Towards a Stochastic Model for Integrated Security and Dependability Evaluation

Karin Sallhammar, Bjarne E. Helvik and Svein J. Knapskog
Centre for Quantifiable Quality of Service [*]
Norwegian University of Science and Technology
O.S. Bragstads plass 2E, N-7491 Trondheim, Norway
{sallhamm, bjarne, knapskog}@q2s.ntnu.no

Abstract

We present a new approach to integrated security and dependability evaluation, which is based on stochastic modelling 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 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 [6]. 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. 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 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.

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.

Modelling 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 modelled as transition between states in the model. In Section 3, 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 4, we demonstrate how the expected attacker behavior

[*] "Centre for Quantifiable Quality of Service in Communication Systems, Centre of Excellence" appointed by the Research Council of Norway, funded by the Research Council, NTNU and UNINETT. http://www.q2s.ntnu.no/

is affected by the parameters of the game. To illustrate the approach, Section 5 includes a small case study. In Section 6 we compare our work with previous related work. Section 7 includes some concluding remarks and points to future work.

2 The Stochastic Model

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 [7]. 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, one needs to use a fine-granular system description, which is capable of incorporating the dynamic behavior of a system. This is the main strength of state transition models where, at a low level, the system is modelled as a finite state machine (most systems consist of a set of interacting components and the system state is therefore the set of its component states). During its operational lifetime, a system will alternate between the 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. In a state transition model, one usually discriminates between good states and failed states. Normally, a system will be subject to multiple failure cases, so that the model will have multiple failure modes.

2.1 The Failure Process

It has been shown in [2, 9, 16] that the "fault-error-failure" pathology, which is commonly used for modelling 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 modelled as randomly distributed in time, as is common practice in dependability modelling 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 these faults is generally referred to as an *intrusion*. Because they are intentional in nature, intrusions cannot be modelled 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 [13], 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 Modelling 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, we need to take a closer look at the intrusion process itself. According to [16], 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. In this paper we model the expected time to exploit a vulnerability when using action a as negatively exponentially distributed (this is primarily to simplify analytical assessment of the model. In reality, other types of distributions may be more suitable.) with rate $\lambda_{ij}(a)$, where i and j are two different states in the stochastic model. 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. Hence, the *failure rate* between state i and j may be computed as $q_{ij} = \pi_i(a) \cdot \lambda_{ij}(a)$. This is illustrated in Figure 1 where the good state $i = 1$ is depicted as a circle and the failed state $j = 2$ as a square.

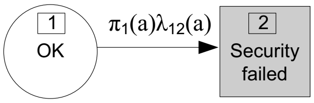

Figure 1. A two-state Markov model with assigned failure rate.

By introducing the decision probability $\pi_i(a)$, the result from a successful attack, i.e. a malicious external human-made fault, can be modelled as one or more *intentional state changes* of the underlying stochastic process, which represents the dynamic behavior of the system.

2.3 Obtaining Steady State Probabilities

In mathematical terms, the stochastic process describing the dynamic system behavior is a continuous time Markov chain with discrete state space. Let

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

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

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

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 a decision probability and a success rate, as explained in Section 2.2. If the initial state of the system, i.e. $\mathbf{X}(0)$, is known, the steady state probabilities $X_i = \lim\limits_{t \to \infty} X_i(t), i = 1, \dots, N$ can be obtained by solving the set of N equations given by $N - 1$ of the N equations

$$\mathbf{XQ} = \mathbf{0}, \tag{3}$$

and with the Nth equation

$$\sum_{l=1}^{N} X_l = 1. \tag{4}$$

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). Similarly, by making the failure states absorbing, i.e. removing all outgoing transitions, one can compute the mean time to first failure ($MTFF$) for a system. See e.g. [4] for details.

2.4 Model Parameterization

In order to obtain measures, the stochastic model has to be parameterized. 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 $\lambda_{ij}(a)^{-1}$'s, i.e. the expected times to succeed with attacks given that they are pursued, remains a challenge. The most straightforward solution is to let security experts assess the rates 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 [8] indicates that the time between successful intrusions during the standard attack phase is exponentially distributed. Even though the process of assessing the exploit rates 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 [17, 18], we propose *game theory* as a means for computing the expected attacker strategy. The procedure is summarized in the next section.

3 Computing Expected Attacker Behavior

In this section we demonstrate how a game theoretic model can be used to compute the expected attacker behavior, in terms of a set of strategies $\pi = \{\pi_i\}$. The procedure is as follows:

Step 1: Identify the game elements. From the stochastic model, pick all states where the system is vulnerable to malicious faults. Each of these states can be viewed as a game element Γ_i in the two-player, zero-sum, stochastic game Γ. For example, in Figure 2 the shaded states 2, 3 and 4 represent states where the system is vulnerable to attacks and which have the game elements Γ_2, Γ_3 and Γ_4, respectively.

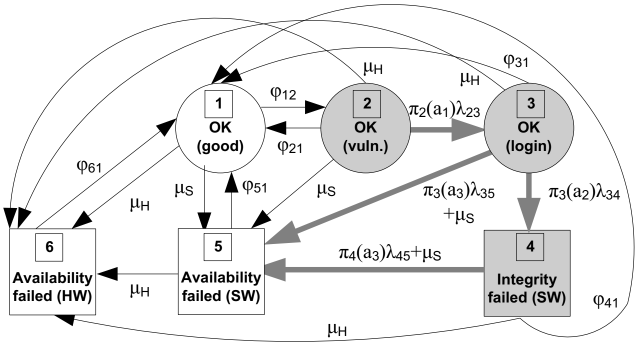

Figure 2. State transition model of DNS server (cf. Section 5) with game elements identified.

Step 2: Construct the action set. The next step is to construct the action set A, which 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 there will always be an inaction ϕ, which represents that an attacker takes no action at all. The action set is the complete set of all these actions, ϕ included. All actions will not necessarily be available in all states; we use A_i to refer to the set of actions available in state i. In Fig. 2 the complete action set is $A = \{a_1, a_2, a_3, \phi\}$, however $A_2 = \{a_1, \phi\}$, $A_3 = \{a_2, a_3, \phi\}$ and $A_4 = \{a_3, \phi\}$.

Step 3: Assign reward and cost values. To model the attackers' motivation we make use of a reward- and cost concept. For each game element, we assign two values to each attack action; one that represents the reward gained by the attacker if the action remains undetected, and another to represent the negative reward, i.e. cost, experienced if the action is detected and reacted to. These values are denoted $r_i(a|\text{undetected})$ and $r_i(a|\text{detected})$, respectively. 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, e.g. financial gain and loss. For instance, in [12] the reward of a successful attack action is the expected amount of recovery effort required from the system administrator and in [11] the reward is the degree of bandwidth occupied by a DDoS attack. In contrast to [11, 12], we use the cost as an alternative outcome of the game to represent the fact that risk averse attackers may sometimes refrain from certain attack actions due to the possible consequences of detection. This topic will be further discussed in Section 4.

Step 4: Compute transition probabilities between the game states. Given that action a is chosen, there is a probability that the intrusion will succeed and the game will continue. The transition probability between game elements can therefore be computed by conditioning on the chosen action. For the example in Figure 2: if the system is in state 2 and an attacker decides to perform action a_1, then $\pi_2(a_1) = 1$. Hence, the transition probability between game elements 2 and 3 for this particular "play of the game" is computed as

$$p_{23}(a_1) = \frac{\lambda_{23}}{\lambda_{23} + \varphi_{21} + \mu_S + \mu_H} \tag{5}$$

Step 5: Solve the game model. The last step is to solve the game model. Recall that A_i is the set of actions available in state i. Each game element Γ_i is therefore represented by a $|A_i| \times 2$ matrix, which has the form

$$\Gamma_i = \begin{pmatrix} \vdots & \vdots \\ \gamma_1(a) & \gamma_2(a)) \\ \vdots & \vdots \end{pmatrix}$$
$$= \begin{pmatrix} \vdots & \vdots \\ r_i(a|\text{undetected}) + \sum_{\Gamma_j \in \Gamma} p_{ij}(a)\Gamma_j & r_i(a|\text{detected}) \\ \vdots & \vdots \end{pmatrix} \tag{6}$$

Solving the model means 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 her own reward. For each system state i, which is modelled as a game element Γ_i, we can therefore expect an attacker to behave in accordance with the probability distribution $\pi_i = \{\pi_i(a)\}$ that maximizes $E(\pi_i, \theta_i)$, where

$$E(\pi_i, \theta_i) = \sum_{\forall a \in A_i} \pi_i(a) \Big(\big(1 - \theta_i(a)\big)\gamma_1(a) + \theta_i(a)\gamma_2(a) \Big). \tag{7}$$

$\theta_i(a)$ is the probability that action a will be detected in state i. The probability distribution π_i that maximizes (7) is called the *optimal strategy of* Γ_i. An attacker who does not know θ_i should think of the system as a counterplayer in the game who tries to minimize the attacker's reward. Hence, the optimal strategy of Γ_i is obtained by solving

$$\max_{\pi_i} \min_{\theta_i} E(\pi_i, \theta_i), \tag{8}$$

which is denoted the *Nash Equilibrium* of Γ_i. To find the optimal strategies for all game elements in the stochastic game, one can use a set of inductive formulas. For further details on the underlying assumptions and solution of the game model, the reader is referred to [17, 18], or [15, pp. 96–101].

4 Tuning Parameters of the Game

The game model presented in the previous section is based on a reward- and cost concept. Whenever an attacker performs an attack action, he immediately receives a reward. Furthermore, if the action succeeds, additional rewards may be gained. The reward values will therefore represent the attackers' motivation when deciding on attack actions. 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 demotivation factor when modelling, 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. In this section we demonstrate how the cost parameters in our game model will affect the expected attacker behavior.

4.1 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 [1] Rogers summarizes earlier research on hacker categorization and provides a new taxonomy based on a two dimensional classification model. *Skill* and *motivation* are identified as the primary classification criteria, which fit well into our mathematical framework where an attacker's skill is represented by the expected time to success, $\lambda^{-1}(a)$, and the motivation by the reward- and cost concept. Rogers' model suggests eight primary categories, whereof seven represent outsiders: ”novices”, ”cyber-punks”, ”petty thieves”, ”virus writers”, ”old guard hackers”, ”professional criminals” and ”information warriors”. The eighth category is ”internals”. Our model does not depend on any attacker classification. Instead, in our approach it is possible to tune the reward- and cost values of the game elements and thereby be able to model the motivation of any kind of attacker.

4.2 Varying the Cost Parameters

To illustrate the effect of the cost parameters, we use a generic 2×2 game element

$$\Gamma_i = \begin{pmatrix} \gamma_1(a) & \gamma_2(a) \\ \gamma_1(\phi) & \gamma_2(\phi) \end{pmatrix} = \begin{pmatrix} 1 & b \\ c & 0 \end{pmatrix} \tag{9}$$

The generic game element in (9) represents a system state i where the system is vulnerable to one single attack action a. Hence, an attacker can choose either to perform the attack (a), or to resign (ϕ). By varying b and c we can now demonstrate how the relation $\gamma_2(a)/\gamma_1(a)$ (i.e. the cost of a detected attack versus the reward of an undetected attack) and $\gamma_1(\phi)/\gamma_1(a)$ (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 use (8), i.e. the Nash Equilibrium of Γ_i.

Example 1: reducing b. If $b = -2$ and $c = -3$ in (9), 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.

Example 2: reducing c. Again, if $b = -2$ and $c = -3$ in (9), 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.

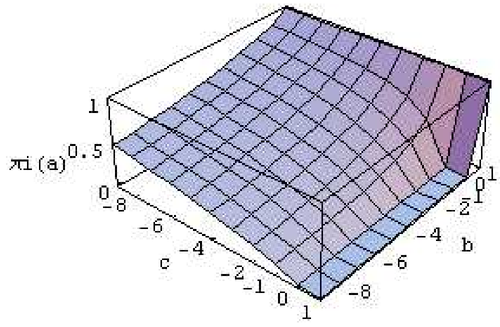

Figure 3. The expected attacker behavior $\pi_i(a)$ w.r.t. b and c.

Figure 3 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 payoff whether she 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. (Remark: this is an ideal situation which is 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. (9) is c. Unfortunately, since c represents a mental factor in this game (an attacker's reluctance to resign) it will be difficult for system administrators to take preventive measures influencing c in a way that will reduce $\pi_i(a)$.

5 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 integrity and availability. We distinguish between two different types of accidental failures: hardware availability failures which require a manual repair, and software availability failures, 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 during its lifetime the system will alternate between a good state (1) where it is secure against these types of attacks and another good, but vulnerable, state (2) 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 (3), and thereby make it possible to insert false entries in the server cache (software integrity failure) or to shut the server down (software availability failure). In this case, all the three states 1-3 are considered to be good states. Even though the system is erroneous in states 2 and 3, it still manages to deliver the intended service: i.e. to provide clients with correct replies to DNS requests.

The state transition model in Figure 2 in Section 3 represents the security and dependability of the service 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 an imaginary system administrator's possible actions and the λ rates represent the success rates of the possible attack actions. The action set in the stochastic game is $A = \{a_1, a_2, a_3, \phi\} = \{\textit{illegal login}, \textit{cache poisoning}, \textit{server shut down}, \textit{do nothing}\}$. Using the rate values $\lambda_{23} = 1/3$, $\lambda_{34} = \lambda_{35} = \lambda_{45} = 3$, $\varphi_{12} = 1/480$, $\varphi_{21} = 1/120$, $\varphi_{31} = \varphi_{41} = 1$, $\varphi_{51} = 3$, $\varphi_{61} = 1/24$, $\mu_H = 1/3600$ and $\mu_S = 1/120$ per hour, together with a fictitious set of reward- and cost values, the game elements become

$$\Gamma_2 = \begin{pmatrix} 1 + 0.952\Gamma_3 & -4 \\ -5 & 0 \end{pmatrix}, \Gamma_3 = \begin{pmatrix} 1 + 0.748\Gamma_4 & -3 \\ 1 & -2 \\ -5 & 0 \end{pmatrix},$$

$$\Gamma_4 = \begin{pmatrix} 1 & -2 \\ -5 & 0 \end{pmatrix}.$$

Solving the stochastic game in accordance to (8) provides the strategy vectors $\pi_2 = (0.568, 0.432)$, $\pi_3 = (0, 0.625, 0.375)$ and $\pi_4 = (0.625, 0.375)$, hence, the state transition rate matrix for the DNS server is as displayed in Table 1.

Using (3) and (4) in Section 2.3, we compute the stationary probabilities $\mathbf{X} = \{X_1, \ldots, X_6\} = \{0.98, 0.01, 6.50 \cdot 10^{-4}, 0, 3.16 \cdot 10^{-3}, 6.62 \cdot 10^{-3}\}$. By using the traditional Markov analysis described in [4] we compute the stationary probability of being in any of the failed states ($i = 4, 5, 6$) to $9.78 \cdot 10^{-3}$, the mean time to first failure $MTFF = 101.6$ hours, the mean time to failure $MTTF = 101.0$ hours, the mean time between failures $MTBF = 102.6$ hours and the mean time spent in the good states ($i = 1, 2, 3$) as $MUT = 101.6$ hours. It should be noted that all values in this example, and the corresponding operational measures, are chosen for illustration purposes only.

6 Related Work

Security-dependability A thorough definition of basic dependability- and security concepts is provided by Laprie et.al. [2]. A deliverable produced by the MAFTIA project [16] 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. [9] 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. [13] 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. [14] 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

$$\mathbf{Q} = \begin{pmatrix} -1.07 \cdot 10^{-2} & 2.08 \cdot 10^{-3} & 0 & 0 & 8.33 \cdot 10^{-3} & 2.78 \cdot 10^{-4} \\ 8.33 \cdot 10^{-3} & -0.20 & 0.18 & 0 & 8.33 \cdot 10^{-3} & 2.78 \cdot 10^{-4} \\ 1 & 0 & -2.88 & 0 & 1.88 & 2.78 \cdot 10^{-4} \\ 1 & 0 & 0 & -2.88 & 1.88 & 2.78 \cdot 10^{-4} \\ 3 & 0 & 0 & 0 & -3.89 & 2.78 \cdot 10^{-4} \\ 4.17 \cdot 10^{-2} & 0 & 0 & 0 & 0 & -4.17 \cdot 10^{-2} \end{pmatrix}$$

Table 1. The state transition rate matrix for the DNS server (rate values in matrix reduced to three significant numbers).

of operational security expressed as the mean effort to security failure as proposed by [10]. Further, Madan et. al. [3] use traditional stochastic modelling 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 [19] 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 modelling approach is similar to [3], but differs in that we use decision probabilities to integrate attacker behavior in the transition rates of the 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 [12] use a game theoretic method to analyze the security of computer networks. The interactions between an attacker and the administrator are modelled as a two-player stochastic game for which best-response strategies (Nash Equilibrium) are computed. In [11] a preliminary framework for modelling 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 [12], 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 [11] we use the same game model for different threat environments.

7 Concluding Remarks

This paper presents a stochastic model for integrated security and dependability assessment. Our model considers many 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 Nash equilibrium (8) has frequently been used to derive predictions of what players in a game will do [5]. As pointed out in Section 3, the Nash equilibrium of the game 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, the solution to the maximization problem (7) will be straightforward to compute, hence, (8) 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 modelling the attackers' interactions with the system as a zero-sum stochastic game will always provide us with a single unique solution to the game.

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?" or "Should we perform additional monitoring?". The effect of these two countermeasures can be evaluated in our modelling framework before implementation, by changing the corresponding transition rates in the model and then comparing the results.

In the future we plan to investigate whether time-dependent success rates can be used to compute more realistic strategies (we must assume that attackers learn over time!). This will exclude analytical assessment of the model. However simulation is still an option. Furthermore, verifying the model's ability to predict real-life attacks will require further research, including validation of the model against empirical data.

References

[1] The development of a meaningful hacker taxonomy: A two dimensional approach. Technical report, CERIAS Tech Report 2005-43, 2005.

[2] 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.

[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] Charles A. Holt and A. E. Roth. The Nash equilibrium: A perspective. In *Proceedings of the National Academy of Sciences*, volume 101, March 23 2004.

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

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

[8] 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.

[9] E. Jonsson, L. Strömberg, 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.

[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] 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.

[12] Kong-wei Lye and Jeannette M. Wing. Game strategies in network security. *International Journal of Information Security*, 4(1-2):71–86, 2005.

[13] 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.

[14] 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.

[15] G. Owen. *Game Theory*. Academic Press, 3rd edition, 2001.

[16] David Powell and Robert Stroud (eds.). Malicious- and accidental-fault tolerance for internet applications - Conceptual model and architecture, 2001.

[17] 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 (Saint2005) Workshops*, 2005.

[18] 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)*, 2005.

[19] 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.