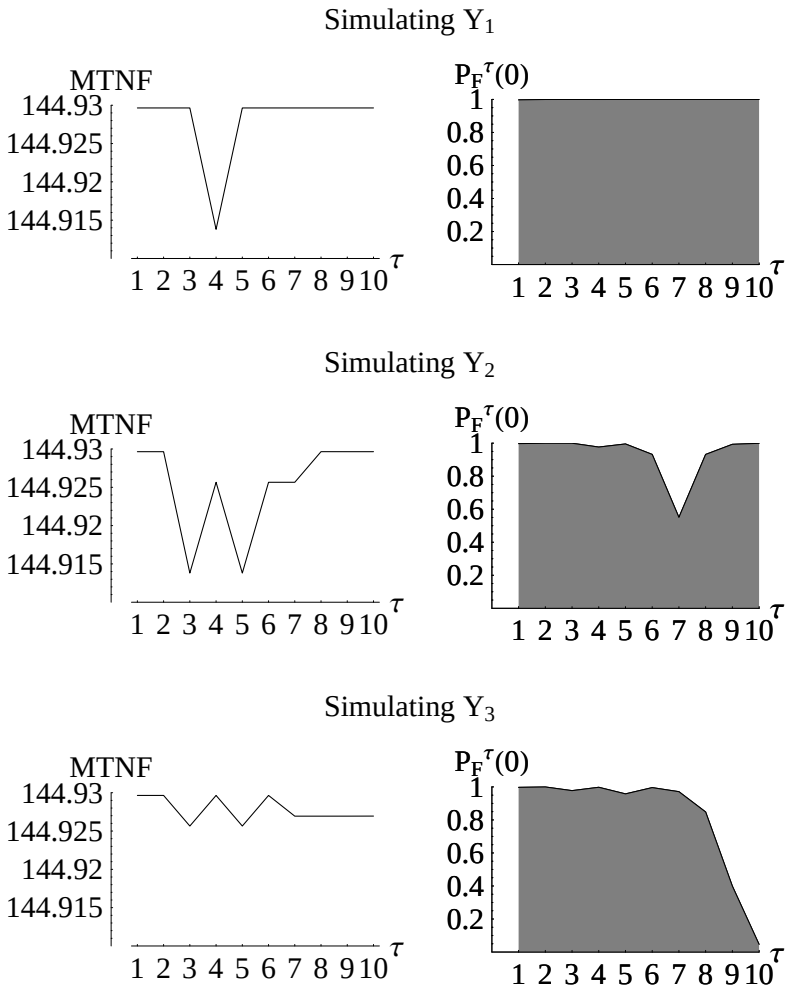


Figure 10. The $MTNF$ measures together with the corresponding $P_F^\tau(0)$'s.

From the second simulation we observe that since $y_3 = y_5 = a$, the predicted $MTNF$ measure will be slightly lower at $\tau = 3$ and $\tau = 5$. Interestingly, $MTNF$ will rise at $\tau = 4$, $\tau = 6$ and $\tau = 7$, even though $y_4 = y_6 = y_7 = sf$. The corresponding $P_F^\tau(0)$ graph explains this phenomenon; since the graph is lower at $\tau = 4, 6, 7$ the system may already be in a failed state at these particular sampling instants.

The results from the third simulation indicates that sf and hf symbols will lower the predicted $MTNF$ measures. Since the simulated trace ends with four subsequent hf symbols (at $\tau = 7, 8, 9, 10$), $P_F^\tau(0) \to 0$ as $\tau \to 10$.

8. Concluding Remarks

This paper presents a framework for integrated security and dependability assessment of computing systems. By using data provided by a monitoring architecture, the current system state and its future behavior can be predicted in real-time. The proposed model for computing system measures is based on Markov analysis, where model parameters are determined by a game theoretic analysis of attacker behavior. To demonstrate the feasibility of the proposed prediction architecture, we performed three simulation experiments for a small case study.

The stochastic modeling approach used in this paper relies on a few assumptions. First, we assume that the security and dependability behavior of the system can be modeled as a Markov process, which means that the conditional probability distribution of future states of the process depends only upon the current state. Even though it is very common to assume these properties when modeling and analyzing systems in a traditional dependability context (considering accidental failures only), it is not (yet) a well established practice in the security community. Second, the HMM approach relies on independent observations, which means that the observations that a sensor produces depend on the current system state only, and not on any previous observations. The main drawback of this approach is that, because security indications and alerts can be highly correlated, the sampling interval must be large enough so that the observations received by the estimator can be considered independent, for the model to be valid. Of course the exact lower limit for the sampling interval will depend on the particular system that are to be assessed, and on the types of sensors that monitors the system. As an example, for the database server in the case study 15 minutes was suggested as a reasonable sampling interval.

The case study used to demonstrate the approach in Section 7 is kept simple for illustration. In the future we plan to model and simulate the security and dependability assessment process for a more extensive example. A validation by a prototype system also remains.

References

[1] A. Årnes, K. Sallhammar, K. Haslum, T. Brekne, M. E. Gaup Moe, and S. J. Knapskog. Real-time Risk Assessment with Network Sensors and Intrusion Detection Systems. In *International Conference on Computational Intelligence and Security (CIS)*, Dec 2005.

[2] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, 1:11–33, January-March 2004.

[3] K. B. B. Madan, K. Vaidyanathan Goseva-Popstojanova, and K. S. Trivedi. A method for modeling and quantifying the security attributes of intrusion tolerant systems. In *Performance Evaluation*, volume 56, 2004.

[4] John A. Buzacott. Markov approach to finding failure times of repairable systems. *IEEE Transactions on Reliability*, R-19:128–134, November 1970.

[5] R. Evans, V. Krishnamurthy, G. Nair, and L. Sciacca. Networked sensor management and data rate control for tracking maneuvering targets. *IEEE Transactions on Signal Processing*, 53:1979–1991, June 2005.

[6] Bjarne E. Helvik. *Dependable Computing Systems and Communication Networks, Design and Evaluation*. Draft lecture notes (256 p.), Department of Telematics, NTNU, Nov. 2003.

[7] Bjarne E. Helvik. Statediagrams.m; a Mathematica package for dependability analysis of systems by CTMC. Unpublished NTNU, 2002.

[8] ISO/IEC 13335: Information Technology - Guidelines for the management of IT Security. http://www.iso.org.

[9] S. Jha, O. Sheyner, and J. Wing. Two formal analyses of attack graphs. In *Proceedings of the 2002 Computer Security Foundations Workshop*, 2002.

[10] B. Littlewood, S. Brocklehurst, N. Fenton, P. Mellor, S. Page, D. Wright, J. Dobson, McDermid J., and D. Gollmann. Towards operational measures of computer security. *Journal of Computer Security*, 2:211–229, Oct 1993.

[11] David M. Nicol, William H. Sanders, and Kishor S. Trivedi. Model-based evaluation: From dependability to security. *IEEE Transactions on Dependable and Secure Computing*, 1:48–65, January-March 2004.

[12] R. Ortalo and Y. Deswarte. Experimenting with quantitative evaluation tools for monitoring operational security. *IEEE Transactions on Software Engineering*, 25(5):633–650, Sept/Oct 1999.

[13] Lawrence R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Readings in speech recognition*, pages 267–296, 1990.

[14] Sheldon M. Ross. *Introduction to Probability Models*. Academic Press, 8th edition, 2003.

[15] Karin Sallhammar, Bjarne E. Helvik, and Svein J Knapskog. Towards a stochastic model for integrated security and dependability evaluation. In *Proceedings of the First International Comference on Availability, Reliability and Security (AReS)*, 2006.

[16] Mehmet Savsar and Fawaz S. Al-Anzi. Reliability of data allocation on a centralized server configuration with distributed servers. *The Computer Journal*, 49:258–267, 2005.

[17] F. Stevens, T. Courtney, S. Singh, A. Agbaria, J. F. Meyer, W. H. Sanders, and P. Pal. Model-based validation of an intrusion-tolerant information system. In *Proceedings of the 23rd Symposium on Reliable Distributed Systems (SRDS 2004)*, Oct 18-20 2004.

[18] Wei Wei, Bing Wang, and Don Towsley. Continuous-time Hidden Markov Models for Network Performance Evaluation. *Performance Evaluation*, 49:129–146, 2002.

III

THESIS APPENDIX

Appendix: Scaling the Forward Variables

Recall the definition of the *forward variable* in PAPER E and F

$$\alpha_t(i) = P(y_1 \dots y_t, x_t = s_i|\lambda) = \begin{cases} q_i(y_1)\pi_i, & t = 1, \\ q_i(y_t)\sum_{j=1}^{N} \alpha_{t-1}(j)p_{ji}, & t > 1. \end{cases} \tag{A.1}$$

For large observation sequences ($t > 100$) these probabilities get really small. To keep the computations within the precision range of the computer, the forward variables need to be scaled. The basic scaling procedure that has been adopted is to multiply $\alpha_t(i)$ by a scaling coefficient that is independent of i

$$\hat{\alpha}_t(i) = C_t\alpha_t(i), \tag{A.2}$$

where $C_t = \left(\sum_{i=1}^{N} \alpha_t(i)\right)^{-1}$, $\forall i$. By applying (A.2) one makes sure that the forward variables are scaled so that they sum to one at any given time t. Even though it is shown in [Rab90] that these scaling coefficients cancel out during the parameter reestimation procedure, it has not been shown for the state estimation procedure used in PAPER E and F. Moreover, Rabiner makes use of a *backward variable*, which is not defined in the state estimation algorithms at all. The purpose of this appendix is therefore to clarify the scaling procedure that has to be used when implementing the algorithms in PAPER E and F.

The Scaling Procedure

To explain how the forward variables are scaled, one can distinguish between the following three variables

- $\alpha_t(i)$ the unscaled forward variable,
- $\hat{\alpha}_t(i)$ the scaled forward variable,
- $\hat{\hat{\alpha}}_t(i)$ the local version of the forward variable before scaling.

The purpose of defining the local (unscaled) $\hat{\hat{\alpha}}_t(i)$ variable is to facilitate the implementation of the scaling procedure, while avoiding messing up the notation. The scaling procedure that has been implemented is then

$$\hat{\alpha}_t(i) = c_t\hat{\hat{\alpha}}_t(i) = (\sum_{i=1}^{N} \hat{\hat{\alpha}}_t(i))^{-1}\hat{\hat{\alpha}}_t(i), \tag{A.3}$$

such that $\sum_{i=1}^{N} \hat{\alpha}_t(i) = 1, \forall t$. To compute the local forward variable $\hat{\hat{\alpha}}_t(i)$ one can apply (A.1) recursively, and then use the local variables to compute the scaled ones.

- Initialization at $t = 1$

$$\hat{\hat{\alpha}}_1(i) = q_i(y_1)\pi_i, \tag{A.4}$$

$$\hat{\alpha}_1(i) = c_1\hat{\hat{\alpha}}_1(i) = \frac{\hat{\hat{\alpha}}_1(i)}{\sum_{i=1}^{N} \hat{\hat{\alpha}}_1(i)}. \tag{A.5}$$

- Iteration for $t > 1$

$$\hat{\hat{\alpha}}_t(i) = q_i(y_t)\sum_{j=1}^{N} \hat{\alpha}_{t-1}(j)p_{ji}, \tag{A.6}$$

$$\hat{\alpha}_t(i) = c_t\hat{\hat{\alpha}}_t(i) = \frac{\hat{\hat{\alpha}}_t(i)}{\sum_{i=1}^{N} \hat{\hat{\alpha}}_t(i)}. \tag{A.7}$$

Proof of Correctness

It is easy to see that because $\hat{\tilde{\alpha}}_1(i) = q_i(y_1)\pi_i = \alpha_1(i)$, then $\hat{\alpha}_1(i)$ will be correctly scaled, according to (A.2). For $t > 1$, note that the implemented scaling procedure multiplies each local variable $\hat{\tilde{\alpha}}_t(i)$ with c_t, where $c_t = (\sum_{i=1}^{N} \hat{\tilde{\alpha}}_t(i))^{-1}$. Hence, by using (A.6)-(A.7) and re-arranging terms

$$\hat{\alpha}_t(i) = \frac{\hat{\tilde{\alpha}}_t(i)}{\sum_{i=1}^{N} \hat{\tilde{\alpha}}_t(i)} = \frac{q_i(y_t)\sum_{j=1}^{N} \hat{\alpha}_{t-1}(j)p_{ji}}{\sum_{i=1}^{N} q_i(y_t)\sum_{j=1}^{N} \hat{\alpha}_{t-1}(j)p_{ji}} = \frac{\sum_{j=1}^{N} \hat{\alpha}_{t-1}(j)p_{ji}q_i(y_t)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \hat{\alpha}_{t-1}(j)p_{ji}q_i(y_t)}. \quad \text{(A.8)}$$

But since $\hat{\tilde{\alpha}}_1(i) = \alpha_1(i)$ one can write

$$\hat{\alpha}_{t-i}(j) = c_1 c_2 \dots c_{t-1}\alpha_{t-1}(j) = \big(\prod_{\tau=1}^{t-1} c_\tau\big)\alpha_{t-1}(j), \quad \text{(A.9)}$$

which is true by induction. Hence, the scaled forward variable in (A.8) becomes

$$\begin{aligned}\hat{\alpha}_t(i) &= \frac{\sum_{j=1}^{N} \hat{\alpha}_{t-1}(j)p_{ji}q_i(y_t)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \hat{\alpha}_{t-1}(j)p_{ji}q_i(y_t)} = \frac{\sum_{j=1}^{N} \big(\prod_{\tau=1}^{t-1} c_\tau\big)\alpha_{t-1}(j)p_{ji}q_i(y_t)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \big(\prod_{\tau=1}^{t-1} c_\tau\big)\alpha_{t-1}(j)p_{ji}q_i(y_t)} \\ &= \frac{\sum_{j=1}^{N} \alpha_{t-1}(j)p_{ji}q_i(y_t)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_{t-1}(j)p_{ji}q_i(y_t)}.\end{aligned} \quad \text{(A.10)}$$

Using the original definition of the forward variable in (A.1), the scaled variable in (A.10) can be written as

$$\hat{\alpha}_t(i) = \frac{\sum_{j=1}^{N} \alpha_{t-1}(j)p_{ji}q_i(y_t)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_{t-1}(j)p_{ji}q_i(y_t)} = \frac{\alpha_t(i)}{\sum_{i=1}^{N} \alpha_t(i)}, \quad \text{(A.11)}$$

and one can see that the scaled parameter $\hat{\alpha}_t(i)$ satisfies the desired property (A.2) with $C_t = \big(\sum_{i=1}^{N} \alpha_t(i)\big)^{-1}$.

Implementation Issues

Recall how the estimated state probability distribution is computed in PAPER E and F

$$\gamma_t(i) = \frac{\alpha_t(i)}{\sum_{j=1}^{N} \alpha_t(j)}. \quad \text{(A.12)}$$

By looking closer at (A.11) and (A.12), it turns out that

$$\gamma_t(i) = \hat{\alpha}_t(i), \quad \text{(A.13)}$$

i.e., the scaled forward variable at time t is in fact equal the estimated state probability at time t, in the proposed model in PAPER E and F.

Bibliography

[AB03] T. Alpcan and T. Basar. A game theoretic approach to decision and analysis in network intrusion detection. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, Dec 2003.

[AB04] T. Alpcan and T. Basar. A game theoretic analysis of intrusion detection in access control systems. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, Dec 2004.

[ACF+00] J. Allen, A. Christie, W. Fithen, J. McHugh, J. Pickel, and E. Stoner. State of the practice of intrusion detection technologies. Technical Report CMU/SEI-99TR-028, 2000.

[ADD+05] Eric Alata, Marc Dacier, Yves Deswarte, Mohamed Kaaniche, Kostya Kortchinsky, Vincent Nicomette, Van-Hau Pham, and Fabien Pouget. Collection and analysis of attack data based on honeypots deployed on the internet. In *Proceedings of the First Workshop on Quality of Protection (QoP)*, Milan, Italy, September 2005.

[ALR00] A. Avizienis, J.-C. Laprie, and B. Randell. Fundamental concepts of dependability. In *Proceedings of the Third Information Survivability Workshop (ISW-2000)*, October 2000.

[ALRL04] Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, 1:11–33, January-March 2004.

[Aro] Ashish Arora. A statistical analysis of computer attacks using data from the honeynet project. Carnegie Mellon CyLab. http://www.cylab.cmu.edu/.

[ÅSH+05] André Årnes, Karin Sallhammar, Kjetil Haslum, Tønnes Brekne, Marie E. G. Moe, and Svein J. Knapskog. Real-time Risk Assessment with Network Sensors and Intrusion Detection Systems. In *Proceedings of the International Conference on Computational Intelligence and Security (CIS'05)*, Dec 2005.

[ÅSHK06] André Årnes, Karin Sallhammar, Kjetil Haslum, and Svein Johan Knapskog. Real-time Risk Assessment with Network Sensors and Hidden Markov Models. In *Proceedings of the 11th Nordic Workshop on Secure IT-Systems (Nordsec2006)*, Oct 2006.

[ÅVVK06] André Årnes, Fredrik Valeur, Giovanni Vigna, and Richard A. Kemmerer. Using Hidden Markov Models to Evaluate the Risks of Intrusions - System Architecture and Model Validation. In *Proceedings of the 9th International Symposium On Recent Advances In Intrusion Detection (RAID 2006)*, Sep 2006.

[AWK02] P. Ammann, D. Wijesekera, and S. Kaushik. Scalable, graph-based network vulnerability analysis. In *Proceedings of the 5th ACM conference on Computer and communications security*, 2002.

[BBMGPT04] K. B. B. Madan, K. Vaidyanathan Goseva-Popstojanova, and K. S. Trivedi. A method for modeling and quantifying the security attributes of intrusion tolerant systems. In *Performance Evaluation*, volume 56, 2004.

[Bey] BEYOND-THE-HORIZON: Security, Dependability and Trust. http://www.beyond-the-horizon.net/.

[BGFI+98] J. S. Balasubramaniyan, J. O. Garcia-Fernandez, D. Isacoff, E. Spafford, and D. Zamboni. An architecture for intrusion detection using autonomous agents. In *Proceedings of the 14th Annual Computer Security Applications Conference*, pages 13–24. IEEE Computer Society, 1998.

[BLMJ+01] Michel Beaudouin-Lafon, Wendy E. Mackay, Mads Jensen, Peter Andersen, Paul Janecek, Michael Lassen, Kasper Lund, Kjeld Mortensen, Stephanie Munck, Anne Ratzer, Katrine Ravn, Sren Christensen, and Kurt Jensen. CPN/tools: A tool for editing and simulating coloured petri nets - ETAPS tool demonstration related to TACAS. In *LNCS 2031: Tools and Algorithms for the Construction and Analysis of Systems*, pages 574–pp. Springer Verlag, 2001.

[BLOJ94] S. Brocklehurst, B. Littlewood, J. Olovsson, and E. Jonsson. On measurement of operational security. *Aerospace and Electronic Systems Magazine, IEEE*, 9, 1994.

[BMS05] Davide Balzarotti, Mattia Monga, and Sabrina Sicari. Assessing the risk of using vulnerable components. In *Proceedings of the First Workshop on Quality of Protection (QoP)*, Milan, Italy, September 2005.

[But02] S.A. Butler. Security attribute evaluation method: A cost benefit approach. In *Proceedings of the International Conference on Software Engineering*, May 2002.

[Buz70] John A. Buzacott. Markov approach to finding failure times of repairable systems. *IEEE Transactions on Reliability*, R-19:128–134, November 1970.

[CINU05] Ramkumar Chinchani, Anusha Iyer, Hung Q. Ngo, and Shambhu Upadhyaya. Towards a theory of insider threat assessment. In *Proceedings of the International Conference on Dependable Systems and Network (DSN-2005)*, Jun/Jul 2005.

[CMR04] Huseyin Cavusoglu, Birendra Mishra, and Srinivasan Raghunathan. A model for evaluating it security investments. *Communications of the ACM*, 47(7):87–92, 2004.

[CMR05] Olivier Cappé, Eric Moulines, and Tobias Rydén. *Inference in Hidden Markov Models (Springer Series in Statistics)*. Springer, 2005.

[Coh99] Fred Cohen. Managing Network Security - Attack and Defense Strategies. *Network Security Magazine*, July 1999.

[CPN06] CPN Group, University of Aarhus, Denmark. CPN Tools - Computer Tool for Coloured Petri Nets, Sep 08 2006. http://wiki.daimi.au.dk/cpntools/.

[CS95] Mark Crosbie and Gene Spafford. Defending a computer system using autonomous agents. In *Proceedings of the 18th National Information Systems Security Conference*, October 1995.

[DCF05] H. Debar, D. Curry, and B. Feinstein. Intrusion detection message exchange format (IDMEF) – Internet-Draft, 2005.

[dSeSG92] Edmundo de Souza e Silva and H. Richard Gail. Performability analysis of computer systems: from model specification to solution. *Performance Evaluation*, 14:157–196, 1992.

[EFL+97] B. Ellison, D. Fisher, R. Linger, H. Lipson, T. Longstaff, and N. Mead. Survivable network systems: An emerging discipline. Technical report, CMU/SEI-97-TR-013, Software Engineering Institute, Carnegie Mellon University, Nov 1997.

[Eur] EuroNGI WP.JRA.6.3: Creation of Trust by Advanced Security Concepts. http://eurongi.enst.fr/.

[EWI] EWICS: European Workshop on Industrial Computer Systems Reliability, Safety and Security. http://www.ewics.org/.

[GGK+03] Michael Greenwald, Carl A. Gunter, Bjorn Knutsson, Andre Scedrov, Jonathan M. Smith, and Steve Zdancewic. Computer Security is Not a Science (but it should be). In *Proceedings of the Large-Scale Network Security Workshop*, 2003.

[Gib92] R. Gibbons. *Game Theory for Applied Economists*. Princeton University Press, 1992.

[GK04] Ashish Gehani and Gershon Kedem. Rheostat: Real-time risk management. In *Recent Advances in Intrusion Detection: 7th International Symposium, RAID 2004, Sophia Antipolis, France, September 15-17, 2004. Proceedings*, pages 296–314. Springer, 2004.

[GLR+03] Vishu Gupta, Vinh Lam, HariGovind V. Ramasamy, William H. Sanders, and Sankalp Singh. Stochastic Modeling of Intrusion-Tolerant Server Architectures for Dependability and Performance Evaluation. Technical Report UILU-ENG-03-2227 (CRHC-03-13), University of Illinois at Urbana - Champaign Coordinated Science Laboratory, Dec 2003.

[GPWW+01] Katerina Goseva-Popstojanova, Feiyi Wang, Rong Wang, Fengmin Gong, Kalyanaraman Vaidyanathan, Kishor Trivedi, and Balamurugan Muthusamy. Characterizing Intrusion Tolerant Systems Using a State Transition Model. In *DARPA Information Survivability Conference and Exposition (DISCEX II)*, volume 2, 2001.

[HÅ06] Kjetil Haslum and André Årnes. Multisensor Real-time Risk Assessment Using Continuous-time Hidden Markov Models. In *Proceedings of the International Conference on Computational Intelligence and Security (CIS'06)*, Nov 2006.

[Hac05] The development of a meaningful hacker taxonomy: A two dimensional approach. Technical report, CERIAS Tech Report 2005-43, 2005. Centre for Education and Research in Information Assurance and Security, Purdue University.

[Har68] John C. Harsanyi. Games with incomplete information played by bayesian players. In *Management Science 14*, 1967-1968.

[Hel] Bjarne E. Helvik. *Dependable Computing Systems and Communication Networks, Design and Evaluation*. Draft lecture notes (256 p.), Department of Telematics, NTNU, Nov. 2003.

[Hel02] Bjarne E. Helvik. Statediagrams.m; a Mathematica package for dependability analysis of systems by CTMC. Unpublished NTNU, 2002.

[HPW03] Michael Howard, John Pincus, and Jeannette M. Wing. Measuring relative attack surfaces. In *Proceedings of Workshop on Advanced Developments in Software and Systems Security*, Dec 2003.

[HR04] Charles A. Holt and A. E. Roth. The Nash equilibrium: A perspective. In *Proceedings of the National Academy of Sciences*, volume 101, March 2004.

[HS05] Siv Hilde Houmb and Karin Sallhammar. Modeling system integrity of a security critical system using colored petri nets. In *Safety and Security Engineering*, pages 3–12. Wit Press, Jun 2005.

[HSK07] Bjarne E. Helvik, Karin Sallhammar, and Svein J. Knapskog. *Integrated Dependability and Security Evaluation using Game Theory and Markov Models*. In Qian et al. [QJTK07], to be published in 2007.

[HWH+03] Guy Helmer, Johnny S. K. Wong, Vasant G. Honavar, Les Miller, and Yanxin Wang. Lightweight agents for intrusion detection. *J. Syst. Softw.*, 67(2):109–122, 2003.

[IFI] IFIP WG 10.4 on Dependable Computing and Fault Tolerance. http://www.dependability.org/wg10.4/.

[Ins04] Insider threat study: Illicit cyber activity in the banking and finance sector. Technical report, CERT/United States Secret Service, 2004.

[ISOa] ISO/IEC 27005: Guidelines for Information Security Risk Management. Under development.

[ISOb] ISO/IEC 27007: Information Security Management Metrics and Measurement. Under development.

[ISO99] ISO 15408: Common Criteria for Information Technology Security Evaluation, 1999. http://www.commoncriteria.org.

[ISO04] ISO/IEC 13335: Information technology - Security techniques - Management of information and communications technology security, 2004. http://www.iso.org.

[ISO05] ISO/IEC 17799:2005. Information technology - Security techniques - Code of practice for information security management, 2005. http://www.iso.org.

[Jen97a] K. Jensen. *Coloured Petri Nets (2nd ed.): Basic Concepts, Analysis Methods and Practical Use: Volume 1*. Berlin, Heidelberg, New York: Springer-Verlag, 2 edition, 1997. In: EATCS Monographs on Theoretical Computer Science.

[Jen97b] K. Jensen. *Coloured Petri Nets (2nd ed.): Basic Concepts, Analysis Methods and Practical Use: Volume 2*. Berlin, Heidelberg, New York: Springer-Verlag, 2 edition, 1997. In: EATCS Monographs on Theoretical Computer Science.

[Jen97c] K. Jensen. *Coloured Petri Nets (2nd ed.): Basic Concepts, Analysis Methods and Practical Use: Volume 3*. Berlin, Heidelberg, New York: Springer-Verlag, 2 edition, 1997. In: EATCS Monographs on Theoretical Computer Science.

[JO92] E. Jonsson and T. Olovsson. On the integration of security and dependability in computer systems. In *IASTED Int'l Conf. Reliability, Quality Control and Risk Assessment*, pages 93–97. Washington, Nov. 4–6 1992.

[JO97] E. Jonsson and T. Olovsson. A quantitative model of the security intrusion process based on attacker behavior. *IEEE Transactions on Software Eng.*, 4(25):235, April 1997.

[Jon98] E. Jonsson. An integrated framework for security and dependability. In *Proceedings of the New Security Paradigms Workshop 1998*, Sep. 22–25 1998.

[JSL99] E. Jonsson, L. Stromberg, and S. Lindskog. On the functional relation between security and dependability impairments. In *Proceedings of the New Security Paradigms Workshop 1999*, Sep. 22–24 1999.

[JSW02a] S. Jha, O. Sheyner, and J. Wing. Minimization and reliability analysis of attack graphs. Technical report, CMU-CS-02-109, School of Computer Science, Carnegie Mellon University, 2002.

[JSW02b] S. Jha, O. Sheyner, and J. Wing. Two formal analyses of attack graphs. In *Proceedings of the 2002 Computer Security Foundations Workshop*, 2002.

[JTM01] S. Jha, K. Tan, and R.A. Maxion. Markov chains, classifiers, and intrusion detection. In *Proceedings of the Computer Security Foundations Workshop (CSFW)*, June 2001.

[KBMP99] K. Kanoun, M. Borrel, T. Morteveille, and A. Peytavin. Availability of CAUTRA, a subset of the french air traffic control system. *IEEE Transactions on Computers*, 48(5):528–535, 1999.

[Lap92] Jean-Claude Laprie. *Dependability: Basic Concepts and Associated Terminology*, volume 5. Springer, 1992.

[LB05] Stefan Lindskog and Anna Brunstrom. Design and implementation of a tunable encryption service for networked applications. In *Proceedings of the First IEEE/CREATE-NET Workshop on Security and QoS in Communication Networks (SecQoS 2005)*, Sep 2005.

[LBF+93] B. Littlewood, S. Brocklehurst, N. Fenton, P. Mellor, S. Page, D. Wright, J. Dobson, McDermid J., and D. Gollmann. Towards operational measures of computer security. *Journal of Computer Security*, 2:211–229, Oct 1993.

[LEH+97] T. Longstaff, J. Ellis, S. Hernan, H. Lipson, R. McMillan, L. Pesante, and D. Simmel. Security of the Internet. *The Froehlich/Kent Encyclopedia of Telecommunications*, 15:231–255, 1997.

[Lin05] Stefan Lindskog. *Modeling and Tuning Security from a Quality of Service Perspective*. PhD thesis, Department of computer science and engineering, Chalmers University of Technology, 2005.

[LJ02] Stefan Lindskog and Erland Jonsson. Adding security to qos architectures. In *Proceedings of the SSGRR 2002's conference*, Jul/Aug 2002.

[LLBFH05] Reine Lundin, Stefan Lindskog, Anna Brunstrom, and Simone Fischer-Hubner. Using guesswork as a measure for confidentiality of selectively encrypted messages. In *Proceedings of the First Workshop on Quality of Protection (QoP 2005)*, Sep 2005.

[LSHJ04] Stefan Lindskog, Johan Strandbergh, Mikael Hackman, and Erland Jonsson. A content-independent scalable encryption model. In *Proceedings of the 2004 International Conference on Computational Science and its Applications (ICCSA 2004)*, May 2004.

[LT04] Yun Liu and Kishor S. Trivedi. A general framework for network survivability quantification. In *Proceedings of the 12th GI/ITG Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB)*, Sep 2004.

[Lun88] Teresa F. Lunt. Automated audit trail analysis and intrusion detection: A survey. In *Proceedings of the 11th National Computer Security Conference*, Baltimore, MD, 1988.

[LW02] K. Lye and J. M. Wing. Game strategies in network security. In *Proceedings of the 15th IEEE Computer Security Foundations Workshop*, 2002.

[LW05] Kong-wei Lye and Jeannette M. Wing. Game strategies in network security. *International Journal of Information Security*, 4(1-2):71–86, 2005.

[LZ03] Peng Liu and Wanyu Zang. Incentive-based modeling and inference of attacker intent, objectives, and strategies. In *Proceedings of the 10th ACM conference on computer and communication security*, pages 179–189, 2003.

[MAF] MAFTIA: Malicious and Accidental Fault Tolerance for Internet Applications. IST Research Project IST-1999-11583. http://www.maftia.org/.

[McD05] J. McDermott. Attack-Potential-Based Survivability Modeling for High-Consequence Systems. In *IWIA '05: Proceedings of the Third IEEE International Workshop on Information Assurance (IWIA'05)*, 2005.

[Mea95] Catherine Meadows. Applying the dependability paradigm to computer security. In *Proceedings of the 1995 New Security Paradigms Workshop*, Aug. 22–25 1995.

[Mey89] J. F. Meyer. Performability evaluation of telecommunication networks. *Teletraffic Science*, 12:1163–1172, 1989.

[MHL94] B. Mukherjee, L. T. Heberlein, and K. Levitt. Network intrusion detection. *IEEE Network*, 8(3):26–41, May/June 1994.

[MKF03] J. McDermott, A. Kim, and J. Froscher. Merging paradigms of survivability and security: Stochastic faults and designed faults. In *Proceedings of the New Security Paradigms Workshop 2003*, 18 August 2003.

[MM99] C. Meadows and J. McLean. Security and Dependability: Then and Now. In *Computer Security, Dependability, and Assurance: From Needs to Solutions*, pages 166–170. IEEE Computer Society Press, 1999.

[MN03] Michael G. Merideth and Priya Narasimhan. Metrics for evaluation of proactive and reactive survivability. In *Supplement of the 2003 International Conference on Dependable Systems and Networks. (DSN-2003), Fast Abstracts*, June 2003.

[MT95] M. Malhotra and K. S. Trivedi. Dependability modeling using Petri-net based models. *IEEE Transactions on Reliability*, 44(3):428–440, September 1995.

[MT04] McLennan Andrew M. McKelvey, Richard D. and Theodore L. Turocy. Gambit: Software tools for game theory, version 0.97.0.6, 2004. http://econweb.tamu.edu/gambit/.

[Mur89] Tadao Murata. Petri nets: Properties, analysis and applications. In *Proceedings of the IEEE*, pages 541–580, April 1989.

[MVT02] B.B. Madan, K. Vaidyanathan, and K.S. Trivedi. Modeling and quantification of security attributes of software systems. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN'02)*, 2002.

[MW04] Pratyusa Manadhata and Jeannette M. Wing. Measuring a system's attack surface. Technical report, CMU-CS-04-102, School of Computer Science, Carnegie Mellon University, Jan 2004.

[MW05] Pratyusa Manadhata and Jeannette M. Wing. An attack surface metric. Technical report, CMU-CS-05-155, School of Computer Science, Carnegie Mellon University, Jul 2005.

[NIoST03] NIST National Institute of Standards and Technology. NIST 800-55: security metrics guide for information technology systems, 2003.

[NN99] Stephen Northcutt and Stephen Northcult. *Network Intrusion Detection: An Analyst's Handbook*. New Riders Publishing, Thousand Oaks, CA, USA, 1999.

[NST04] David M. Nicol, William H. Sanders, and Kishor S. Trivedi. Model-Based Evaluation: From Dependability to Security. *IEEE Transactions on Dependable and Secure Computing*, 1:48–65, January-March 2004.

[ODK99] R. Ortalo, Y. Deswarte, and M. Kaaniche. Experimenting with quantitative evaluation tools for monitoring operational security. *IEEE Transactions on Software Engineering*, 25(5):633–650, Sept/Oct 1999.

[OMSH03] Dirk Ourston, Sara Matzner, William Stump, and Bryan Hopkins. Applications of hidden markov models to detecting multi-stage network attacks. In *Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS)*, 2003.

[Owe01] G. Owen. *Game Theory*. Academic Press, 3rd edition, 2001.

[Pe01] David Powell and Robert Stroud (eds.). Malicious- and accidental-fault tolerance for internet applications - Conceptual model and architecture, 2001.

[Pro04] The Honeynet Project. *Know Your Enemy*. Addison-Wesley, 2 edition, 2004.

[PS98] C. Philips and L. Swiler. A graph-based system for network-vulnerability analysis. In *Proceedings of the New Security Paradigms Workshop*, 1998.

[QJTK07] Yi Qian, James Joshi, David Tipper, and Prashant Krishnamurthy, editors. *Information Assurance: Dependability and Security in Networked Systems*. Morgan Kaufmann Publishers, an imprint of ELSEVIER, INC., to be published in 2007.

[Rab90] Lawrence R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Readings in speech recognition*, pages 267–296, 1990.

[RAS96] A. Puliafito R. A. Sahner, K. S. Trivedi. *Performance and Reliability Analysis of Computer Systems*. Kluwer Academic Publishers, 1996.

[Ros03] S. M. Ross. *Introduction to Probability Models*. Academic Press, 8 edition, 2003.

[RS97] M.K. Reiter and S.G. Stubblebine. Toward Acceptable Metrics of Authentication. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 10–20, 1997.

[Sah05] Mehmet Sahinoglu. Security meter: A practical decision-tree model to quantify risk. *IEEE Security & Privacy*, pages 18–24, May/June 2005.

[SBD+91] Steven R. Snapp, James Brentano, Gihan V. Dias, Terrance L. Goan, L. Todd Heberlein, Che lin Ho, Karl N. Levitt, Biswanath Mukherjee, Stephen E. Smaha, Tim Grance, Daniel M. Teal, and Doug Mansur. DIDS (distributed intrusion detection system) - motivation, architecture, and an early prototype. In *Proceedings of the 14th National Computer Security Conference*, pages 167–176, Washington, DC, 1991.

[SCCC+96] S. Staniford-Chen, S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, C. Wee, R. Yip, and D. Zerkle. GrIDS – A graph-based intrusion detection system for large networks. In *Proceedings of the 19th National Information Systems Security Conference*, 1996.

[Sch99] B. Schneier. Attack trees: Modeling security threats. *Dr. Dobb's Journal*, Dec 1999.

[Sch05] Stuart E. Schechter. Toward Econometric Models of the Security Risk from Remote Attacks. *IEEE Security and Privacy*, 3(1):40–44, 2005.

[SCS03] S. Singh, M. Cukier, and W.H. Sanders. Probabilistic validation of an intrusion-tolerant replication system. In de Bakker, J.W., de Roever, W.-P., and Rozenberg, G., editors, *International Conference on Dependable Systems and Networks (DSN'03)*, June 2003.

[SCS+04] F. Stevens, T. Courtney, S. Singh, A. Agbaria, J. F. Meyer, W. H. Sanders, and P. Pal. Model-Based Validation of an Intrusion-Tolerant Information System. In *Proceedings of the 23rd Symposium on Reliable Distributed Systems (SRDS 2004)*, Oct 18-20 2004.

[SDE+04] Evans S., Heinbuch D., Kyle E., Piorkowski J., and Wallner J. Risk-based systems security engineering: Stopping attacks with intention. *IEEE Security & Privacy*, pages 59–62, 2004.

[SEK02] G. Vigna S.T. Eckmann and R.A. Kemmerer. STATL: An Attack Language for State-based Intrusion Detection. *Journal of Computer Security*, 10(1/2):71–104, 2002.

[Sha53] L. S. Shapley. Stochastic games. *Proceedings of the National Academy of Science USA*, 39:1095–1100, 1953.

[Shi00] R. Shirey. Internet Security Glossary. RFC 2828 (Informational), May 2000.

[SHJ+02] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J.M. Wing. Automated generation and analysis of attack graphs. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2002.

[SHK05] Karin Sallhammar, Bjarne E. Helvik, and Svein J. Knapskog. Incorporating attacker behavior in stochastic models of security. In *Proceedings of the 2005 International Conference on Security and Management (SAM'05)*, June 2005.

[SHK06a] Karin Sallhammar, Bjarne E. Helvik, and Svein J. Knapskog. A game-theoretic approach to stochastic security and dependability evaluation. In *Proceedings of the 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC'06)*, Sep/Oct 2006.

[SHK06b] Karin Sallhammar, Bjarne E. Helvik, and Svein J. Knapskog. On stochastic modeling for integrated security and dependability evaluation. *Journal of Networks*, 1(5), Sep/Oct 2006.

[SHK06c] Karin Sallhammar, Bjarne E. Helvik, and Svein J. Knapskog. Towards a stochastic model for integrated security and dependability evaluation. In *Proceedings of the First International Conference on Availability, Reliability and Security (AReS 2006)*, April 2006.

[SHK07] Karin Sallhammar, Bjarne E. Helvik, and Svein J. Knapskog. A framework for predicting security and dependability measures in real-time. *International Journal of Computer Science and Network Security (IJCSNS)*, 7(3):169–183, March 2007.

[SJ05] Evans S. and Wallner J. Risk-based systems security engineering through the eyes of the adversary. In *Proceedings of the 2005 IEEE Workshop on Information Assurance and Security*, 2005.

[SK04] Karin Sallhammar and Svein J. Knapskog. Using game theory in stochastic models for quantifying security. In *Proceedings of the 9th Nordic Workshop on Secure IT-Systems (Nordsec2004)*, December 2004.

[SKH05] K. Sallhammar, S.J. Knapskog, and B.E. Helvik. Using stochastic game theory to compute the expected behavior of attackers. In *Proceedings of the 2005 International Symposium on Applications and the Internet Workshops (Saint2005)*, Jan/Feb 2005.

[Sno02] SNORT: The Open Source Network Intrusion Detection System, 2002. http://www.snort.org.

[Som04] R. Somla. New algorithms for solving simple stochastic games. In *Proceedings of the Games in Design and Verification Workshop*, 2004.

[SOQW95] W. H. Sanders, W. D. Obal, M. A. Qureshi, and F. K. Widjanarko. The ultraSAN modeling environment. *Performance Evaluation*, 24(1-2):89–115, 1995.

[Sta91] S. Stahl. *A Gentle Introduction to Game Theory*. American Mathematical Society, 2 edition, 1991.

[Sta03] William Stallings. *Network Security Essentials, Applications and Standards*. Prentice Hall, 2nd edition, 2003.

[Sta04] Standards Australia and Standards New Zealand. AS/NZS 4360: 2004 risk management, 2004.

[The06] The Honeynet Project, Sep 2006. http://www.honeynet.org/.

[WFP99] C. Warrender, S. Forrest, and B. Pearlmutter. Detecting intrusions using system calls: Alternative data models. In *Proceedings of the 1999 IEEE Symposium on Security and Privacy*, 1999.

[WMT03] D. Wang, B.B. Madan, and K.S. Trivedi. Security Analysis of SITAR Intrusion Tolerance System. *ACM SSRS'03*, 2003.