International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 3 Issue 5, August 2019 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Survival of an Intrusion Tolerance Database System

Su Thawda Win

Faculty of Computer Systems and Technologies, University of Computer Studies, Mandalay, Myanmar

How to cite this paper: Su Thawda Win "Survival of an Intrusion Tolerance Database System" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-3 | Issue-5, August 2019, pp.1748-1751, **https://doi.org/10.31142/ijtsrd26748**

IJTSRD26748

ABSTRACT

While traditional secure database systems rely on prevention control and are very limited in surviving malicious attack, an intrusion-tolerant database system can operate through attacks in such a way that the system can continue delivering essential services in the face of attacks. The emphasis of survivability is on continuity of operations, with the understanding that the security precautions cannot guarantee that systems will not be penetrated and compromised. In this paper, we propose a framework of model-based evaluation of the survivable intrusion tolerant database system. We focus on modeling the behaviors of an intrusion tolerant database system which can detect intrusions, isolate attacks, contain, assess, rejuvenate and repair limited in surviving malicious attacks. We contain the necessary quantitative metrics to measure the availability, integrity, and survivability. Quantitative measures are proposed to characterize the capability of a resilient database system surviving intrusions.

KEYWORDS: *Security, Availability, Survivability, Intrusion Tolerance, Software Rejuvenation, Database System*

I. INTRODUCTION

As society increasingly relies on database systems to store, manage, and access information digitally (e.g., database products are today a multi-billion dollar industry;

Database systems motivated 32% of the hardware server volume in 1995 [10], and 39% of the server volume in 2000), maintaining the integrity, availability, and confidentiality of databases is crucial.

Database security concerns the confidentiality, integrity, and availability of information stored in a database. A broad span of research addresses primarily how to protect the secrecy of a database, namely its confidentiality.

However, very limited research has been done on how to survive successful database attacks, which can seriously impair the integrity and availability of a database. Many large-scale database systems critical to businesses are expected to be available continuously and can only be stopped for repair at great cost. As a result, survivable database systems that can tolerate malicious attacks are getting increasingly important.

One critical step towards survivable database systems is intrusion detection. Intrusion detection systems monitor systems or network activities to discover attempts to disrupt or gain illicit access to systems. However, intrusion detection makes a system attack-aware but not attack resilient; that is, intrusion detection itself cannot maintain the integrity and availability of a database in face of attacks. To overcome the inherent limitation of intrusion detection, a broader perspective is introduced, saying that, in addition to detecting attacks, countermeasures to these successful attacks should be planned and deployed in advance [12]. In the literature, this is referred to as survivability or intrusion tolerance.

A database attack can be enforced at four possible levels: processor (or instruction) level, OS level, DBMS level, and transaction (or application) level. And in general, malicious insiders tend to attack at the transaction level and outsiders tend to attack at the other three levels. A study performed in [3] shows that most attacks are from insiders. As web-based applications (e.g., e-business) evolve, people have seen (and will see) more and more cyber attacks for a couple of reasons. For example, more critical and valuable information is now processed through the web, which is world-wide accessible. Expanding (successful) cyber attacks pressure applications to not only prevent unauthorized access, but also tolerate intrusions.

Although intrusion tolerance techniques, which gain impressive attention recently, are claimed to be able to enhance the system survivability, quantifying survivability metrics of computer systems is needed and important to meet the user requirements. 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. 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.

The goal of this paper is to develop a survivability model using a state transition graph to model an intrusion tolerant database system (ITDB). 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.

The rest of the paper is organized as follows. In section A, we discuss the related work. Section B gives an overview of the proposed intrusion tolerance database system framework. In section C, a series of state transition model are proposed. Next, quantitative measures of database system survivability are proposed in section C. We conclude our paper in section D.

A. Related Work

The need for intrusion tolerance, or *survivability*, has been recognized by many researchers in such contexts as *information warfare* [5]. Recently, extensive research has been done in general principles of survivability [7], survivable software architectures [9], survivable storage systems [13], etc. This research is helpful for database survivability, but the techniques cannot be directly applied to build intrusion tolerant database systems.

Some research has also been done in database intrusion tolerance. In [1], a fault tolerant approach is taken to survive database attacks where (a) several useful survivability phases are suggested, though no concrete mechanisms are proposed for these phases; (b) a color scheme for marking damage (and repair) and a notion of integrity suitable for partially damaged databases are used to develop a mechanism by which databases under attack could still be safely used.

There is also some work on OS-level database survivability. In [8] a technique is proposed to detect *storage jamming*, malicious modification of data, using a set of special *detect objects* which are indistinguishable to the jammer from normal objects. Modification on detect objects indicates a storage jamming attack. In [2], checksums are smartly used to detect data corruption. Similar to trusted database system technologies, both detect objects and checksums can be used to make ITDB more resistant to OS level attacks.

Zhang and Liu [14] take the first step towards delivering database services with information assurance guarantees. In particular, the authors introduce the concept of Quality of Integrity Assurance (QoIA) services; a data integrity model. They also 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.

In [6], formal definitions of survivability are presented and compared with related concepts of reliability, availability, and dependability. This paper 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. We propose a framework for survivable intrusion tolerant database system which can provide levels of data integrity and availability to applications in the face of attacks.

B. Proposed Survival Intrusion Tolerant Database System

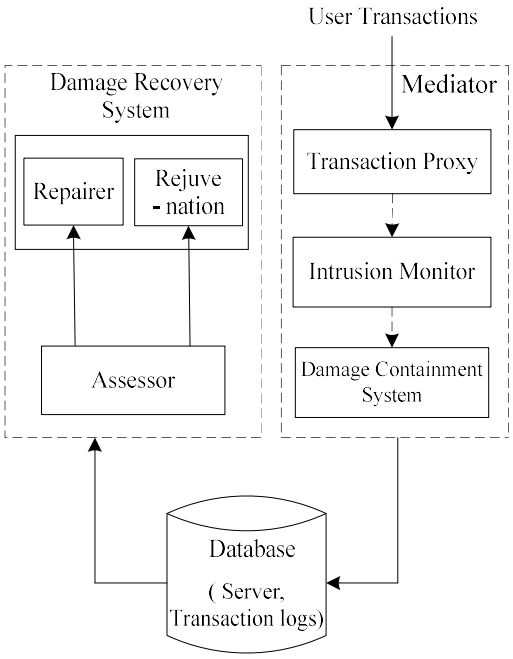


Figure 1 Proposed Survival Intrusion Tolerant Database System

As proposed for survival intrusion tolerant database system, we construct the flowchart in figure1. As mentioned in figure, there are two parts namely, Damager Recovery System and Mediating System.

In our mediator, "proxy" is a function which uses transaction and transaction processing call for each user to the database system. It is able to keep useful information about transaction's read and write operations which are important.

Afterwards, intrusion monitor will detect the transaction proxy and report wrong transactions to the damage containment system. This intrusion monitor subsystem has four aspects intrusion detection, application awareness, transaction level and developing new techniques for detection.

At the stage damage containment system, it is responsible to contain malicious information and data. Latency assessment of the damage will be reported to the Assessor. Otherwise, it will send to the database as a transaction log.

The responsibility of the assessor is to perform accurate damage assessment for the corresponding repair and rejuvenation. Then ITDB system traces damage spreading among transactions and capturing the dependent upon relationship for them. Accessing the coming out malicious data, it is decided to rejuvenate. If it is failed to rejuvenate, we consider for the last resort for repair.

Our proposed model provides the following 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".

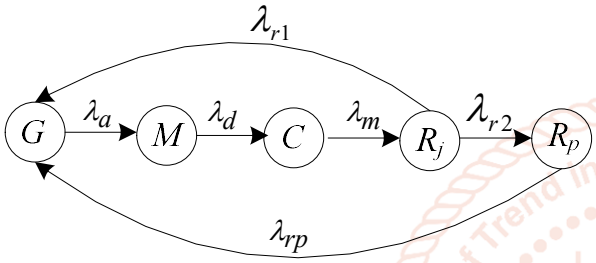
C. Modeling Intrusion Tolerant Database System

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.

II. Basic State Transition Model

Basic state transition model is shown in figure 2. There are five phases which are Good State *G*, Monitoring State *M*, Containment State *C*, Rejuvenation State *Rj*, and Repair State *Rp*. The five phases which attack penetration, error monitoring and detection, attack containment, damage assessment, rejuvenation and error recovery, describe the basic phenomenon for each intrusion tolerant system will encounter. These can lead to the basic requirements for the design and implementation of an intrusion tolerant database system.



G – Good State
M – Mark State
C – Containment State
R_j – Rejuvenation State
R_p – Repair State
Figure 2 Basic State Transition Model

Generally, traditional computer security system leads to the design of systems which will prevent to attack. If the strategies for prevention fail, the system is entered from Good stat G into the Monitoring state M during the exploration phases of an attack. If the attack is monitored successfully, intrusion tolerance will pick up where attack prevention leaves off. Next, the system enters the containment state in which all the damages by attack are contained. On the other hand, undamaged items are released and the system enters to the rejuvenation state. If not, repair process will compensate all the damages and the system return back to the good state *G*...

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 expect time for the intrusion to be detected; $1/\lambda_m$ is the mean time to mark (MTTM), the expected time for the system to mark "dirty" data items; $(1/\lambda_{r1} + 1/\lambda_{r2})$ is the mean time to rejuvenate ($MTTR_J$), the expected time for the system to rejuvenation, and $1/\lambda_{rp}$ is the mean time to repair ($MTTR_p$), they expect time for the system to repair damaged data items.

III. State Transition Model Analysis

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

$$\frac{dP(t)}{d(t)} = P(t)\ Q \qquad (1)$$

Where P (0) represents the initial probability vector (at time t = 0). In addition, cumulative probabilities are sometimes of interest. Let $L(t) = \int_o^t P(u)du$; 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 :

$$\frac{dL(t)}{dt} = L(t)\ Q + P(0) \text{ where } L(0) = 0. \qquad (2)$$

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

$$\pi\ Q = 0, \sum_{i\in S} \pi_i = 1 \qquad (3)$$

By solving the equation 1, 2 and 3, we can get some important survivable metrics of an intrusion tolerant database system.

Consider state space *S* = { *G, M, C, Rj, Rp* } . The generator matrix *Q* for the basic state transition model is ;

$$Q = \begin{bmatrix} -\lambda_a & \lambda_a & 0 & 0 & 0 \\ 0 & -\lambda_d & \lambda_d & 0 & 0 \\ 0 & 0 & -\lambda_m & \lambda_m & 0 \\ \lambda_{r1} & 0 & 0 & -(\lambda_{r1}+\lambda_{r2}) & \lambda_{r2} \\ \lambda_{rp} & 0 & 0 & 0 & -\lambda_{rp} \end{bmatrix}$$

By solving the equations (2) and (3), we have steady state probabilities as follows.

$$\pi_G = \frac{1/\ \lambda_a}{1/\lambda_a + 1/\lambda_d + 1/\lambda_m + (1/\lambda_{r1} + 1/\lambda_{r2}) + 1/\lambda_{rp}} \qquad (4)$$

$$\pi_M = \frac{1/\lambda_d}{1/\lambda_a + 1/\lambda_d + 1/\lambda_m + (1/\lambda_{r1} + 1/\lambda_{r2}) + 1/\lambda_{rp}} \qquad (5)$$

$$\pi_C = \frac{1/\lambda_c}{1/\lambda_a + 1/\lambda_d + 1/\lambda_m + (1/\lambda_{r1} + 1/\lambda_{r2}) + 1/\lambda_{rp}} \qquad (6)$$

$$\pi_{R_j} = \frac{(1/\lambda_{r1} + 1/\lambda_{r2})}{1/\lambda_a + 1/\lambda_d + 1/\lambda_m + (1/\lambda_{r1} + 1/\lambda_{r2}) + 1/\lambda_{rp}} \qquad (7)$$

$$\pi_{R_p} = \frac{1/\lambda_{rp}}{1/\lambda_a + 1/\lambda_d + 1/\lambda_m + (1/\lambda_{r1} + 1/\lambda_{r2}) + 1/\lambda_{rp}} \qquad (8)$$

IV. 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 [4]. 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 our model, survivability is quantified in terms of integrity and availability.

V. Integrity

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. From this definition, the integrity for the basic state transition model which we can get:

$$I = \pi_G + \pi_C + \pi_{R_j} + \pi_{R_p}$$

$$= \frac{MTTA + MTTM + MTTR_j + MTTR_p}{MTTA + MTTD + MTTM + MTTR_j + MTTR_p}$$

VI. Availability

Availability is defined as a fraction of time that the system is providing service to its users [11]. It is also defined as a fraction of time that the all clean data items are accessible. If the clean data cannot be accessed, it is a loss of service to users. ITDB will release the all contained clean data items after damage assessment. Availability can be defined by :

$$A = \pi_G + \pi_{R_j} + \pi_{R_p}$$

$$= \frac{MTTA + MTTR_j + MTTR_p}{MTTA + MTTD + MTTM + MTTR_j + MTTR_p}$$

VII. Conclusion

In this paper, we have presented a survivability model by means of availability and integrity of transaction-level through intrusion-tolerant database system. Comprehensive state transition approaches are applied to study the complex relationships among states and their transition states 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), Mean time to Rejuvenation ($MTTR_j$), and Mean Time to Repair ($MTTR_p$) are proposed as the basic measures of survivability and tolerance. Quantitative metrics are used to measure the integrity and availability which are defined to evaluate the survivability of intrusion tolerance database system. As our later works to proceed, it is required to validate the empirical data for the survivability model we proposed.

References

[1] P. Ammann, S. Jajodia, C.D. McCollum, and B. T. Blaustein, "Surviving information warfare attacks on databases", In Proceedings of the IEEE Symposium on Security and Privacy, pages 164–174, Oakland, CA, May 1997.

[2] Barbara, R. Goel, and S. Jajodia, "Using checksums to detect data corruption", In Proceedings of the 2000 International Conference on Extending Data Base Technology, Mar 2000.

[3] Carter and Katz., "Computer Crime: An Emerging Challenge for Law Enforcement. FBI Law Enforcement Bulletin, 1(8), December 1996.

[4] R. J. Ellison, D. A. Fisher, R. C. Linger, H. F. Lipson, T. A. Long staff, N.R. Mead, "Survivability: Protecting your critical systems", IEEE Internet Computing 3(6)(1999), pp. 55–63.

[5] R. Graubart, L. Schlipper, and C. McCollum, "Defending database management systems against information warfare attacks", Technical report, The MITRE Corporation, 1996.

[6] J. C. Knight, E. A. Strunk, K. J. Sullivan, "Towards a rigorous definition of information system survivability. Volume 1. (2003) 78–89

[7] J. Knight, K. Sullivan, M. Elder, and C. Wang, "Survivability architectures: Issues and Approaches", In Proceedings of the 2000 DARPA Information Survivability Conference & Exposition, pages 157–171, CA, June 2000.

[8] J. McDermott and D. Goldschlag. "Towards a model of storage jamming", In Proceedings of the IEEE Computer Security Foundations Workshop, pages 176–185, Kenmare, Ireland, June 1996.

[9] Stavridou, "Intrusion tolerant software architectures", In Proceedings of the 2001 DARPA Information Survivability Conference & Exposition, CA, June 2001.

[10] P. Stenstrom and et al. Trends in shared memory multiprocessing. IEEE Computer, (12):44–50, December 1997.

[11] K. S. Trivedi, " Probability and statistics with reliability, queuing and computer science applications", John Wiley and Sons Ltd. (2002).

[12] H. Wang, P.Liu, and L. Li, "Evaluating the Survivability of Intrusion Tolerant Database Systems and the Impact of Intrusion Detection Deficiencies", Int. J . Accounting and Performance Evaluation.

[13] J. J. Wylie, M. W. Bigrigg, J. D. Strunk, G. R. Ganger, H. Kiliccote, and P. K. Khosla., "Survivable information storage systems", IEEE Computer, (8):61–68, August 2000.

[14] J. Zhagn, P. Liu, "Delivering services with integrity guarantees in survivable database system", In: Proceedings of the 17th Annual Working Conference on Data and Application Security. (2003) 33-46.