Modeling and Evaluating the Survivability of an Intrusion Tolerant Database System

Hai Wang and Peng Liu

College of Information Sciences and Technology
Pennsylvania State University
University Park, PA 16802, USA
{haiwang, pliu}@ist.psu.edu

Abstract. The immaturity of current intrusion detection techniques limits the traditional security systems in surviving malicious attacks. Intrusion tolerance approaches have emerged to overcome these limitations. Before intrusion tolerance is accepted as an approach to security, there must be quantitative methods to measure its survivability. However, there are very few attempts to do quantitative, model-based evaluation of the survivability of intrusion tolerant systems, especially in database field. In this paper, we focus on modeling the behaviors of an intrusion tolerant database system in the presence of attacks. Quantitative measures are proposed to characterize the capability of a resilient database system surviving intrusions. An Intrusion Tolerant DataBase system (ITDB) is studied as an example. Our experimental results validate the models we proposed. Survivability evaluation is also conducted to study the impact of attack intensity and various system deficiencies on the survivability.

1 Introduction

Although intrusion tolerance techniques, which gain impressive attention recently, are claimed to be able to enhance the system survivability, survivability evaluation models are largely overlooked in the previous research. Quantifying survivability metrics of computer systems is needed and important to meet the user requirements and compare different intrusion tolerant architectures. Efforts aimed at survivability evaluation have been based on classic reliability or availability models.

The work described in this paper is motivated by the limitations of using the evaluation criteria for availability to evaluate survivability. The evaluation criteria for system availability are quantified by availability modeling, which has a fairly matured literature as summarized in [1]. However, the availability model cannot be used to quantify the survivability of a security system. Besides the differences between security and fault tolerance, a fundamental reason is because the availability model assumes the "fail-stop" semantics, but the "attack-stop" semantics probably can never be assumed in trustworthy data processing systems, not only because of the substantial detection latency, but also because of the needs for degraded services.

D. Gollmann, J. Meier, and A. Sabelfeld (Eds.): ESORICS 2006, LNCS 4189, pp. 207–224, 2006.

The goal of this paper is taking the first step to develop a survivability evaluation model that can systematically address the inherent limitations of classic availability evaluation models in measuring survivability. The approach we proposed is using a state transition graph to model an intrusion tolerant database system. We attempt to model the system in a modular way, so that it can be easily adapted to a wide variety of intrusion tolerant database systems. Quantitative measures are proposed to characterize the capability of a resilient database system surviving intrusions. Furthermore, we are interested in understanding the impact of existing system deficiencies and attack behaviors on the survivability. In this paper, we take the first step to do detailed, quantitative evaluation of the survivability of intrusion tolerant database systems and the impact of system deficiencies and attack behaviors on it.

In particular, the main contributions of this paper are four-fold:

1. We extend the classic availability model to a new survivability (evaluation) model. Comprehensive state transition approaches are applied to study the complex relationships among states and their transition structures encoding sequential response of intrusion tolerant database systems facing attacks.
2. Novel quantitative survivability evaluation metrics are proposed by us. Mean Time to Attack (MTTA), Mean Time to Detection (MTTD), Mean Time to Marking (MTTM), and Mean Time to Repair (MTTR) are proposed as basic measures of survivability. We find that there is a natural mapping between the MTTA-MTTD-MTTM-MTTR model and the steady state probabilities of the system in state transition modeling. This mapping not only provides valuable insights on why the MTTA-MTTD-MTTM-MTTR model can measure survivability, but also provides a convenient way to use mathematical analysis to quantify survivability. Based on the MTTA-MTTD-MTTM-MTTR model, this survivability measuring methodology is no longer ad hoc.
3. To validate the survivability models we proposed, a representative intrusion tolerant database system, ITDB [2], is studied as an empirical example. A real testbed is established to conduct comprehensive validation experiments running TPC-C benchmark transactions. Experimental results show the validity of the survivability models we proposed.
4. To further evaluate the security of ITDB, we have done an empirical survivability evaluation, where maximum-likelihood methods are applied to estimate the values of the parameters used in our state transition models. The impacts of existing system deficiencies and attack behaviors on the survivability are then studied using quantitative measures we defined.

The rest of the paper is organized as follows. In Section 2, we give an overview of the ITDB framework. In Section 3, a series of state transition models are proposed. In Section 4, quantitative measures of database system survivability are proposed. The experiments are conducted in Section 5 to validate the models we established. Survivability evaluation results are reported in Section 6. In Section 7, we discuss the related work. We conclude our paper in Section 8.

2 ITDB: An Motivating Example

ITDB is motivated by the following practical goal: "after the database is damaged, automatically locate the damaged part, contain and repair it as soon as possible, so that the database can continue being useful in the face of attacks or intrusions". The major components of ITDB are shown in Figure 1. Note that in [3], a comprehensive ITDB system has been proposed. In this paper, we only focus on important components of ITDB, namely the damage containment and recovery subsystems. In the rest of this section, we give a brief overview of the functions of major ITDB components.

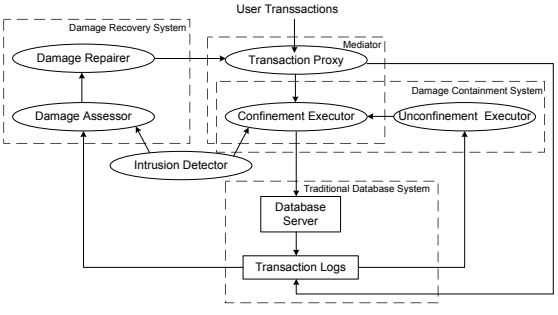


Fig. 1. Basic ITDB System Architecture

The Mediator subsystem functions as a "proxy" for each user transaction and transaction processing call to the database system. Through this proxy, ITDB is able to keep useful information about user transactions, such as information about transactions' read and write operations, which is important to generate the corresponding logs for damage recovery and containment. This part is the foundation of the whole ITDB system. All other subsystems of ITDB rely on this part.

Traditional damage containment approaches are one-phase. An item o will not be contained until it is identified as damaged. However, significant damage assessment latency can cause the damage on o spreading to many other data items before o is contained. To overcome this limitation, ITDB uses a novel technique called multi-phase damage containment as an alternative. This approach has one containing phase, which instantly contains the damage that might have been caused by an intrusion as soon as the intrusion is identified, and one or more later on uncontaining phases, denoted *containment relaxation*, to uncontain the items that are mistakenly contained during the containing phase.

The damage recovery subsystem has the responsibility to perform accurate damage assessment and repair. To do this job, first, the damage recovery subsystem retrieves reported malicious transaction messages from the intrusion detection subsystem. ITDB then traces damage spreading by capturing the dependent-upon relationship among transactions. ITDB repairs the damage caused by T_i using a

special *cleaning* transaction which restores each contaminated data item to its latest undamaged version.

The intrusion detection subsystem has the responsibility to detect and report malicious transactions to the damage containment and recovery subsystems. It uses the trails kept in the logs and some relevant rules to identify malicious transactions.

3 Modeling Intrusion Tolerant Database Systems

To analyze and evaluate the survivability of an intrusion tolerant database system, a quantitative evaluation model is required. A variety of modeling techniques can be applied in the research of survivability study. Deterministic models are quite limited in the stochastic behavior. State transition models are much more comprehensive. All possible system states can be captured by state transition models. In this section, we apply state transition models to explore the complex relationships and transition structure of an intrusion tolerant database system.

3.1 Basic State Transition Model

Figure 2 shows the basic state transition model of an intrusion tolerant database system. Traditional computer security leads to the design of systems that rely on prevention to attacks. If the strategies for prevention fail, the system is brought from good state G into the infected state I during the penetration and exploration phases of an attack. If the attack is detected successfully, intrusion tolerance system picks up where attack prevention leaves off. The system enters the containment state M. In this state, all suspicious data items are contained. After marking all the damage made by the attack, undamaged items are released and the system enters to the recovery state R. The repair process will compensate all the damage and the system returns to the good state G. The four phases which are attack penetration, error detection, attack containment, damage assessment and error recovery, describe the basic phenomena that each intrusion tolerant system will encounter. These can and should be the basic requirement for the design and implementation of an intrusion tolerant database system.

Parameters in Figure 2 are: $1/\lambda_a$ is the mean time to attacks (MTTA), the expected time for the system to be corrupted; $1/\lambda_d$ is the mean time to detect (MTTD), the expected time for the intrusion to be detected; $1/\lambda_m$ is the mean time to mark (MTTM), the expect time for the system to mark "dirty" data

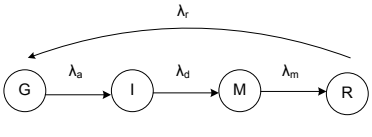


Fig. 2. Basic State Transition Model

items; $1/\lambda_r$ is the mean time to repair (MTTR), the expect time for the system to repair damaged data items.

3.2 Intrusion Detection System Model

As an important part of an intrusion tolerant system, the Intrusion Detection System (IDS) is largely ignored in intrusion tolerant system modeling. [4] assumes that the IDS can report intrusion without delay or false alarm. Some works only consider part of IDS parameters, like true positive [5]. In this part, we will integrate a comprehensive model of IDS into the whole system.

False alarm rate and detection probability are widely used to evaluate the performance of an IDS in either networking [6] or database field [7]. Detection latency, so called *detection time* in [8], is another metrics to evaluate an IDS. We define detection latency as the duration that elapses from the time when an attack compromises a database system successfully to the time when the IDS identifies the intrusion. All these three metrics are included in our model.

Let T_a and T_{fa}, respectively, denote the times to intrusion and the time to the failure of the IDS. If the IDS fails before the intrusion, then a false alarm is said to have occurred. Let A denote the time to intrusion occurrence. Clearly,

$$A = min\{T_A, T_{fa}\} \tag{1}$$

We assume that T_a and T_{fa} are mutually independent and exponentially distributed with parameter λ_a and α, respectively. Then, clearly, A is exponentially distributed with parameter $\lambda_a + \alpha$.

After the intrusion, it takes a finite time T_d (*detection latency*) to detect the intrusion. We assume that the time to identify one successful intrusion is exponentially distribution with parameter λ_d. For the imperfect detection, we assume that all attacks will be identified by the database administrator eventually. We use state MD and MR to represent the *undetected state* and *manual repair state* respectively. We assume that the detection probability of an IDS to identify a successful intrusion is d. The transition probability that the system transfers from state I to state MD is $(1 - d)$. We assume that the time to manually identify a successful intrusion is exponentially distribution with parameter λ_{md} and the time to manually repair infected data items is exponentially distribution with parameter λ_{mr}. The state transition model considering the deficiencies of the IDS is presented in Figure 3.

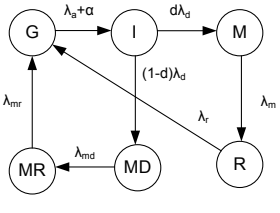


Fig. 3. State Transition Model with IDS

3.3 Damage Propagation and Repair Model

The damage keeps propagating the effect of the intrusion during the detection phase. The purpose of the IDS is to reduce its detection latency. Although damage spreading is a normal phenomenon, little work puts effort on studying the effect of damage propagation on the survivability in their intrusion tolerant system models. In this part, we want to take the first step to study the effect of detection delay on damage propagation, which may affect damage assessment and repair correspondingly.

Let T_{di} denotes the time between the infection of $(i-1)$th and ith data item. Obviously,

$$T_d = \sum_{i=1}^{k} T_{di} \tag{2}$$

where k is the number of infected data items during the detection latency. Let's assume that T_{di} is exponentially distributed with parameter λ_{di} and

$$F_{Di}(t) = 1 - e^{-\lambda_{di} t} \tag{3}$$

As soon as the intrusion is identified, the containing phase instantly contains the damage that might have been caused by an intrusion. At the same time, the damage assessment process begins to scan the contained data items and locate the infected ones. We assume that the time to scan one infected data item is exponentially distributed with parameter λ_m.

After all infected data items are identified via the damage assessment process, the repair system begins to compensate the damage caused by the intrusion. We assume the time to repair one infected data item is exponentially distributed with parameter λ_r

Let $(I:k)$ denote the *infect state* with k infected data items in the database, and $(M:k)$ denote the *mark state* with k infected data need to be located. Figure 4 shows the comprehensive state transition model of ITDB.

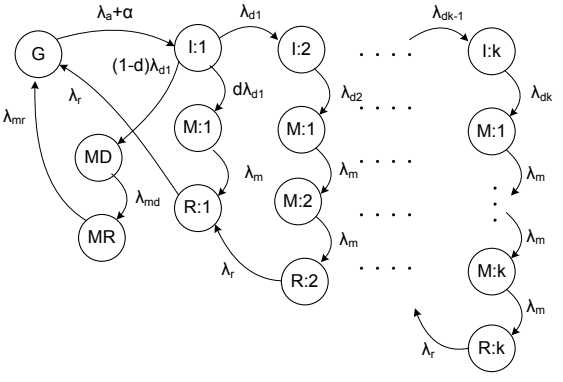


Fig. 4. Comprehensive State Transition Model

4 Survivability Evaluation

Evaluation criteria in trustworthy data processing systems are often referred to as *survivability* or *trustworthiness*. Survivability refers to the capability of a system to complete its mission, in a timely manner, even if significant portions are compromised by attacks or accidents [9]. However, in the context of different types of systems and applications, it can mean many things. This brings a difficulty in the measurement and interpretation of survivability. For a database system, survivability is the quantified ability of a system or subsystem to maintain the integrity and availability of essential data, information, and services. Also a survivable database system should maintain the performance of essential services facing attacks. In this section, based on the models we established in Section 3, quantitative metrics are proposed to facilitate evaluating the survivability of intrusion tolerant database systems from several aspects.

4.1 State Transition Model Analysis

Let $\{X(t), t \geq 0\}$ be a homogeneous finite state Continuous Time Markov Chain (CTMC) with state space S and generator matrix $\mathbf{Q} = [q_{ij}]$. Let $P_i(t) = P\{X(t) = i, i \in S\}$ denote the unconditional probability that the CTMC will be in state i at time t, and the row vector $\mathbf{P}(t) = [P_1, P_2, \cdots, P_n]$ represent the transient state probability vector of the CTMC. The transient behavior of the CTMC can be described by the Kolmogorov differential equation:

$$\frac{d\mathbf{P}(t)}{dt} = \mathbf{P}(t)\mathbf{Q} \tag{4}$$

where $\mathbf{P}(0)$ represents the initial probability vector (at time $t = 0$).

In addition, cumulative probabilities are sometimes of interest. Let $L(t) = \int_0^t P(u)du$; then, $L_i = (t)$ represents the expected total time the CTMC spends in state i during the interval $[0, t)$. $L(t)$ satisfies the differential equation:

$$\frac{d\mathbf{L}(t)}{dt} = \mathbf{L}(t)\mathbf{Q} + \mathbf{P}(0) \tag{5}$$

where $\mathbf{L}(0) = 0$.

The steady-state probability vector $\pi = lim_{t\to\infty}\mathbf{P}(t)$ satisfies:

$$\pi\mathbf{Q} = 0, \sum_{i\in S} \pi_i = 1 \tag{6}$$

By solving the equations 4, 5 and 6, we can get some important survivable metrics of an intrusion tolerant database system.

4.2 Survivability Evaluation Metrics

In our model, survivability is quantified in terms of *integrity* and *availability*. According to survivability, we define integrity in a way different from integrity constraints. In this paper, we define integrity as follow:

Definition 1: Integrity is defined as a fraction of time that all accessible data items in the database are clean.

High integrity means that the intrusion tolerant database system can serve the user with good or clean data at a high probability. Obviously, all data items are clean and accessible in state G. When attacks occur, some data items will be affected. So in state I, part of accessible data items are "dirty". After the intrusion is identified, the ITDB can contain the all damaged data until it finish the repair process. Since the ITDB does selective containment and repair, the database system is still available, and accessible data items are clean during the containment, damage assessment, and repair process.

Consider the model in Figure 2, state space $S = \{G, I, M, R\}$. The generator matrix $\mathbf{Q}$ for the basic state transition model in Section 3.1 is:

$$\mathbf{Q} = \begin{bmatrix} -\lambda_a & \lambda_a & 0 & 0 \\ 0 & -\lambda_d & \lambda_d & 0 \\ 0 & 0 & -\lambda_m & \lambda_m \\ \lambda_r & 0 & 0 & -\lambda_r \end{bmatrix} \tag{7}$$

By solving the equations 5 and 6, we can get:

$$\pi_G = \frac{1/\lambda_a}{1/\lambda_a + 1/\lambda_d + 1/\lambda_m + 1/\lambda_r} = \frac{MTTA}{MTTA + MTTD + MTTM + MTTR}$$
$$\pi_I = \frac{1/\lambda_d}{1/\lambda_a + 1/\lambda_d + 1/\lambda_m + 1/\lambda_r} = \frac{MTTD}{MTTA + MTTD + MTTM + MTTR}$$
$$\pi_M = \frac{1/\lambda_m}{1/\lambda_a + 1/\lambda_d + 1/\lambda_m + 1/\lambda_r} = \frac{MTTM}{MTTA + MTTD + MTTM + MTTR}$$
$$\pi_R = \frac{1/\lambda_r}{1/\lambda_a + 1/\lambda_d + 1/\lambda_m + 1/\lambda_r} = \frac{MTTR}{MTTA + MTTD + MTTM + MTTR}$$

From Definition 1, we can get the integrity for the basic state transition model in Section 3.1:

$$I = \pi_G + \pi_M + \pi_R = \frac{MTTA + MTTM + MTTR}{MTTA + MTTD + MTTM + MTTR} \tag{8}$$

Similarly, we can get the integrity for the comprehensive state transition model we proposed in Section 3.3:

$$I = \pi_G + \sum_{i=1}^{k} \pi_{M_i} + \sum_{i=1}^{k} \pi_{R_i} \tag{9}$$

Availability [1] is defined as a fraction of time that the system is providing service to its users. Since the ITDB does on-the-fly repair and will not stop its service facing attacks, its availability is nearly 100%, which can not show the performance of ITDB clearly. To better evaluate the survivability of ITDB, we define another type of availability, Rewarding-availability:

Definition 2: Rewarding-availability (RA) is defined as a fraction of time that the all clean data items are accessible.

If the clean data can not be accessed, it is a loss of service to users. Rewarding-availability means that the system not only can serve its users, but also do not deny the request for the clean data. ITDB will release the all contained clean data items after damage assessment. For the basic state transition model in Section 3.1, the Rewarding-availability is:

$$RA = \pi_G + \pi_R = \frac{MTTA + MTTR}{MTTA + MTTD + MTTM + MTTR} \tag{10}$$

The Rewarding-availability for the comprehensive state transition model in Section 3.3 is:

$$RA = \pi_G + \sum_{i=1}^{k} \pi_{R_i} \tag{11}$$

5 Empirical Validation

The models we proposed in the above section need to be validated. In this section, we compare the prediction of our model with a set of measured ITDB behaviors facing attacks. For our test bed, we use Oracle $9i$ Server to be the underlying DBMS. The TPC-C benchmark [10] is in general DBMS independent, thus the transaction application can be easily adapted to tolerate the intrusions on a database managed by almost every "off-the-shelf" relational DBMS such as Microsoft SQL Server, Informix, and Sybase.

5.1 Parameters Setting and Estimation

In the models we proposed, some parameters can be controlled by us. In our experiments, the behaviors of attackers, human interaction and the properties of IDS can be controlled by us. So we will set the value of attack hitting rate λ_a, false alarm rate α, detection probability d, detection rate λ_d, manual repair rate λ_{mr} and manual detection rate λ_{md}. We will also vary their value to investigate the impact of them on system survivability.

Assume we generate n attack events and k data items are damaged by the attacks. Let assume the total attack time is A_n, the total detect time is D_k, the total manual detection time is MD_n, and the total manual repair time is MR_n. The transition rates are:

$$\lambda_a = \frac{n}{A_n}, \lambda_d = \frac{k}{D_k}, \lambda_{md} = \frac{(1-d)k}{MD_n}, \lambda_{mr} = \frac{(1-d)k}{MR_n} \tag{12}$$

Some parameters in our model are the characters of ITDB, which are not controlled by us. In this section, we will use the method of maximum-likelihood to produce estimators of these parameters. Assume we observed k scan events

Table 1. Parameter Setting and Estimation

Parameters	Value
Attack Hitting Rate, λ_a	0.5(Low); 1(Moderate); 5(Heavy)
Detect Rate, λ_d	10(Slow); 15(Medium); 20(Fast)
Mark Rate, λ_m	27
Repair Rate, λ_r	22
Manual Detection Rate, λ_{md}	0.02
Manual Repair Rate, λ_{mr}	0.02
False Alarm Rate, α	10%; 20%; 50%
Detection Probability, d	80%; 90%; 99%

and repair events, the total mark time is M_k, and the total repair time is R_k. The maximum-likelihood estimators of λ_m, λ_r are

$$\widetilde{\Lambda}_M = \frac{k}{M_k}, \widetilde{\Lambda}_R = \frac{k}{R_k} \tag{13}$$

Table 1 shows the values of parameter setting and estimation of our experiments.

5.2 Validation

The steady state probability of occupying a particular state computed from the model was compared to the estimated probability from the observed data. The steady state probabilities for the Markov model are computed by using Equation 6. The measured data are estimated as the ratio of the length of time the system was in that state to the total length of the period of observation. The results are shown in Table 2.

Table 2. Comparison of state occupancy probabilities. ($\lambda_a = 0.5$, $\lambda_d = 10$, $\alpha = 10\%$, $d = 90\%$).

State	Observed Value	Value from Model	Difference (%)
G	71.64	72.15	0.7169
I	3.96	3.72	6.4516
M	2.64	2.45	7.7551
R	1.98	1.89	4.7619
U	0.55	0.57	3.5088
M	4.4	4.09	7.5795

It can be seen that the computed values from the model and the actual observed values match quite closely. This validates the model building methodology, and so the Markov model can be taken to model the real system reasonably well.

6 Results

In this section, we use ITDB [2] as an example to study intrusion tolerant database systems' survivability metrics we proposed in section 4. Instead of evaluating the performance of a specified system, we focus on the impact of different system deficiencies on the survivability in the face of attack. Experiments run using different system settings and workloads. The analysis presented here is designed to compare the impact of different parameters of intrusion detection subsystems, such as False Alarm Rate, Detection Latency; and different workload parameters, such as Attack Rates on the relative survivability metrics of ITDB.

6.1 Impact of Attack Intensity

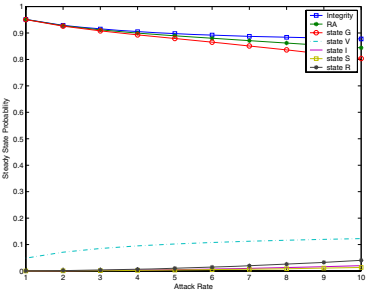
The attack rate can challenge the survivability of an intrusion tolerant system. As an intrusion tolerant system, a key problem is whether ITDB can handle different attack intensity. To answer this question, in this part, we will study the impact of attack rate on survivability of ITDB.

We compare the steady state probabilities of different system configuration of ITDB under different attack rates. In Figure 5(a), an example of a good system, which has a good IDS and fast damage assessment and repair system, is shown. As can be seen, the heavy attacks have little impact on the survivability of ITDB. The damage assessment and repair subsystems can locate and mask the intrusion quickly. As a result, the steady state probabilities of state I, R, and M are very slow. The integrity and rewarding-availability remain at a high level (> 0.8). The only impact of high attack rate is that the probability of ITDB staying at state I is increased. This does not hurt the survivability of ITDB.

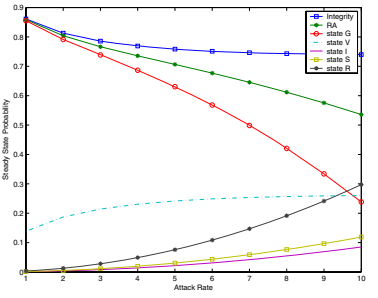
An example of a bad system is shown in Figure 5(b). The high attack rate increases the work load for damage marking and repairing subsystems. As a result, steady state probabilities of state R ($\pi_R > 0.3$) and state M go up quickly. This keeps ITDB busy on analyzing and masking the heavy attacks. However, the system integrity is not impacted by the attacks significantly. The reason is that the ITDB applies the damage containment strategy. This enables the ITDB having the capability to provide clean information to users even facing heavy attacks.

6.2 Impact of False Alarms

False alarm is a key factor to evaluate the performance of an IDS. ITDB adopts the behavior-based intrusion detection techniques. The high false alarm rate is often cited as the main drawback of behavior-based detection techniques since the entire scope of the behavior of an information system may not be covered during the learning phase. High false alarm rate may bring extra workload to the recovery subsystem and waste some system resources. Will ITDB tolerant the relatively high false alarm rates? To answer this question, we will evaluate the impact of false alarms on the steady state of ITDB in this part.

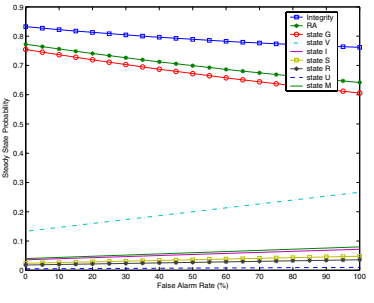


(a) good system ($d = 99\%$, $\alpha = 0.1$, $\lambda_d = 20$)

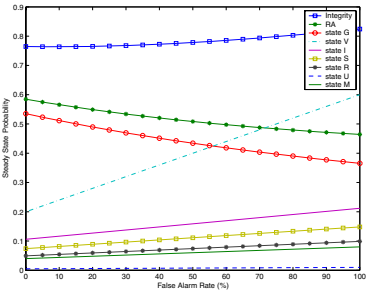


(b) poor system ($d = 80\%$, $\alpha = 0.5$, $\lambda_d = 10$)

Fig. 5. Impact of Attack Intensity



(a) Light Attack ($\lambda_a = 1$, $\lambda_d = 15$, $d = 90\%$)



(b) Heavy Attack ($\lambda_a = 5$, $\lambda_d = 15$, $d = 90\%$)

Fig. 6. Impact of False Alarm Rate

Figure 6(a) shows the variation of steady state probabilities when ITDB is under light attacks ($\lambda_a = 1$). ITDB maintains the integrity (> 0.85) and rewarding-availability (> 0.6) at a high level, even though facing a nearly 100% false alarm rate. This indicates that the system can tolerate a high false alarm rate under light attacks. Also the steady state probabilities of state I, M, R are at a very low level (< 0.1). This indicates that the system can contain, locate, and repair the attacks efficiently and quickly. Another case that ITDB is under heavy attacks ($\lambda_a = 5$) is shown in Figure 6(b). As can be seen, high false alarm brings pressure on ITDB. The steady state probability of state I, π_D, is higher than the probability state G, π_G, when false alarm rate is higher than 60%. The heavy attacks and extra load brought by false alarms increase the steady state probabilities of state I, M, and R. These mean that ITDB spends much more time on state I and keeps busy on analyzing and repairing the damage. The rewarding-availability decreases as the damage containment and assessment process becomes longer.

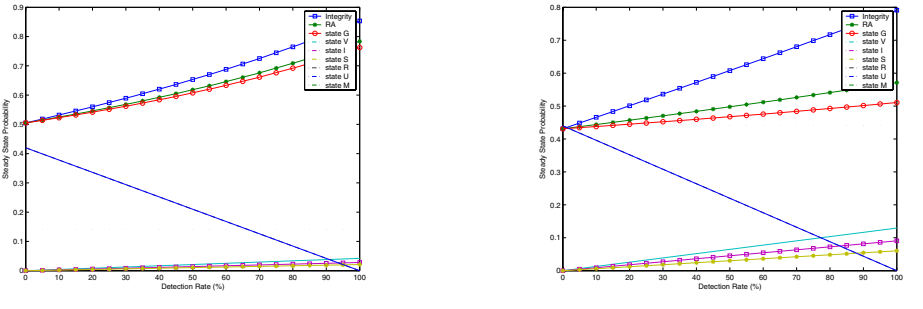
At the same time, the system still can maintain the integrity (> 0.85) at a high level. This means that the probability that the system can provide clean data to some users is high.

6.3 Impact of Detection Probability

Detection probability is another important feature to measure the performance of an intrusion detector. In this section, we will study the impact of detection probabilities on the survivability under different attack intensity.

Figure 7(a) shows that ITDB is under light attack ($\lambda_a = 1$). When detection rate is 0%, the system totally depends on manual detection. Since the manual detection requires human intervention, it takes a relatively long time to detect the intrusion manually. As a result, ITDB has a high probability (> 0.4) staying at state MD and a low probability staying at state G when $d = 0$. The integrity and rewarding-availability are also at a low level (≈ 0.5). The steady state probability of state MD goes back to 0 when the detection probability is 100%. The steady state probabilities of state M and R go up while the detection probability is increasing. This indicates that, with more attacks are identified by the IDS, the system will spend more time on damage assessment and recovery. Since the manual repair is much slower than the repair subsystem of ITDB, the rewarding-availability and integrity go up while the detection probability is increasing. When ITDB faces a heavy attack as shown in Figure 7(b), low detection rate hurts the performance of ITDB. The steady state probability of state G, π_G is lower than 0.5.

Compared with the false alarms, the impact of detection probability on the survivability of an intrusion tolerant database system is severer. The variance of integrity and rewarding-availability is less than 0.2 when the detection probability changes from 0% to 100%, while the variance is nearly 0.4 when changing false alarm rate from 0% to 100%. One reason is that the high false alarms will bring extra load to the security system to contain and repair unaffected data items,



(a) Light Attack ($\lambda_a = 1$, $\alpha = 0.2$, $\lambda_d = 15$)

(b) Heavy Attack ($\lambda_a = 5$, $\alpha = 0.2$, $\lambda_d = 15$)

Fig. 7. Impact of Detection Rate

while low detection probability will bring more work for the administrator to mark and repair the damage manually. If the system can identify and recover the damage faster than manual operation, the impact of low detection probability is severer and more dangerous to the survivability. This result encourages us to consider more on improving the detection probability for the future intrusion tolerant system development.

6.4 Transient Behaviors

Much of the theory developed for solving Markov chain models is devoted to obtaining steady state measures, that is, measures for which the observation interval is "sufficiently large" ($t \to \infty$). These measures are indeed approximations of the behavior of the system for a infinite, but long, time interval, where long means with respect to the interval of time between occurrences of events in the system. However, in some cases the steady state measures are not good approximations for the measures during a relatively "short" period of time.

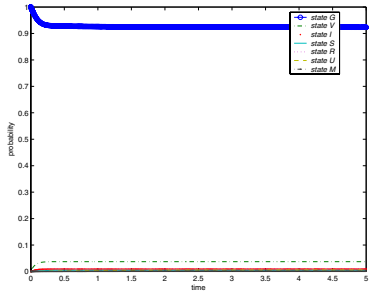
Before reaching the steady state, the system will go through a transient period. If the damage containment and recovery systems are not efficient enough, the system may never reach steady states, or take a very long time. The cumulative time distribution of contain and repair states will be dominant. Even through the steady state probability of good state is high, obviously we can not satisfy the system's performance. The limitation of steady state measures motivates us to observe the transient behaviors of different intrusion tolerant systems in this part. Figure 8 and 9 show the comparison results. We start the system from state G, which means $P_G(0) = 1$.

A better system's behaviors are shown in Figure 8. We assume that a better intrusion tolerant system has a good IDS, which can detect intrusion quickly and have a high detection rate and a low false alarm rate. Damage assessment and repair systems can locate and mask the intrusion quickly. As can be seen in Figure 8(a), a better system reaches steady state quickly. The probability of staying at state G is high, while the probabilities of staying at another states, like state I, R, and M, are very low. From Figure 8(b), we can also find that the cumulative time distribution of staying at state G is dominant, which means the system will spend most of time at good state. Since the damage assessment and repair system can accomplish their tasks quickly, the cumulative time distribution of state I, R, and M are low.

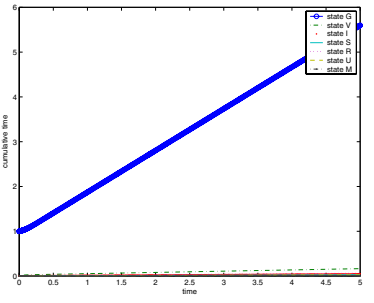
In Figure 9, we give an example of a poor system, which has a slow assessment and repair system. Compared with Figure 8, we can find that it takes a longer time for the system to reach steady states. The cumulative time of state G is not dominant. The system spends more time on damage assessment and repair.

7 Related Works

Despite that intrusion tolerance techniques, which gain impressive attention recently, are claimed to be able to enhance the system survivability, suitable and

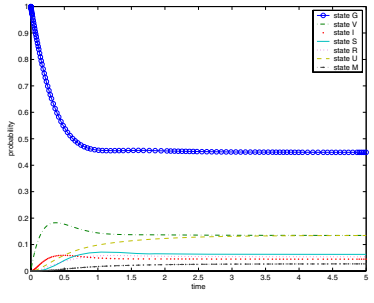


(a) transient probabilities of a good system

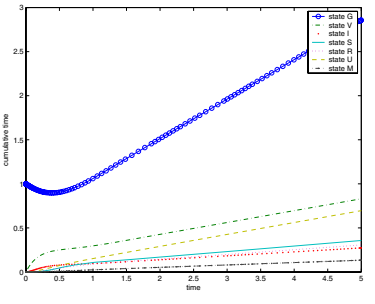


(b) cumulative time distribution of a good system

Fig. 8. Transient Behaviors of a good system



(a) transient probabilities of a poor system



(b) cumulative time distribution of a poor system

Fig. 9. Transient Behaviors of a poor system

precise measures to evaluate the survivability of an intrusion tolerant system are largely missed in the previous research. Most of the research in the literature report and discuss the survivable capability of their work from a qualitative point of view. Little research has proposed the quantitative evaluation metrics of survivability.

In [9] and [11], formal definitions of survivability are presented and compared with related concepts of reliability, availability, and dependability. [11] defined the survivability from several aspects and claimed that the big difference between reliability and survivability is that degraded services of survivable systems are acceptable to users, reliability assumes that the system is either available or not. However, the quantitative measurements of survivability and the level of degraded services are missing in that study.

The attacks and the response of an intrusion tolerant system are modeled as a random process in [5]. Stochastic modeling techniques are used to capture

the attacker behavior as well as the system's response to a security intrusion. Quantitative security attributes of the system are proposed in the paper. Steady-state behaviors are used to analyze the security characterization. A security measure called the *mean time (or effort) to security failure* is proposed. However, "good guestimate" values of model parameters were used in their experiments. And the validation of their models is missing in their work.

Efforts for quantitative validation of security have usually been based on formal methods [12]. [13] shows that probabilistic validation through stochastic modeling is an attractive mechanism for evaluating intrusion tolerance. The authors use stochastic activity networks to quantitatively validate an intrusion-tolerant replication management system. Several measures defined on the model were proposed to study the survivability provided by the intrusion tolerant system. The impacts of system parameters variations are studied in that work.

Although several survivability models and corresponding measurements were proposed in the literature, they are limited in evaluating the security attributions of an intrusion tolerant database system. Zhang and Liu [14] take the first step towards delivering database services with information assurance guarantees. In particular, (a) the authors introduce the concept of Quality of Integrity Assurance(QoIA) services; (b) a data integrity model, which allows customers or applications to quantitatively specify their integrity requirements on the services that they want the database system to deliver, is proposed; and (c) the authors present an algorithm that can enable a database system to deliver a set of QoIA services without violating the integrity requirements specified by the customers on the set of services.

An online attack recovery system for work flow is proposed in [4]. The behaviors of the recovery system are analyzed based on a Continuous Time Markov Chain model. Both steady-state and transient behaviors are studied in that paper. Only 'NORMAL', 'SCAN', and 'RECOVERY' three categories of states are considered in the model. The deficiency of intrusion detection and damage propagation are not considered in that model.

In [15], we have done detailed, quantitative evaluation on the impact of intrusion detection deficiencies on the performance and survivability by running TPC-C benchmark. However, only some ad hoc survivability metrics were used. Systematic survivability model and measurements were not proposed in [15].

8 Conclusion

In this paper, we extend the classic availability model to a new survivability model. Comprehensive state transition approaches are applied to study the complex relationships among states and their transition structure encoding sequential response of intrusion tolerant database systems facing attacks. Mean Time to Attack (MTTA), Mean Time to Detection (MTTD), Mean Time to Marking (MTTM), and Mean Time to Repair (MTTR) are proposed as basic measures of survivability. Quantitative metrics integrity and rewarding-availability are defined to evaluate the survivability of intrusion tolerant database systems.

A real intrusion tolerant database system is established to conduct comprehensive experiments running TPC-C benchmark transactions to validate the state transition models we established. Experimental results show the validity of proposed survivability models. To further evaluate the security of ITDB, we have done an empirical survivability evaluation, where maximum-likelihood methods are applied to estimate the values of the parameters used in our state transition models. The impacts of existing system deficiencies and attack behaviors on the survivability are studied using quantitative measures we defined. Our evaluation results indicate that (1) ITDB can provide essential database services in the presence of attacks, and (2) maintain the desired essential survivability properties without being seriously affected by various system deficiencies and different attack intensity.

References

1. Trivedi, K.S.: Probability and statistics with reliability, queuing and computer science applications. John Wiley and Sons Ltd. (2002)
2. Liu, P.: Architectures for intrusion tolerant database systems. In: Proceedings of 18th Annual Computer Security Applications Conference (ACSAC 2002). (2002) 311–320
3. Liu, P., Jing, J., Luenam, P., Wang, Y., Li, L., Ingsriswang, S.: The design and implementation of a self-healing database system. Journal of Intelligent Information Systems (JIIS) **23**(3) (2004) 247–269
4. Yu, M., Liu, P., Zang, W.: Self-healing workflow systems under attacks. In: Proceedings of 24th International Conference on Distributed Computing Systems (ICDCS 2004). (2004) 418–4025
5. Madan, B.B., Goseva-Popstojanova, K., Vaidyanathan, K., Trivedi, K.S.: A method for modeling and quantifying the security attributes of intrusion tolerant systems. Performance Evaluation **56**(1-4) (2004) 167–186
6. Lippmann, R., Fried, D., Graf, I., Haines, J., Kendall, K., McClung, D., Weber, D., Webster, S., Wyschogrod, D., Cunningham, R., Zissman, M.: Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. In: Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX). (2000) 12–26
7. Hu, Y., Panda, B.: A data mining approach for database intrusion detection. In: Proceedings of the 2004 ACM Symposium on Applied Computing (SAC). (2004) 711–716
8. Chen, W., Toueg, S., Aguilera, M.K.: On the quality of service of failure detectors. IEEE Transactions on Computers **51**(1) (2002) 13–32
9. Ellison, R.J., Fisher, D.A., Linger, R.C., Lipson, H.F., Longstaff, T.A., Mead, N.R.: Survivability: Protecting your critical systems. IEEE Internet Computing **3**(6) (1999) 55–63
10. TPC: Tpc-c benchmark. http://www.tpc.org/tpcc/ (2004)
11. Knight, J.C., Strunk, E.A., Sullivan, K.J.: Towards a rigorous definition of information system survivability. Volume 1. (2003) 78–89
12. Landwehr, C.E.: Formal models for computer security. ACM Computing Surveys **13**(3) (1981) 247–278

13. Singh, S., Cukier, M., Sanders, W.H.: Probabilistic validation of an intrusion-tolerant replication system. In: Proceedings of the International Conference on Dependable Systems and Networks (DSN 2003). (2003) 615–624
14. Zhang, J., Liu, P.: Delivering services with integrity guarantees in survivable database systems. In: Proceedings of the 17th Annual Working Conference on Data and Application Security. (2003) 33–46
15. Wang, H., Liu, P., Li, L.: Evaluating the impact of intrusion detection deficiencies on the cost-effectiveness of attack recovery. In: Proceedings of 7th International Information Security Conference (ISC 2004). (2004) 146–157