JCSE Journal of Computing Science and Engineering

Invited Paper

Journal of Computing Science and Engineering,
Vol. 7, No. 4, December 2013, pp. 242-250

A Survey on Intrusion-Tolerant System

Seondong Heo, Pyeong Kim, Yongjoo Shin, Jungmin Lim, Dongyoung Koo, Yonggon Kim, Ohmin Kwon, and Hyunsoo Yoon*

Department of Computer Science, Korea Advanced Institute of Science and Technology, Daejeon, Korea
sdheo@nslab.kaist.ac.kr, pkim@nslab.kaist.ac.kr, yj shin@nslab.kaist.ac.kr, jmlim@nslab.kaist.ac.kr
dykoo@nslab.kaist.ac.kr, ygkim@nslab.kaist.ac.kr, omkwon@nslab.kaist.ac.kr, hyoon@nslab.kaist.ac.kr

Abstract

Many information systems that provide useful services to people are connected to the Internet for convenience and efficiency. However, improper accessibility might make the systems susceptible to a variety of attacks. Although existing security solutions such as an intrusion detection system, intrusion prevention system, and firewalls have been designed to protect against such attacks, it is impossible to block all kinds of attacks. Furthermore, most of the proposed solutions require information about attacks for efficient prevention. Research on intrusion-tolerant systems (ITSs) have been conducted in order to continue providing proper services in threatening environments. The purpose of an ITS is to survive against every intrusion, rather than to prevent them. In this paper, previous studies on ITS are introduced and classified according to the centric scheme as middleware-based ITS, hardware-based ITS, and recovery-based ITS. Recent research focusing on adaptive transformation schemes is also introduced.

Category: Ubiquitous computing

Keywords: Intrusion-tolerant system; ITS; Availability; Security

I. INTRODUCTION

Improvements in communication and networking technologies have enabled various types of information services, such as social networking, data searching, and commercial banking via the Internet. Such openness provides convenience and efficiency, but also has fundamental vulnerabilities.

Traditional protection techniques, such as intrusion detection systems (IDS) [1], intrusion prevention systems, and firewalls [2], sometimes achieve excellent attack detection and prevention by analyzing previous intrusion records. However, because they are based on existing attack signatures, they cannot detect new and unreported intrusions. A zero-day attack is a representative example. To moderate such a problem, an anomaly-based IDS was proposed, but it had substantial disadvantages such as higher false positive rates. Situations in which an attack is successful and damages the system performance must be considered. To provide service reliability and survivability in a threatening environment, the intrusion-tolerant system (ITS), an advanced-concept security solution, was proposed.

Many studies on ITS have been introduced, and they have been based on two principles: redundancy and diversity. Since ITS must provide reliable service even in the vent of damage, redundancy is a mandatory mechanism to avoid a single-point-of-failure problem. However, homogeneous redundancy still retains common vulnerabilities that can be utilized by attackers. Diversity can

Open Access **http://dx.doi.org/10.5626/JCSE.2013.7.4.242** **http://jcse.kiise.org**

Received 10 July 2013, **Accepted** 30 July 2013
*Corresponding Author

 pISSN: 1976-4677 eISSN: 2093-8020

address this problem by implementing functions in various ways. For example, a system administrator operating Web servers can use different operating systems and software to remove common weaknesses.

In this survey, representative studies on ITS are presented. Classification is provided according to the operational strategies to supply various technical insights.

The rest of the paper is organized as follows. Section II is about conventional studies on middleware-based ITS, hardware-based ITS, and recovery-based ITS. In Section III, recent work developed from the conventional studies is presented. The paper ends with a conclusion in Section IV.

II. CONVENTIONAL WORK

A. Hardware-Based ITS

Hardware-based ITS utilizes hardware architectures, which are specially designed to prevent certain classes of attacks to maintain crucial services. Hierarchical adaptive control of quality of service for intrusion tolerance architecture includes the diverse replication [3] and network-based IDS [1], as shown in Fig. 1. The firewall is responsible for the qualified access to the server from the Internet. The monitor manages the servers, and reconfigures services immediately using the private communication channels if necessary. It is possible to assemble the architecture with commercial off-the-shelf (COTS) servers. Nevertheless, it is not adequate for dealing with more complicated situations, because the main purpose of the architecture is to serve for a short period of time in a threatening environment. Flexible extension is not supported either.

To manage the reliability of a system, designing protection and adaptation into a survivability architecture has various zones and layers to include approaches of adversaries [4]. The main security policy of protection is adapted at the innermost zone, which uses a traditional intrusion detection. The architecture integrates defense principles, such as redundancy and static diversity, detection and correlation, and adaptive response. Also, it is possible to improve the survivability of the architecture, employing adaptive middleware, cyber-defense mechanisms (IPsec, access control, etc.), and high-watermark technique [5]. Even though the architecture has high resiliency against attacks, it requires expert operators to respond to attacks.

Self-cleansing intrusion tolerance with hardware-enforced security (SCIT/HES) architecture is an improved version for resilience against attacks that uses a hardware-based generic framework, which uses centralized control [6]. The SCIT trusted interface modules (TIMs) in the architecture provide unidirectional communication links from the controller to each server. Server-side TIM also contains an SCIT switch that physically cuts off the connections from the online server to the internal network, which includes the controller and trusted storage server. Therefore, the controller can keep its pristine state and the control states of each server.

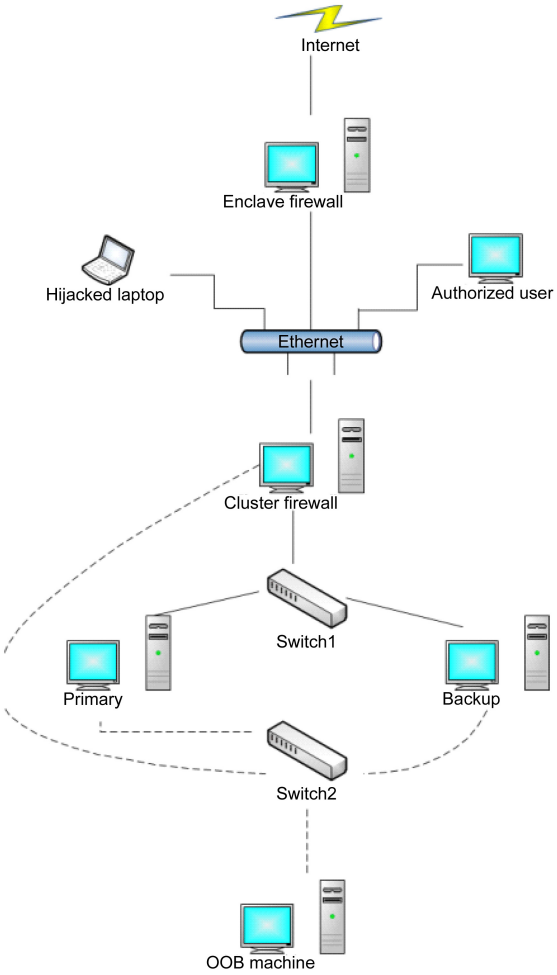

Fig. 1. Hierarchical adaptive control of quality of service for intrusion tolerance architecture. OOB: out-of-band.

B. Middleware-Based ITS

Similar to hardware-based ITS architecture, middleware-based ITS architecture provides a function to find intrusion or attacks using IDS or detection techniques. Proposed architectures also exploit redundancy and diversity to design fault-tolerance systems. An intrusion-tolerant model was proposed using several middleware components, such as proxies or monitoring systems.

The Willow architecture [7] protects the entire system with the detection of malicious faults, analysis of system vulnerabilities, and reconfiguration of a distributed computing environment. It is built in a comprehensive way to protect critical distributed applications. The powerful reconfiguration is a key feature of the overall system, and it

includes a general control structure that is capable of sensing the network state, analyzing it, and estimating required changes.

Dependable intrusion tolerance [8] is an adaptive architecture that provides an alert system for intrusions. The architecture consists of proxies and a monitoring system. In addition to the intrusion detection, a cluster is also capable of detecting anomalies based on an agreement protocol.

The intrusion tolerance by unpredictable adaptation (ITUA) project [9] defined a set of possible attacks and developed a middleware-based intrusion tolerance solution. ITUA identifies important features of attacks and uses them in the validation of defensive strategies and algorithms. For example, a set of security domains, such as a host and LAN can be used to describe the boundary of attacks. ITUA adds uncertainty to its intrusion tolerance technology and shows how unpredictability can be used to increase the intrusion tolerance of the system. ITUA has middleware that adaptively protects applications by using protocols for group communications and cryptography. ITUA consists of managers, replicas and subordinates located in various groups.

Scalable intrusion-tolerant architecture (SITAR) [10] consists of five middleware components to protect COTS servers. As shown in Fig. 2, SITAR contains the following three layers, which encapsulate and protect the overall system: proxy server, ballot monitor, and acceptance monitor. SITAR provides services to secure COTS servers from external attacks and has an advantage of managing tasks without modifications on servers. When SITAR detects an attack, it reconfigures the compromised servers.

Malicious and accidental fault tolerance for Internet applications (MAFTIA) [11] can detect an intrusion using incorporated IDS sensors. The architecture of MAFTIA is composed of several layers which provide a complete solution for ITS. It is built from local host machines or network devices. It also contains a middleware layer which provides services and algorithms for applications. An application must be implemented on top of the middleware platform using given application programming interfaces and protocols specified by MAFTIA.

Generally, middleware-based ITSs have been designed for distributed services. In recent years, there has been increasing demand for sustainability in distributed systems against cyber threats due to the importance of critical applications such as finance and healthcare services [12]. Also, protecting common Web servers is one of the most important applications for middleware-based ITS [13, 14].

C. Recovery-Based ITS

Most network infrastructures, such as firewalls [2], Web servers, and DNS servers, can be exposed to attackers for a long period of time. They give enough time for attackers to perform malicious actions. A recovery-based ITS called SCIT can be a solution to this problem.

The SCIT system uses a periodic recovery and simultaneously maintains service availability using redundant servers. It is composed of the SCIT controller and redundant servers, as depicted in Fig. 3. The SCIT controller is responsible for the state transitions of each server. The states are:

1) Active: Server is online, receives and processes the requests from clients.
2) Grace period: Server is online and does not receive any more requests. But, it processes requests that

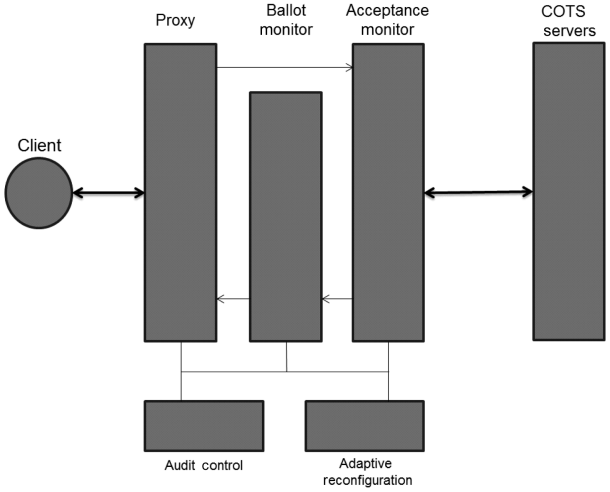

Fig. 2. The system architecture of scalable intrusion-tolerant architecture. Located between clients and servers, middleware components provide intrusion tolerant functionalities. COTS: commercial off-the-shelf.

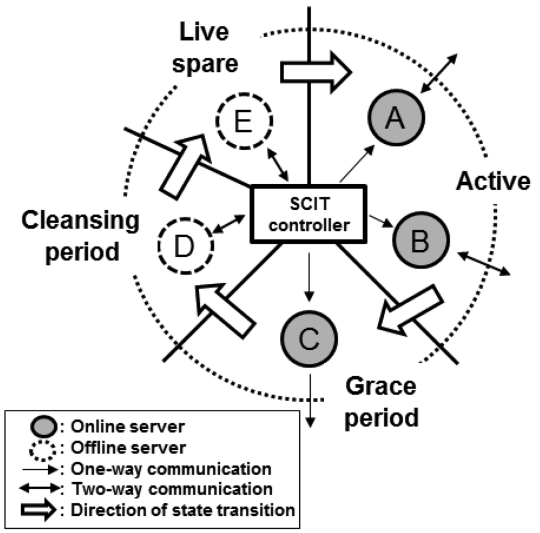

Fig. 3. Self-cleansing intrusion tolerance (SCIT) architecture and server states.

are received when it was in the active state.
3) Cleansing period: Server is offline and recovers the system configuration files, service files, and so on.
4) Live spare: Server is offline and waits to be active.

These four states constitute a continuous cycle: Active → Grace period → Cleansing period → Live spare → Active. The SCIT controller commands each server in order to change states through the internal network. When a server is online, the internal network from the SCIT controller to the server is a one-way communication. This prevents infected servers from performing malicious operations on the SCIT controller. This architecture can be easily constructed using simple hardware controllers, such as on-off switches [15].

To maintain service availability through periodic recoveries, SCIT requires enough redundant servers and a secure central controller. If there are more redundant servers than required servers to keep service available, the exposure time can be reduced, making the security of the system stronger. The SCIT controller schedules the states of each server depending on the control algorithm. A control algorithm was suggested to keep service available [16]. An administrator sets reliability, which is defined as the probability that the system runs without a failure. Then, the SCIT controller keeps the needed number of online servers and changes extra servers' states to perform recovery.

Virtualization has also been exploited in SCIT system design [17, 18]. Virtualization is a technique to create several virtual hardware platforms and OSs on one host. Therefore, it can be used to make enough virtual servers. With a snapshot feature, the SCIT system can recover the state of the virtual machine (VM), including the servers' recovery operation.

Designing an SCIT system that controls many firewalls is relatively easy [17]. A firewall just inspects incoming packets based on signatures and drops suspicious packets. So, each firewall does the same operation, and does not need to have a short-term (session) memory. When the offline firewall finishes recovering, the SCIT controller changes its state to Active and sets its IP address to that of the currently running firewall to be recovered.

Considering systems in which each server performs a different function and has a short-term (session) memory, the design of an SCIT system may not be so straightforward [18]. This is because when a running server's state is changed, the session information must be transferred to the other running servers. This problem can be solved by using a multicast protocol or an external server. To alleviate this problem, Terracotta has been used [18].

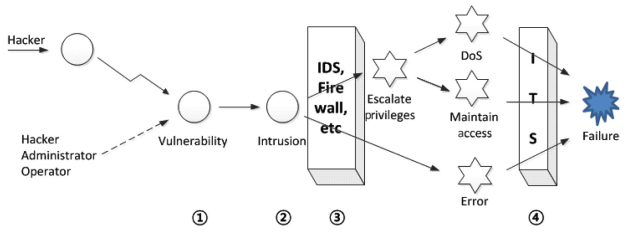

Fig. 4. The roles of intrusion-tolerant system (ITS). IDS: intrusion detection system, DoS: denial-of-service.

III. RECENT RESEARCH

A. Adaptive Cluster ITS

Lim et al. [19] added additional functions to the earlier virtualization-based ITS, based on the assumption that a successful intrusion can cause not only internal error but a denial-of-service (DoS) attack in order to achieve a security failure, as shown in Fig. 4.

The suggested system is called adaptive cluster transformation (ACT). In ACT, the central controller has more responsibilities in addition to the proactive rejuvenation in SCIT.

The main idea, adaptive cluster expansion and reduction scheme, involves the use of a changeable cluster size depending on the response delay of the current cluster instead of using a fixed cluster size. If an operator designates a certain level of performance, σ_{up} and σ_{down}, the threshold values for changing the cluster size, TH_{CL_up} and TH_{CL_down}, are decided based on ideal response delay values, I_{CL}, which are obtained from a direct process [14] as shown in Eqs. (1) and (2).

$$TH_{CL\_up} = \frac{100}{\sigma_{up}} \cdot I_{CL} \tag{1}$$

$$TH_{CL\_down} = \frac{I_{CL}}{I_{CL-2}} \cdot \frac{100}{\sigma_{down}} \cdot I_{CL-2} \tag{2}$$

This makes the system sensitive to performance degradation or enhancement in order to maintain specific performance levels. Thus, this scheme contributes to maintaining QoS enhancement.

A Byzantine fault-resilient system was designed in order to resist against a DoS attack even though all Byzantine faults cannot be detected. In the case of a large number of packets incoming for a short period of time, the system is able to predict this kind of inappropriate state in advance, and current active VMs supplying services can be substituted with pristine VMs. This is also decided by the threshold value related to the designated performance, as in Eq. (3).

$$TH_{CL\_fault} = I_{CL} \times x \tag{3}$$

The performance enhancements of ACT were verified by CSIM 20 simulation. Also, it was shown that ACT prevents the rapid increase of the average response delay in circumstances with incoming packets increasing rapidly.

Kim et al. [20] suggested a more improved system based on historical data. The main idea of this system is that if historical data is given through enough learning processes, then a faster transformation is possible.

The processing ratio is considered as an important variable representing both the current request volume and response time from the current cluster, as in Eq. (4).

$$Processing\ ratio = \frac{Processing\ capacity}{Request\ volume} \tag{4}$$

This variable is used to determine the cluster size and to calculate the performance degradation according to the number of waiting requests. The performance degradation is calculated using Eq. (5).

$$Rate_{deg} = \frac{\frac{RT}{RT_{ideal}} \cdot VM + \frac{Ratio_{ideal}}{Ratio_{new}} \cdot (VM_{new} - VM)}{Request\ volume} \tag{5}$$

The history map which includes the processing ratio, response time, cluster size, and utilization rate of each VM is obtainable through enough learning processes. If the history map is usable, then the central controller might change the cluster size in advance compared to ACT. Also, in the DoS attack prediction, it can use historical data. When the processing ratio and response time are observed in abnormal values, the system considers incoming packets as an attack. If the increasing rate of a request volume is higher than the designated DoS attack multiplier, then the incoming packets are considered as DoS packets, and the central controller expands the current cluster size as much as possible. All experiments showed performance enhancements with respect to both higher utilization of the computing resources and better resistance against a DoS attack.

The two adaptive ITS schemes might be applicable to design architectures for many systems which require a higher performance and a higher level of security.

B. Adaptive Response ITS

In this section, ITS studies which use an adaptive response technique are introduced. There are various types of attacks in the real world. For example, a Web server can be attacked by a DoS attack, stealth attack, and other unknown attacks. Since an ideal countermeasure that can prevent all kinds of attacks does not exist, adaptive-response ITS selects an appropriate countermeasure considering external and internal conditions.

Effective resource transformation (ERT) [21] determines how many VMs will process requests according to a circumstance. Because the total number of VMs belonging to the system is fixed, ERT decides the size of two groups: a cleansing group and a processing group.

VMs in the cleansing group are isolated from the external network and make its state pristine. VMs in the processing group handle requests from the external network and send responses. Naturally, if the size of the processing group increases, the processing speed will improve. On the other hand, if the exposure time of each VM in the processing group is extended, the system becomes more vulnerable to attacks. In contrast, a system with a small processing group is resistant to attacks, but its processing speed is low.

In ERT, the central controller determines the size of two groups according to the conditions. The size of the processing group is adjusted while taking into account the response to an incoming packet rate

$$N_{min} = \lceil R/P \rceil \tag{6}$$

where P is the processing capacity of one VM and R is the request volume. The number of required VMs can be expressed as N_{min}. If the incoming packet rate increases, N_{min} will grow to process requests within a reasonable amount of time. In contrast, N_{min} will decrease if the processing group is too large.

The simulation results show that the system applying ERT processes requests with acceptable speed. Furthermore, it shows that ERT decreases data leakage compared to using a processing group with fixed size.

Heo et al. [22] proposed an adaptive recovery scheme (ARS) to improve the performance and security of ITS. Fig. 5 shows the architecture of ITS based on ARS.

The states of VMs are controlled by the central controller. A VM's state is periodically switched to active → cleansing → ready → active. VMs in the active state are exposed to the external network and process requests. VMs in the cleansing state are offline and restored. VMs in the ready state are completely clean and ready to provide service.

When a VM changes its state from ready to active, a snapshot of its critical sections in the VM is taken and saved in the central controller. VMs in the active state

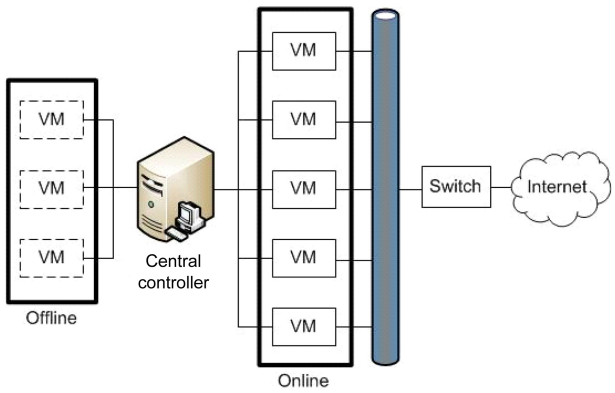

Fig. 5. Architecture of intrusion-tolerant system based on adaptive recovery scheme. VM: virtual machine.

send their snapshot to the central controller to check the integrity.

ARS utilizes a proactive recovery and a reactive recovery. The central controller recovers VMs in the active state by a predefined time to perform a proactive recovery. There are three different reactive recovery schemes in ARS according to the circumstances. First, if the central controller detects an attack on a VM and there are VMs in the ready state, the central controller will replace the attacked VM with the ready state VM and restore the attacked VM. If there is not any VM in the ready state, the central controller will send a snapshot to the attacked VM. The VM can use it to recover itself. Lastly, if a self-recovered VM is attacked again, the central controller will change its state to cleansing and increase the exposure time of active VMs that have less exposure time than the attacked VM in order to reduce the performance degradation.

Experimental results show that ARS can significantly reduce the negative effects of attacks that affect performance and availability. ARS protects the integrity of critical files by a snapshot. The snapshot comparison detects most attacks whose purpose is compromising the integrity of files. Additionally, a snapshot can be used to repair damaged files.

C. Policy-Based ITS

With sophisticated attacks, there are many defendable solutions, such as the detection approach using neural networks [1]. Due to the limitations of the detection approach, a new paradigm is needed. One representative virtualization-based intrusion tolerant scheme is SCIT [17]. However, the repetitive exposure of vulnerabilities of each VM and VM rotation pattern might give chances to compromise a system to attackers. Therefore the suggested scheme, which quantifies the degree of vulnerabilities and applies it to the next rotation, can enable the VM rotation pattern to be hidden and reduce the data leakage of the system.

ITS architectures based on SCIT using virtualization technology have several weaknesses. First, since they do not consider a contaminated VM, malicious attackers can easily exploit the vulnerabilities of each VM. The VM rotation pattern can also be known to attackers if the system does not modify the pattern while supplying services. As a result, information leakage can occur due to the repetitive exposure of vulnerabilities and the VM rotation pattern.

Kim et al. [23] proposed a novel method that quantifies the degree of vulnerability by inspecting file integrity and detecting malicious codes before the cleansing period. In order to select active VMs for the next rotation, the central controller obtains the calculated degree of vulnerability. These inspecting and detecting processes are able to hide the VM rotation pattern and decrease data leakage compared to SCIT, which only considers the random selection of VM.

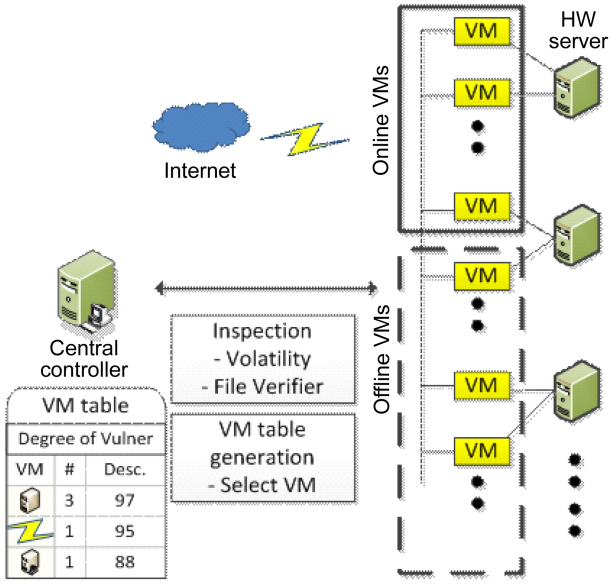

Fig. 6. Proposed system architecture. VM: virtual machine, HW: hardware.

Table 1. The initial time of attack

Exposure time (sec)	Initial attack (sec)
300	273.8
400	299.8
450	322.8
500	349.8
600	441.2
650	446.5

The proposed system architecture is illustrated in Fig. 6. This system consists of hardware servers with several associated VMs, the online VM group supplying services, and the offline VM group preparing for the online state. It is assumed that all VMs have the same function and support the same Web services to clients via the Internet, even though they have different operating systems and applications.

A VM enters into the offline VM group after providing services during a certain period. When a VM is in the cleansing state, the integrity of crucial files and memory should be checked by the central controller. The central controller generates the VM table according to the results of the analysis, and stores it into a secured database. After that, the central controller selects the next online VMs which are ready to enter into the online VM group.

The degree of information leakage can be acquired by figuring out the area at a certain exposure time on the

information loss curve map [17, 24]. The time when the initial attack occurs is acquired from 30 simulations, as shown in Table 1. This time value varies according to the exposure time, and the intruder residence time values at each exposure time can be calculated by subtracting the initial attack time from the exposure time. Thus, this result means that the exposure time is shorter and that there is a less probability to be attacked by malicious adversaries [23].

It is assumed that a repeated pattern of rotations and vulnerabilities of each VM can be a target to attackers [23]. The suggested scheme quantifies the vulnerability of each VM and applies the result of quantifying to select the online VM for the next rotation. It was verified that this scheme can reduce the data leakage rate compared to SCIT. An alternative ITS which hides the pattern of rotation and reduces the exposure of vulnerabilities of the system was suggested.

IV. CONCLUSION

Most current information systems are connected to the Internet, and it is hard to successfully protect such systems against all threats. In this context, ITS has received significant attention as one of the most important service principles for providing stable services.

In this paper, previous approaches to providing sustainable services have been categorized and analyzed. The most important role of ITS is to provide satisfactory services to clients in threatening environments rather than perfectly protecting the system.

Recent novel ITSs have also been introduced. It was verified that they can provide improved quality of services and enhanced security. With the advent of advanced threats, in order to provide better quality of service with secure communication over open networks, studies on ITS have been steadily conducted in both industry and academia.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government (No. 2011-0016584).

REFERENCES

1. A. H. Fares, M. I. Sharawy, and H. H. Zayed, "Intrusion detection: supervised machine learning," *Journal of Computing Science and Engineering*, vol. 5, no. 4, pp. 305-313, 2011.
2. W. Park and C. Park, "Data firewall: a TPM-based security framework for protecting data in thick client mobile environment," *Journal of Computing Science and Engineering*, vol. 5, no. 4, pp. 331-337, 2011.
3. J. Reynolds, J. Just, E. Lawson, L. Clough, R. Maglich, and K. Levitt, "The design and implementation of an intrusion tolerant system," in *Proceedings of International Conference on Dependable Systems and Networks*, Washington, DC, 2002, pp. 258-290.
4. J. Chong, P. Pal, M. Atigetchi, P. Rubel, and F. Webber, "Survivability architecture of a mission critical system: the DPASA example," in *Proceedings of the 21st Annual Computer Security Applications Conference*, Tucson, AZ, 2005, pp. 495-504.
5. P. Pal, F. Webber, and R. Schantz, "The DPASA survivable JBI: a high-water mark in intrusion-tolerant systems," in *Proceedings of the 1st Workshop Recent Advances on Intrusion-Tolerant Systems*, Lisbon, Portugal, 2007.
6. D. Arsenault, A. Sood, and Y. Huang, "Secure, resilient computing clusters: self-cleansing intrusion tolerance with hardware enforced security (SCIT/HES)," in *Proceedings of the 2nd International Conference on Availability, Reliability and Security*, Vienna, Austria, 2007, pp. 343-350.
7. J. Knight, D. Heimbigner, A. L. Wolf, A. Carzaniga, J. Hill, P. Devanbu, and M. Gertz, "The Willow architecture: comprehensive survivability for large-scale distributed applications," Department of Computer Science, University of Colorado at Boulder, Boulder, CO, Technical report, 2001.
8. A. Valdes, M. Almgren, S. Cheung, Y. Deswarte, B. Dutertre, J. Levy, H. Saidi, V. Stavridou, and T. E. Uribe, "An architecture for adaptive intrusion-tolerant server," in *Proceedings of Security Protocols Workshop*, Cambridge, UK, 2002, pp. 158-178.
9. M. Cukier, J. Lyons, P. Pandey, H. V. Ramasamy, W. H. Sanders, P. Pal, F. Webber, R. Schantz, J. Loyall, R. Watro, M. Atighetchi, and J. Gossett, "Intrusion tolerance approaches in ITUA," in *Supplement of 2001 International Conference on Dependable Systems and Networks*, Goteborg, Sweden, 2001.
10. F. Wang, F. Gong, C. Sargor, K. Goseva-Popstojanova, K. S. Trivedi, and F. Jou, "SITAR: a scalable intrusion tolerance architecture for distributed services," in *Proceedings of the Second IEEE/SMC Information Assurance Workshop*, West Point, NY, 2001, pp. 38-45.
11. P. E. Verissimo, N. F. Neves, C. Cachin, J. Poritz, D. Powell, Y. Deswarte, R. Stroud, and I. Welch, "Intrusion-tolerant middleware: the road to automatic security," *IEEE Security and Privacy*, vol. 4, no. 4, pp. 54-62, 2006.
12. K. C. Chen, "Machine-to-machine communications for healthcare," *Journal of Computing Science and Engineering*, vol. 6, no. 2, pp. 119-126, 2012.
13. S. B. E. Raj and G. Varghese, "Analysis of intrusion-tolerant architectures for Web servers," in *Proceedings of the International Conference on Emerging Trends in Electrical and Computer Technology*, Tamil Nadu, India, 2011, pp. 998-1003.
14. A. Saidane, V. Nicomette, and Y. Deswarte, "The design of a generic intrusion-tolerant architecture for Web servers," *IEEE Transactions on Dependable and Secure Computing*, vol. 6, no. 1, pp. 45-58, 2008.
15. Y. Huang, D. Arsenault, and A. Sood, "Incorruptible system self-cleansing for intrusion tolerance," in *Proceedings of the 25th IEEE International Performance, Computing, and*

Communications Conference, Phoenix, AZ, 2006, pp. 493-496.
16. Q. Nguyen and A. Sood, "Realizing S-reliability for services via recovery-driven intrusion tolerance mechanism," in *Proceedings of IEEE International Conference on Dependable Systems and Networks Workshops*, Chicago, IL, 2010, pp. 176-181.
17. Y. Huang and A. Sood, "Self-cleansing systems for intrusion containment," in *Workshop on Self-Healing, Adaptive, and Self-Managed Systems (SHAMAN)*, New York, NY, 2002.
18. A. K. Bangalore and A. Sood, "Securing Web servers using self-cleansing intrusion tolerance (SCIT)," in *Proceedings of the 2nd International Conference on Dependability*, Athens, Greece, 2009, pp. 60-65.
19. J. Lim, Y. Kim, D. Koo, S. Lee, S. Doo, and H. Yoon, "A novel adaptive cluster transformation (ACT)-based intrusion tolerant system for hybrid information technology," *Journal of Supercomputing*, 2013. http://dx.doi.org/10.1007/s11227-013-0928-5.
20. Y. Kim, J. Lim, S. Doo, and H. Yoon, "The design of adaptive intrusion tolerant system (ITS) based on historical data," in *Proceedings of the 7th International Conference for Internet Technology and Secured Transactions*, London, UK, 2012, pp. 662-667.
21. S. Heo, Y. Kim, J. Lim, and H. Yoon, "A design of a novel intrusion tolerant system through effective resource transformation," *Telecommunications Review*, vol. 22, no. 6, pp. 913-921, 2012.
22. S. Heo, J. Lim, M. Lee, S. Lee, and H. Yoon, "A novel intrusion tolerant system based on adaptive recovery scheme (ARS)," in *IT Convergence and Security 2012*, Heidelberg, Germany: Springer, 2013, pp. 71-78.
23. H. Kim, J. Lim, and H. Yoon, "A design of a novel intrusion tolerant system using virtual machine image analysis and secure exposure policy," *Telecommunications Review*, vol. 22, no. 6, pp. 904-912, 2012.
24. D. Pham and A. Sood, "An intrusion tolerance approach to enhance single sign on server protection," in *Proceedings of the 3rd International Conference on Dependability*, Venice, Italy, 2010, pp. 98-103.

Seondong Heo

Seondong Heo received his B.S and M.S. degrees in Computer Science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2009 and 2011, respectively. He is currently working towards Ph.D. degree in Computer Science at KAIST. His research interests include botnet, intrusion detection system, and intrusion tolerant system.

Pyeong Kim

Pyeong Kim received his B.S and M.S. degrees in Computer Science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2007 and 2009, respectively. He is currently working towards Ph.D. degree in Computer Science at KAIST. His research interests include cryptography, encryption and authentication.

Yongjoo Shin

Yongjoo Shin received the B.S. degree in Weapon Engineering from the Korea Military Academy (KMA), Seoul, Korea, in 2000 and the M.S. degree in Information Resource Management from Air force Institute of Technology (AFIT), Ohio, USA, in 2006. He is currently working toward Ph.D. degree from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea. His current research interests are in the mobile security including encryption algorithm, solutions against the denial of service (DoS), and the design and analysis of intrusion tolerance system.

Jungmin Lim

Jungmin Lim received a master degree in electronic engineering and engineering from Seoul National University, Korea, in 2005. He is currently working toward Ph.D. degree in computer science from Korea Advanced Institute of Science and Technology, Korea. His main researches include the system security, especially intrusion tolerant system based on adaptive cluster transformation and historical data.

Dongyoung Koo

Dongyoung Koo received the B.S. degree in computer science from Yonsei University in 2009, and the M.S. degree in computer science from KAIST in 2012. His research interests include information security, secure cloud computing, and cryptography.

Yonggon Kim

Yonggon Kim received his B.S. and M.S. degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea. He is currently working towards Ph.D. degree in computer science at KAIST. His research interests include lattice-based cryptography, public-key encryption and mobile-based authentication.

Ohmin Kwon

Ohmin Kwon received the B.S degree in the Division of Computer and Communication Engineering from Korea University, Seoul, Korea, in 2012. He is currently working toward the M.S degree in the Department of Computer Science, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea. His research interests include applied cryptography, security and privacy in cloud computing

Hyunsoo Yoon

Hyunsoo Yoon received the B.S. degree in electronics engineering from Seoul National University, Seoul, Korea, in 1979, the M.S. degree in computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1981, and the Ph.D. degree in computer and information science from The Ohio State University, Columbus, OH, in 1988. He is currently a Professor of the Department of Computer Science, KAIST. During 1978-1980, he was with the Tongyang Broadcasting Company, Korea, then Samsung Electronics Company, Seoul, Korea, during 1980-1984. From 1988 to 1989, he was a Member of the Technical Staff with AT&T Bell Labs, Indian Hill, IL. Since 1989, he has been a Professor of the Department of Computer Science, Korea Advanced Institute of Science and Technology. His research interests include mobile ad hoc networks, wireless networks, and network security.