A Self-healing Mechanism for an Intrusion Tolerance System

Bumjoo Park[1], Kiejin Park[2], and Sungsoo Kim[3]

[1] Samsung Electronics, Suwon, Korea
bumjoo@samsung.com
[2] Division of Industrial & Information Systems Engineering,
Ajou University, Suwon, Korea
kiejin@ajou.ac.kr
[3] Graduate School of Information and Communication,
Ajou University, Suwon, Korea
sungsoo@ajou.ac.kr

Abstract. The dependability analysis of an ITS (Intrusion Tolerance System - a system that performs continuously minimal essential services even when the computer system is partially compromised because of intrusions) is essential for the design of the ITS. In this paper, we applied self-healing mechanism, the core technology of autonomic computing to analyze the dependability of the ITS. In other words, we described the state transition of the ITS composed of a primary server and a backup server utilizing two factors of self-healing mechanism (fault model and system response) and analyzed it using M/G/1 queuing technique. We also evaluated the availability of the ITS through simulation experiments.

1 Introduction

With the intrusion tolerance method, the network-based computer systems continuously provides minimal essential services even when the system is partially compromised because of the internal and/or external intrusions such as DoS (Denial of Service) [1]. Application of the ITS (Intrusion Tolerance System) method has been arousing a lot of interest recently. This phenomenon results from a limitation of the well-known security technologies such as firewall, vaccine and intrusion detection that have individual weaknesses causing them to be vulnerable to accidental or intentional attacks and faults that are not known to them. Additionally, when we summarize the characteristics of attacking tools recently discovered, they have characteristics such as being stealthy, distributed, automated and performing as an

[1] This work is supported by an Ajou University grant.
[2] This work is supported in part by the 21st Century Frontier Research and Development (R&D) Program "National Center of Excellence in Ubiquitous Computing and Network" from the Ministry of Information and Communication of Korea.
[3] This work is supported in part by the Ministry of Education & Human Resources Development of Korea (Brain Korea 21 Project supervised by Korea Research Foundation).

S. Katsikas, J. López, G. Pernul (Eds.): TrustBus 2005, LNCS 3592, pp. 41 – 49, 2005.

agent. Therefore, the problems are bigger than ever. The intrusion tolerance method is being actively studied as prevention and countermeasure against various malicious attacks to network-based computer systems.

Intrusion tolerance is different from intrusion detection. It does not guarantee that it will beat all the malicious attacks but it guarantees that it will provide services continuously by the ITS with dependability (reliability, availability, maintainability, safety, survivability etc.) even when some parts of the system are damaged because of the successful malicious attacks. In Europe, IST(Information Society Technologies) performed studies through the MAFTIA(Malicious-and Accidental-Fault Tolerance for Internet Applications) project to develop an ITS [2], and the USA is performing intrusion tolerance related projects such as HACQIT(Hierarchical Adaptive Control of Quality of service for Intrusion Tolerance), SITAR(Scalable Intrusion Tolerant Architecture), and ITUA(Intrusion Tolerance by Unpredictable Adaptation) through OASIS(Organically Assured and Survivable Information System) program of DARPA(Defense Advanced Research Projects Agency) [3,4,5].

On the other hand, a new approach using a self-healing mechanism is being proposed [6] where one of the four core technologies of autonomic computing is utilized to implement an ITS with dependability. Although the self-healing method includes various factors related to the dependability of the system just like fault tolerant methods, self-healing provides broader protection than the existing fault tolerant method, in that it can provide appropriate responses to the unexpected internal and external attacks together self-optimization, self-configuration, and self-protection [7].

2 Related Works

In [3], they adopted design diversity to enforce fault tolerant functions of the ITS and configured that the primary server and the backup server would have different OS and web server applications. However, as both servers were interoperating using the Hot-standby method (e.g., dynamic redundancy), both of them can be damaged from external attacks. Study [8] shows the state transition diagram to describe the dynamic abnormal behaviors of intrusion tolerance against external attacks. In this study, they performed the study on intrusion tolerance framework regarding how to model the vulnerabilities and risk factors of the system. In [9], they attempted quantitative performance analysis of several attacks such as DoS based on state transition diagrams. ITS frameworks can be divided into two, which is a layer based one and a replication based one. The layer-based structure may be applied to a single host. In this model, data integrity will be emphasized. The replication-based structure is to increase availability of the distributed computing environments. However, because of the increase of replications, secrecy would be threatened [10]. On the other hand, [11] shows examples applying self-healing technology to enhance the dependability of the distributed embedded system.

As the ITS is to respond to dynamic abnormal behaviors that are made by attackers according to system vulnerabilities or risk factors, it should be able to describe their state changes. In other words, we need to identify the attack type and present state of the ITS and express it into the state transition model in order to make a quantitative

performance analysis of the transition process from the damaged to normal state. In this paper, we described the state transition of an ITS composed of a primary server and a backup server utilizing the two factors of the self-healing mechanism (fault model and system response) and analyzed it using M/G/1 queuing technique. We also evaluated the availability of the ITS through simulation experiments.

3 Intrusion Tolerance System Utilizing Self-healing Mechanism

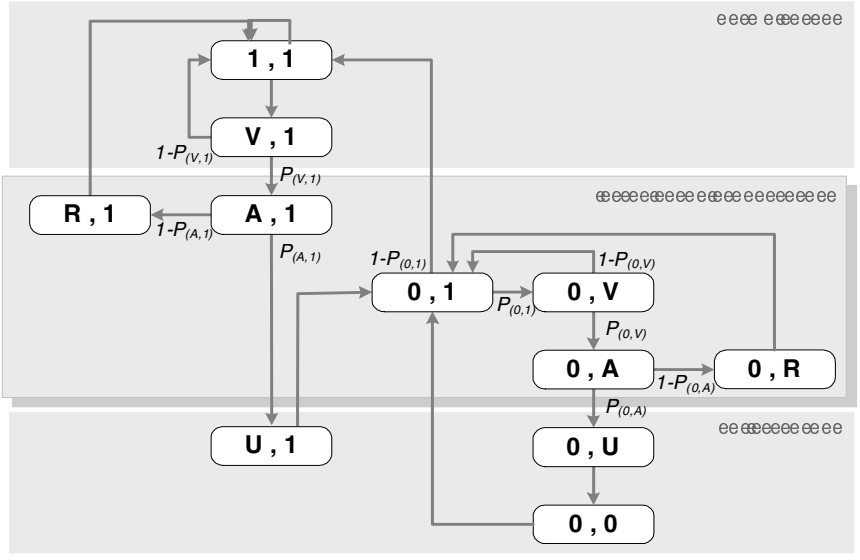
Self-healing is the core technology of autonomic computing to enhance dependability of the system by minimizing malfunctions of the system through the detection, diagnosis and repair of the faults or errors arising from external attacks or internal system problems [12]. To implement self-healing technology as a system, there should be definitions of four factors such as fault model, system response, system completeness and design context [6].

The fault model defines the characteristics of faults that the system should tolerate and system response is a detailed definition regarding the fault detection, response to the fault and recovery strategy against external attacks. For example, DoS such as SYN Flood and Smurfing may cause the performance degradation of system resource by sending malicious requests to a certain server such as DNS(Domain Name Server). However, even in this situation, the 'Gracefully Degradation' concept that guarantees the essential services of the system should be included in elements in system response. On the other hand, system completeness is regarding the element should be implemented to overcome the structural imperfection of the system implementation in the real world. Design context shows self-healing element to secure homogeneity and linearity of the system to be implemented. To let the ITS have self-healing functions, a system state transition diagram was drawn with the detailed items related to fault model and system response among the four self-healing components described above.

Figure 1 shows the state transition of the Cold-standby ITS reflecting self-healing components. It also shows the detailed factors of the fault model responding to DoS, and elements corresponding to those factors such as fault detection, degradation, fault response and fault recovery. The followings are the assumptions that are applied for the intrusion tolerance system modeling.

- The switchover mechanism between the primary server and the backup server of ITS would follow the Cold-standby method.
- The sojourn time in each state of ITS would follow a general distribution.
- The system is properly working in the initial state and intrusion would be only possible in this state.
- After the switchover between the primary server and the backup server, the works will be transferred to the primary server when the backup server is only in normal state (0,1).

When vulnerability is exposed in the state of normal operation of both servers (1,1), the ITS will be transferred to (V,1). If the intrusion tolerance module defends all the vulnerability attacks through network traffic and IP address analysis, it will be recovered after a specified time to the initial state. However, if it does not happen, the primary server will be attacked (A,1) with the probability of $P_{(V,1)}$. When the attack state of the primary server continues for a certain time, the system damage will be



※ State (Primary server, Backup server)

※ V : Vurnerable state, A : Attack state, R : Rejuvenation state, U : Undetected state

Fig. 1. State transition diagram of intrusion tolerance system

accumulated. If the intrusion diagnosis module analyzes system CPU load and memory state and the meaningful performance degradation is detected at the probability of 1- $P_{(A,1)}$, it will transfer primary server into rejuvenation(restoration, reconfiguration or recovery) state (R,1). If it cannot diagnose the performance degradation, it is transferred to (U, 1), which means Undetected. Finally, the switchover takes place and the backup server will do the job on behalf of the primary server (0,1). To prevent the simultaneous down of both servers by external attacks, Cold-standby configuration was adopted. In this case, the time needed for switchover will be prolonged. The process from the state that the backup server plays the role of the primary server (0,1) to the state that the backup server is down (0,0) is same as the job transition from the primary server to the backup server in the initial state.

Generally, in Figure 1, the normal stage is the one where system degradation does not happen at all. The intrusion tolerance stage is the one where there is a certain level of damage but the system performs its essential services. Failure stage is the one where the primary server is not recovered and at the same time, the backup server cannot provide services regardless of the operation of the ITS.

To calculate the availability of the steady-state of the proposed ITS, the stochastic process of equation 1 was defined. Through SMP (Semi-Markov Process) analysis applying M/G/1 whose service time is general distribution, we calculated the steady-state probability in each state.

$$X(t): \ t > 0$$

$$X_S = \{(1,1),(V,1),(A,1),(R,1),(U,1),(0,1),(0,V),(0,A),(0,R),(0,U),(0,0)\} \quad (1)$$

As all the states shown in Figure 1 are attainable to each other, they are irreducible. Additionally, as they do not have a cycle and can return to a certain state they satisfy the ergodicity (aperiodic, recurrent, and nonnull) characteristics. Therefore, there is a probability in the steady-state of SMP for each state of ITS and each corresponding SMP can be induced by embedded DTMC (Discrete-time Markov Chain) using transition probability in each state [13].

If we define the mean sojourn times in each state of SMP as h_i's and define DTMC steady-state probability as d_i's, the steady-state probability in each state of SMP (π_i) can be calculated like equation 2 [14].

$$\pi_i = \frac{d_i h_i}{\sum_j d_j h_j}, \qquad \mathrm{i, j} \in \mathrm{X_S} \tag{2}$$

Whereas, steady-state probability of DTMC d_i's will have the following relationship as shown in equation 3 and equation 4.

$$\vec{d} = \vec{d} \cdot P \tag{3}$$

$$\sum_i d_i = 1 \quad i \in X_S \tag{4}$$

where, $\vec{d} = [d_{(1,1)}\ d_{(V,1)}\ d_{(A,1)}\ d_{(R,1)}\ d_{(U,1)}\ d_{(0,1)}\ d_{(0,V)}\ d_{(0,A)}\ d_{(0,R)}\ d_{(0,U)}\ d_{(0,0)}]$ and P is transition probability matrix of DTMC expressed by the transition probability in each state of X_S in Figure 1 ($p_{(i,j)}$). If we calculate the steady-state probability of DTMC from them, the following equation is made.

$$d_{(1,1)} = \frac{1 - p_{(0,1)}}{2(1 + p_{(V,1)})(1 - p_{(0,1)}) + p_{(V,1)} p_{(A,1)} (1 + p_{(0,1)} + 2p_{(0,1)} p_{(0,V)} + p_{(0,1)} p_{(0,V)} p_{(0,A)})}$$

$$d_{(V,1)} = d_{(1,1)}$$

$$d_{(A,1)} = d_{(V,1)} p_{(V,1)}$$

$$d_{(R,1)} = d_{(A,1)} (1 - p_{(0,1)})$$

$$d_{(U,1)} = d_{(A,1)} p_{(A,1)}$$

$$d_{(0,1)} = d_{(U,1)} + d_{(0,V)} (1 - p_{(0,V)}) + d_{(0,R)} + d_{(0,0)} \tag{5}$$

$$d_{(0,V)} = d_{(0,1)} p_{(0,1)}$$

$$d_{(0,A)} = d_{(0,V)} p_{(0,V)}$$

$$d_{(0,R)} = d_{(0,A)} (1 - p_{(0,A)})$$

$$d_{(0,U)} = d_{(0,A)} p_{(0,A)}$$

$$d_{(0,0)} = d_{(0,U)}$$

On the other hand, if we put the DTMC steady-state probability calculated in equation 5 into equation 2, we can have the steady-state probability of each state of SMP (π_i). The system availability in the steady-state is defined as equation 6, which is same as the exclusion of the probability of being in (U,1), (0,U) and (0,0) in each state of X_S in the state transition diagram.

$$Availability = 1 - (\pi_{(U,1)} + \pi_{(0,U)} + \pi_{(0,0)}) \tag{6}$$

4 Simulation Analysis and Availability Enhancement Methods

To analysis the SMP model for ITS, we need to set parameters for the transition probability and the mean sojourn time in each state. In this paper, simulations were

Table 1. Simulation Parameter

Input Variables	Set Value
Mean Sojourn Time	$h_{(1,1)} = 0.5$, $h_{(V,1)} = 1/3$, $h_{(A,1)} = 0.25$, $h_{(U,1)} = 0.5$, $h_{(R,1)} = 0.2$, $h_{(0,1)} = 0.5$ $h_{(0,V)} = 1/3$, $h_{(0,A)} = 0.25$, $h_{(0,R)} = 0.2$, $h_{(0,U)} = 0.5$, $h_{(0,0)} = 0.5$
Transition Probability	Among 5 transition probabilities ($p_{(V,1)}$, $p_{(A,1)}$, $p_{(0,1)}$, $p_{(0,V)}$, $p_{(0,A)}$), we fixed 3 values and changed 2 values (from 0 to 1) (eg . : $p_{(A,1)} = p_{(0,V)} = p_{(0,A)} = 0.5$, $0 < p_{(V,1)}, p_{(0,1)} < 1$)

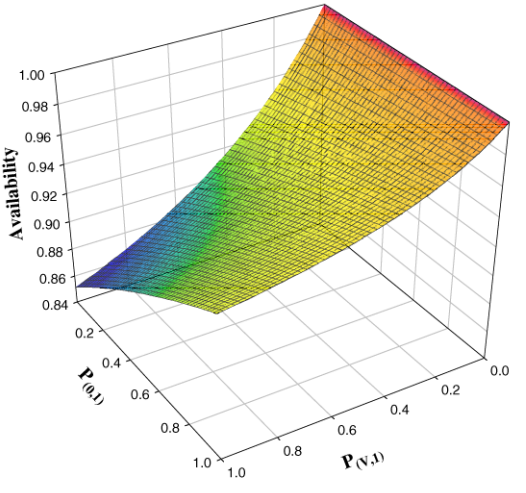


Fig. 2. *Availability analysis according to the changes in* $P_{(V,1)}$ and $P_{(0,1)}$

made based on the values shown in Table 1 [15,16]. As the mean sojourn time in each state does not follow a specific distribution, the values are only meaningful as relative difference. To analyze the independent influences of each transition probability at the 5 points shown in the state transition diagram, we set the initial transition probabilities with the same value, which is 0.5.

Figure 2 shows the system availability fluctuation trend according to the changes in probability that the primary server is attacked because of non-detection of vulnerability ($P_{(V,1)}$) and the probability that the backup server is exposed to the vulnerability ($P_{(0,1)}$), in order to identify the influences of initial state responding competencies of the ITS on the availability in the environments with external malicious attacks.

When the Cold-standby ITS proposed in Figure 1 is configured, the availability is increased when the primary and backup server can detect abnormal behaviors of the system in the initial state before they are exposed to the attacks or vulnerable environments. When $P_{(V,1)}$ and $P_{(0,1)}$ are getting bigger (In other words, detection capabilities in initial state are degraded), the availability of the system is reduced dramatically. However, if $P_{(V,1)}$ is greater than 0.5 and $P_{(0,1)}$ gets bigger, the availability will be increased. It is because the bigger $P_{(V,1)}$ is, the bigger the probability that the primary server fails (U,1) will be. Therefore, even when the switchover is made from the primary server to the backup server, the continued service by the backup server through intrusion tolerance in the state of (0,V) and (0,A) rather than the service through the immediate recovery of the primary server will be better, because it will reduce the probability to make the system be in the state of (U,1).

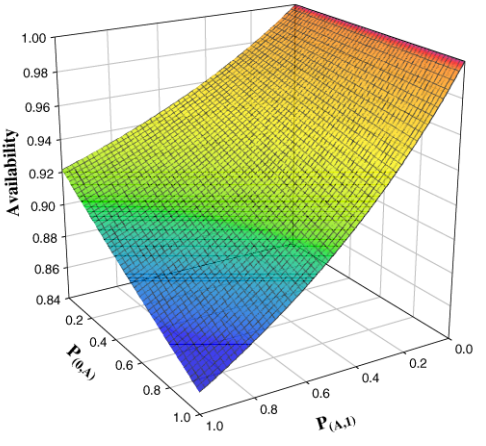


Fig. 3. *Availability analysis according to the changes in* $P_{(A,1)}$ and $P_{(0,A)}$

Figure 3 shows the changes in availability according to the changes of the attack success probability to the primary server, $P_{(A,1)}$ and to the backup server, $P_{(0,A)}$ in

order to judge the system response capabilities in the situations exposed to attacks. In case of $P_{(A,1)}$ is zero regardless of the transition probability related to detection capabilities of abnormal behavior in the initial state($P_{(V,1)}$ and $P_{(0,1)}$), we can see from the graph that the availability is ideal (1.0). In other words, if we can detect meaningful performance degradation immediately in initial state through diagnosis functions of the ITS in (A,1) and (0,A) where primary-backup servers are exposed to attacks, we can guarantee availability by returning the system to the initial state through the switchover to rejuvenation state. However, if $P_{(A,1)}$ and $P_{(0,A)}$ approach 1, the probabilities that the system will be put into no service state which are (U,1),(0,U) and (0,0) and the availability will be reduced.

On the other hand, in Figure 2 and Figure 3, when $P_{(A,1)}$ and $P_{(0,A)}$ are 1, the availability is nearly the same as that in the state where $P_{(V,1)}$ and $P_{(0,1)}$ have the worst values in the system. It is because even though the system does not detect abnormal behavior in the initial state, the structure of the Cold-standby ITS reduces the probability of no service or system down thanks to switchover, recovery and rejuvenation in the environments with external malicious attacks and thereby the availability will not be reduced any more.

5 Conclusion

In this paper, it was proposed to graft the self-healing mechanism, the core technology of autonomic computing in order to analyze the dependability of the ITS. We defined 11 states of a Cold-standby ITS composed of a primary server and a backup server and analyzed system availability by calculating DTMC steady-state probability and SMP steady-state probability through the transition probability and the mean sojourn time of each state. In the future, we will study how to improve system dependability through considering system completeness and design context in addition to the two factors of the self-healing mechanism that have already been considered in this paper.

References

[1] F. Wang, R. Uppalli, and C. Killian, "Analysis of Techniques for Building Intrusion Tolerant Server Systems," Proceedings of Military Communications Conference, pp. 729-734, Oct. 2003.

[2] http://www.laas.research.ec.org/maftia/

[3] J. Reynolds, et. al., "On-line Intrusion Detection Attack Prevention Using Diversity Generate-and-Test, and Generalization," Proceedings of the 36th Annual Hawaii International Conferences on System Sciences, pp. 335-342, Jan. 2003.

[4] F. Wang, et. al., “SITAR : A Scalable Intrusion-Tolerant Architecture for Distributed Services,” Proceedings of the Foundations of Intrusion Tolerant Systems, pp. 359-367, 2003.

[5] T. Courtney, et. al., "Providing Intrusion Tolerance with ITUA," Proceedings of the International Conference on Dependable Systems & Networks, pp. C-5-1 - C-5-3, June 2002.

[6] P. Koopman, "Elements of the Self-Healing System Problem Space," Workshop on Architecting Dependable Systems, pp. 31-36, May 2003.
[7] D. Chess, C. Palmer, and S. White, "Security in an Autonomic Computing Environment," IBM Systems Journal, Vol. 42, No.1, pp. 107-118, 2003.
[8] K. Goseva-Popstojanova, et. al., "Characterizing Intrusion Tolerant Systems using a State Transition Model," DARPA Information Survivability Conference and exhibition, Vol 2, pp. 211-221, June 2001.
[9] D. Wang, B. Madan, and K. Trivedi, "Security Analysis of SITAR Intrusion Tolerance System," Proceedings of the ACM Workshop on Survivable and Self-Regenerative Systems, pp. 23-32, Oct. 2003.
[10] G. Kim, M. Choi, and K. Lee, "Classification of the Intrusion Tolerant Systems and Integrated Framework for Survivability Enhancement," The Korea Information Processing Society Transactions, Vol. 10C, No. 3, pp.295-304, 2003.
[11] C. Shelton, P. Koopman, and W. Nace, "A Framework for Scalable Analysis and Design of System-Wide Graceful degradation in distributed Embedded Systems, " Eighth IEEE International Workshop on Object-oriented Real-time Dependable Systems, pp.156-163, Jan. 2003.
[12] J. Kephart, and D. Chess, "The Vision of Autonomic Computing," IEEE Computer, Vol. 36, No. 2, pp. 41-50, 2003.
[13] L. Kleinrock, Queueing Systems: Volume 1 Theory, John Wiley & Sons, pp. 417, 1975.
[14] K. Trivedi, Probability and Statistics with Reliability Queueing and Computer Science Applications, John Wiley & Sons, Inc., pp. 472, 2002.
[15] B. Madan, et. al., "Modeling and Quantification of Security Attributes of Software Systems," International Conference on Dependable Systems and Networks, pp. 505-514, June 2002.
[16] B. Madan, et. al., "A method for modeling and quantifying the security attributes of intrusion tolerant systems," Performance Evaluation, Vol. 56, Issues 1-4, pp. 167-186, 2004.