Symmetry **2014**, *6*, 938-953; doi:10.3390/sym6040938

OPEN ACCESS

symmetry

ISSN 2073-8994

www.mdpi.com/journal/symmetry

Article

Design of a Secure System Considering Quality of Service

Seondong Heo [1,*], Soojin Lee [2,†], Seokjoo Doo [3,†] and Hyunsoo Yoon [1,†]

[1] Department of Computer Science, Korea Advanced Institute of Science and Technology, 291 Daehak-ro, Yuseong-gu, Daejeon 305-701, Korea; E-Mail: hyoon@nslab.kaist.ac.kr

[2] Department of Computer Science and Engineering, Korea National Defense University, 33 Je2Jayuro, Goyang-si, Gyeonggi-do 412-706, Korea; E-Mail: cyberkma@gmail.com

[3] Department of Electronic Engineering, Korea Army Academy at Yeong-cheon, 495 Hogukro, Gogyeong-myeon, Yeongcheon-si, Gyeongbuk 770-849, Korea; E-Mail: doos.kr@gmail.com

† These authors contributed equally to this work.

* Author to whom correspondence should be addressed; E-Mail: sdheo@nslab.kaist.ac.kr; Tel.: +82-42-350-7829 (ext. 7829).

External Editor: Young-Sik Jeong

Received: 14 August 2014; in revised form: 5 November 2014 / Accepted: 7 November 2014 / Published: 13 November 2014

Abstract: Improvements in networking technologies have provided users with useful information services. Such information services may bring convenience and efficiency, but might be accompanied by vulnerabilities to a variety of attacks. Therefore, a variety of research to enhance the security of the systems and get the services at the same time has been carried out. Especially, research on intrusion-tolerant systems (ITSs) has been conducted in order to survive against every intrusion, rather than to detect and prevent them. In this paper, an ITS based on effective resource conversion (ERC) is presented to achieve the goal of intrusion-tolerance. Instead of using the fixed number of virtual machines (VMs) to process requests and recover as in conventional approaches, the ITS based on ERC can transform the assigned resources depending on the system status. This scheme is proved to maintain a certain level of quality of service (QoS) and quality of security service (QoSS) in threatening environments. The performance of ERC is compared with previous studies on ITS by CSIM 20, and it is verified that the proposed scheme is more effective in retaining a specific level of QoS and QoSS.

Keywords: security; quality of service; quality of security service; intrusion-tolerant system; resource conversion; virtual machine

1. Introduction

Most current information systems are connected to the Internet for efficiency and convenience. However, the growth of accessibility makes the systems vulnerable to attackers. For this reason, many studies have been conducted in order to improve security of information systems [1–3].

Studies on cryptography [4] are aimed to provide confidentiality, integrity and non-repudiation. Since there are rapid growth of commerce and communication via Internet, cryptography is essential for the current service systems. However, though it improves security of information systems and communication, it still cannot prevent malicious attacks on the systems. Therefore research on protection techniques is necessary to defend the systems from various cyber-attacks.

Traditional protection techniques, such as intrusion detection systems (IDSs), intrusion prevention systems (IPSs), and firewalls have been designed to protect against a variety of attacks. Sometimes these schemes achieve excellent detection by analyzing previous attacks. However, it is impossible to prevent all kinds of attacks with current security solutions. They cannot detect new and unreported attacks because their methods are based on attack signatures previously recognized. Anomaly-based techniques, which apply heuristics to detect intrusions, can prevent previously unknown attacks, but they have a false positive rate and can be fooled by other attacks. For this aspect, Intrusion-Tolerant Systems (ITSs) were proposed to provide service reliability and survivability in the face of attacks.

Many studies [5–9] on ITSs have been conducted, and they have been done based on two principles in general: redundancy and diversity. Redundancy is a mandatory mechanism to avoid a single-point-of-failure problem. However, homogeneous redundancy still preserves common fundamental vulnerabilities that can be exploited by attackers. Diversity can overcome this limitation by implementing functions in different ways. Various kinds of architectures, such as hardware [6], middleware [5] and hybrid [7] architectures were presented until now. However, the common problem of these solutions [5–7] is that after a certain period of time, a malicious intruder is able to find and exploit vulnerabilities to collapse the whole system.

Recently, some research [8,9] addressed this problem by using virtual machines (VMs) and proactive recovery. Proactive recovery is a technique that resumes VMs pristine state periodically. Cleansing procedures remove the effects of undetected intrusions occurred during exposure to an external network. If cleansing is performed with a sufficiently short exposure time, attackers are unable to compromise enough VMs to break the system [8]. On the other hand, a short exposure time causes degradation in QoS, because resources of the system are utilized to cleanse VMs rather than to process the requests. In other words, there is a trade-off between QoSS and QoS [10].

In this paper, we introduce an ITS based on effective resource conversion (ERC), which mainly focuses on maintaining a specific level of both QoSS and QoS. ERC includes two main schemes: QoS & QoSS scoring and resource conversion.

First, in order to calculate the ideal exposure time according to system environments, we propose the QoS & QoSS scoring scheme. To achieve the goal, we measure QoS in terms of response time, and predict the amount of damage caused by attacks to evaluate security performance.

Second, the resource conversion scheme assigns the exposure time computed by the previous scoring scheme. This enables the system to be sensitive to internal and external factors, such as a response time and an incoming packet rate, which affect the performance of the system. It allows the system to control the capability of maintaining QoS and QoSS in the end.

In order to show how ERC can be used to retain QoS and QoSS of a system and evaluate the effectiveness of our approach, we implemented the scheme by using CSIM20 [11] and compared it with previous work [8,12]

The contributions of our research are as follows:

- We introduce a new scoring scheme that can consider QoS and QoSS simultaneously.
- We establish the concept of ERC that can be used in any intrusion-tolerant architecture.
- We evaluate the performance of the system in various situations through simulation.

The rest of the paper is organized as follows. Section 2 is about the related work on a recovery-based ITS. In Section 3, the scoring scheme of ERC is proposed. In detail, Section 4 describes the architecture of the ITS based on ERC, and Section 5 shows the experimental design and results. The paper ends with a conclusion in Section 6.

2. Related Work

Designing protections and adaptation into a survivability architecture (DPASA) [6] has various layers to manage system reliability and includes approaches of intruders. The architecture integrates defense principles, such as redundancy, diversity, detection, correlation, and adaptive response. Also, cyber-defense mechanisms can be utilized to improve the survivability of the architecture. Even though DPASA has high resiliency against intrusions, it requires expert operator to effective response to attacks.

Scalable intrusion-tolerant architecture (SITAR) [5] consists of five middleware components to defend commercial off-the-shelf (COTS) servers from intrusions. SITAR contains three components that protect the overall system: proxy server, ballot monitor and acceptance monitor. SITAR manages tasks without modification on COTS servers. When audit control module detects attacks, adaptive reconfiguration module reconfigures the overall system to support the desired security level.

A generic architecture for intrusion tolerant Web servers [7] is based on a redundant cluster of diversified Web servers and proxies. The Web servers in the architecture are running different application and operation system on diverse hardware platforms. Proxies have an essential function in the architecture: monitoring the state of the Web servers, selecting the regime according to the alert level and protecting the database. Also, the architecture uses IDS, which consists of multiple detection mechanisms in the architecture.

Self-cleansing intrusion tolerance (SCIT) [8] is a representative study that deploys proactive recovery. The SCIT system is composed of the central controller and redundant servers, which are

embodied by VMs as depicted in Figure 1. Applications and services are deployed in VMs. The central controller is responsible for the state transitions of each server. The states are:

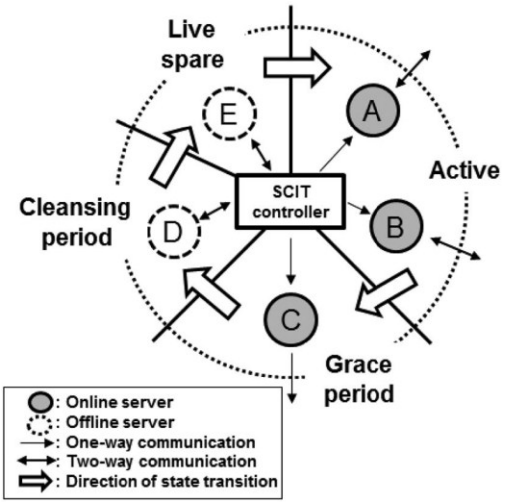
(1) Active: Server is online, receives and processes the request from clients.

(2) Grace period: Server is online and does not receive any more requests from the clients. It processes requests that 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.

Figure 1. Self-cleansing intrusion tolerance (SCIT) architecture and the server states.



If there are more resources than required resources to keep service available, the exposure time can be reduced, to enhance the security of the system. The central controller schedules the states of each VM depending on the control algorithm, which determines the exposure time on the basis of internal and external situations. The main problem is that the control algorithm only considers factors related to the security. An adversary can disturb the processing requests from the legitimate clients by sending a large amount of traffic to the system.

To mitigate this problem, adaptive cluster transformation (ACT) [12] added additional functions, which are adaptive cluster expansion and reduction schemes, involving the use of a variable cluster size depending on the response time instead of using a fixed cluster size. If a system administrator 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}, as shown in equations:

$$TH_{CL\_down} = \frac{100}{\sigma_{down}} \cdot I_{CL} \quad (1)$$

$$TH_{CL\_up} = \frac{100}{\sigma_{up}} \cdot I_{CL} \quad (2)$$

This makes the system sensitive to the performance in order to retain specific performance levels. On the other hand, ACT did not take account of the exposure time. It is limited to control the exposure

time according to attack rates. In addition, physical resources, which are required to expand a cluster size, were not considered proper.

3. The QoS and QoSS Scoring Schemes

In ERC, the central controller determines the number of VMs, which will process requests according to circumstances. Because the total number of VMs belonging to the whole system is constrained by physical resources, ERC 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 process requests and send responses to the clients. If the number of VMs in the processing group increases, the processing speed will be improved. However, a huge processing group can cause the extension of the exposure time of each VM and consequently makes the system become more vulnerable to attacks. In contrast, the system with a small processing group is impervious to attack, but its QoS is low.

In this section, we will present a scheme that allows the central controller to decide the appropriate exposure time. This scheme enables the system to retain a specific level of QoS and QoSS. In Sections 3.1 and 3.2, we explain the methods of the calculation on QoS and QoSS, respectively.

3.1. The QoS Scoring

QoS of a system is considerably affected by the size of the processing group. Since unnecessarily large processing groups may impair the security of the system, it is crucial issue to determine the adequate processing group size. Table 1 shows the parameters associated with ERC.

Table 1. List of Parameters.

Parameter	Description
T_e	Time duration in which a VM is exposed to the external network
T_c	Time duration in which a VM goes through cleansing
N_{max}	Maximum number of nodes that can be operated with the given resources
N_p	Size of the processing group
R	The number of incoming requests per unit time
P	Processing capacity of a VM
λ	Poisson parameter which is the number of attacks per unit time
D_a	The number of compromised records caused by attack per unit time

In ERC, the score of QoS (S_q) is determined by the ratio of the processing capacity to the incoming request. The processing capacity of the system is the product of N_p and P, so S_q can be expressed as:

$$S_q = \frac{P \cdot N_p}{R} \tag{3}$$

The central controller determines the size of processing group according to the exposure time as:

$$N_p = \frac{T_e \cdot N_{max}}{T_c + T_e} \tag{4}$$

Therefore S_q is expressed as:

$$S_q = \frac{P \cdot T_e \cdot N_{max}}{R \cdot (T_c + T_e)} \quad (5)$$

If S_q exceeds one, it means that the total processing capacity of the system exceeds the number of incoming requests per unit time. In this condition, the system can process all the requests from the external network in acceptable response time. Therefore we assign one to S_q in this case. Modifying the exposure time is the only way to control the S_q, since P, T_c and N_{max} are restricted by the given hardware configuration and R is uncontrollable.

3.2. The QoSS Scoring

The measurement of QoSS is more complicated. With the assumption that the attack arrivals are independent and follow a Poisson distribution with parameter λ, the probability of n attacks per unit time is calculated as:

$$P(X = n) = \frac{\lambda^n e^{-\lambda}}{n!} \quad (6)$$

As shown in Figure 2, the attacker who intrudes the VM1 and VM2 will perform malicious activities to collapse the system. Since each VM will be offline periodically and cleansed to remove attacks, the residence time of the attacker is limited by T_c. The damage to VM1, which can be caused by the attacker, is much smaller than to VM2, because T_c of VM1 is half of T_c of VM2 in this example. Therefore, assigning the appropriate exposure time according to λ is essential to improve the security performance of the system.

In ERC, the score of QoSS (S_s) is measured by the number of compromised records, which means the amount of potential damage that could cause to the system. On average, the first attack occurs after 1/λ unit time, since the expected value of a Poisson distribution is λ. If T_e is larger than 1/λ, the attacker who intrudes into the system has a chance to compromise the system for the remaining exposure time ($T_e - 1/\lambda$). The expected damage of the VM can be obtained from the product of D_a and intruder residence time. Therefore, if an operator of the system desires a certain level of reliability (L_R), the expected damage of the VM (D_t) can be expressed as:

$$D_t = \left(T_e + \frac{\ln L_R}{\lambda}\right) \cdot D_a \quad (7)$$

ERC adjusts S_s as:

$$S_s = \frac{D_{max} - D_t}{D_{max}} \quad (8)$$

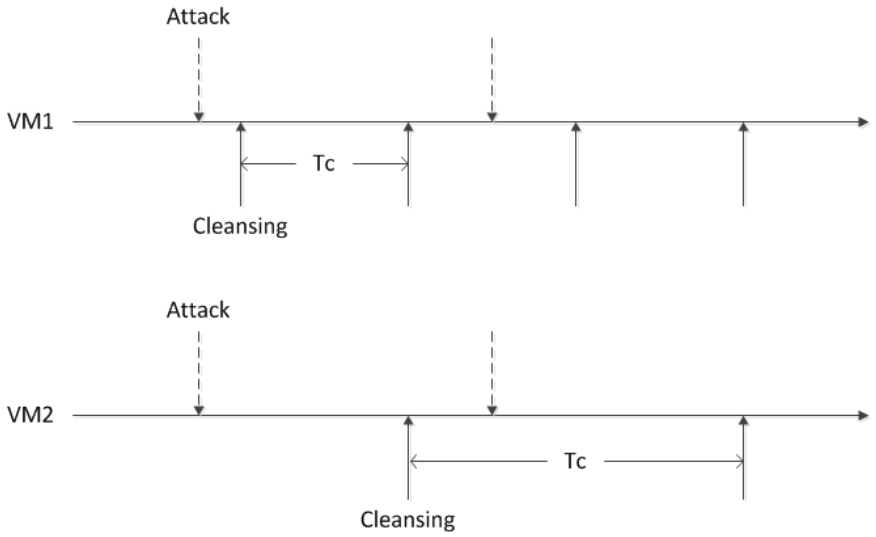
where D_{max} is the maximum amount of damage that can be caused by one attack. Therefore the score of QoSS with T_e can be expressed as:

$$S_s = \frac{D_{max} - \left(T_e + \frac{\ln L_R}{\lambda}\right) \cdot D_a}{D_{max}} \quad (9)$$

D_{max} and D_a are adjusted by referring to the reports of recent intrusions, and λ can be estimated by a monitor component. If S_s exceeds one, it implies that attackers do not have enough time to attack VMs

in the system. In this case, one is assigned to S_s. Similar to the QoS, the exposure time is the only available value which is adjustable by a system operator. Therefore assigning adequate values to the exposure time is necessary to make the system sustainable and resilient. This will be discussed in the next section.

Figure 2. Relation between QoSS and T_c.



3.3. Determining the Exposure Time

The total score of the system can be obtained from the formula as follows:

$$\alpha \cdot S_q + (1-\alpha) \cdot S_s = \alpha \cdot \frac{P \cdot T_e \cdot N_{max}}{R \cdot (T_c + T_e)} + (1-\alpha) \cdot \frac{D_{max} - \left(T_e + \frac{\ln L_R}{\lambda}\right) \cdot D_a}{D_{max}} \quad (10)$$

An administrator can control the sensitivity of the system by modifying α (where 0 < α < one). If a specific level of QoS is mandatory, the system operator has to select a high value of α for the system to meet the amount of requests. In contrast, if the system must survive in the face of intrusions, the system administrator assigns a small value to α to make the system more survivable. As we discussed in previous sections, modifying T_e is the only method to control QoS and QoSS. Therefore we can determine the prime exposure time, which makes the above formula maximum.

4. ERC Model

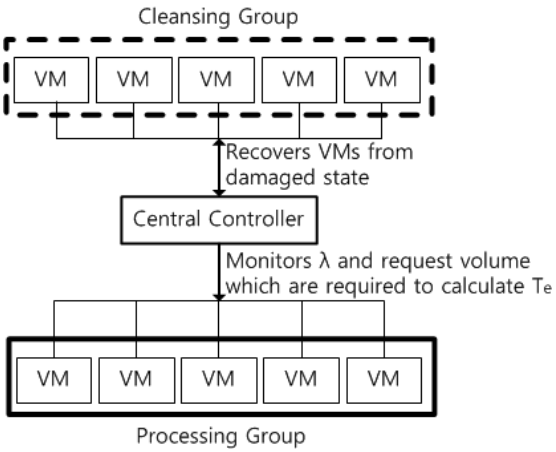
In this section, we present the architecture of the ITS based on ERC, and explain how ERC can be utilized to make a secure system. Our prototype integrates ERC and Web servers, which deliver contents that can be accessed through the Internet.

4.1. ITS Architecture Based on ERC

The proposed system is illustrated in Figure 3. We adopt a proactive recovery technique to protect the system. Each VM periodically rotates its state, active → cleansing → ready → active, sequentially. The central controller is responsible for the states of each VM. The central controller utilizes the ERC

scoring algorithm to respond to the number of requests per time and external threats. All services are supplied by VMs in the processing group, and all the requests are randomly delivered to the active VMs. Since every VM has the same functionality, they provide the same services to the clients. Following assumptions are applied to the design of the architecture.

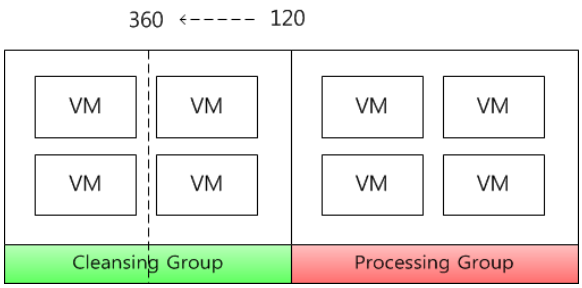
Figure 3. Architecture of the proposed ITS based on ERC.



- Since the internal network is separated from the external network, it is impossible to corrupt the central controller with attacks from the external network.
- All states of VMs are controlled by the central controller. In other words, the central controller can change any VM's state by regulating the exposure time.
- Attacks can be performed on active VMs, and a successful attack compromises the records of a certain VM. But any other malicious behaviors to disturb the operation of the system are not considered in order to simplify the evaluation.

In Figure 4, N_{max} is eight and T_c is 120 s. Therefore the exposure time is 120 s. In this configuration, the numbers of VMs in each group are the same. However, if the system receives a large amount of requests, the central controller assigns the new exposure time, which can be obtained from the ERC scoring algorithm to improve QoS. For instance, if T_e is changed to 360 s, the size of the processing group N_p will be increased to six. In this case, resources for proactive recovery are converted into resources for processing requests. As a result, the resource conversion improves QoS of the system by extending T_e. Naturally, the extension of T_e may give more chance to compromise the system for attackers. However, since our scoring scheme considers performance and security simultaneously, the ITS based on ERC can retain a certain level of QoS and QoSS.

Figure 4. Resource Conversion.

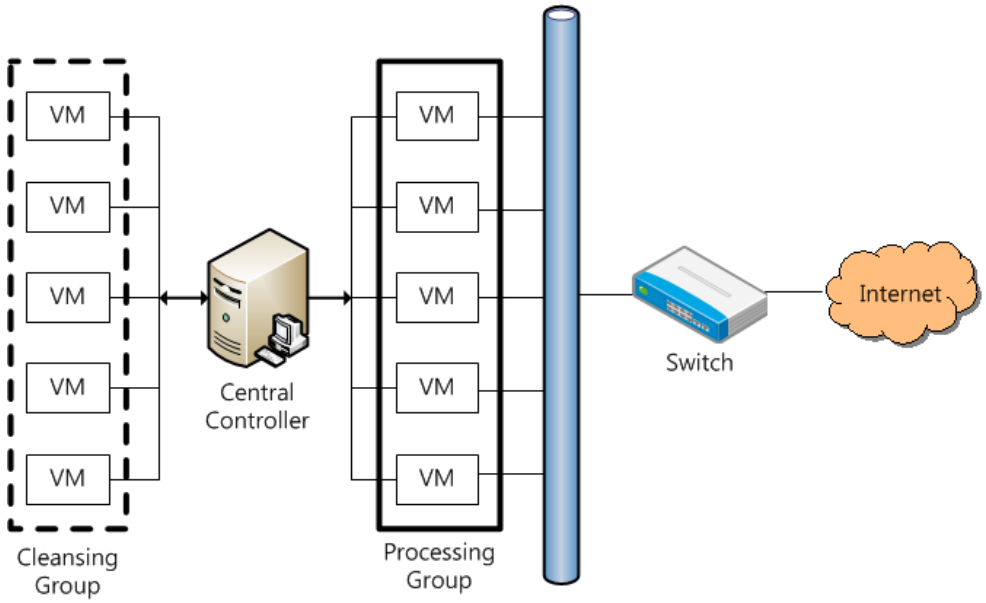


4.2. Prototype

In recent years, there has been increasing demand for survivability in Web servers against cyber-attacks due to importance of services, such as finance and social network. Protecting Web servers is one of the most primary applications for ITSs. Our prototype is depicted as Figure 5.

The direction of the arrows indicates the flow of communication. VMs in the cleansing group can communicate with the central controller to restore, whereas VMs in the processing group cannot send messages to the central controller, because they are exposed to the Internet. In prototype, the central controller not only changes the states of VMs as SCIT, but also assigns the exposure time to each VM. Therefore a separation of the internal and external networks is one of the most important techniques to protect the central controller.

Figure 5. Web server integrated with ERC.



The central controller requires the number of incoming requests per unit time (R) and the Poisson parameter λ. R can be observed by the switch, and we can obtain λ from VM's intrusion report and inspect file integrity and detect malicious activity at cleansing period [13].

Conventional studies on diversification, such as OS and applications [14], can be integrated with the prototype. However we do not consider diversification because this paper focused on only the improvements of intrusion tolerance by applying ERC.

5. Experimental Design and Results

The CSIM 20 simulator is used for estimating efficiency of ERC. CSIM is a simulation toolkit, which involves simulation engine and comprehensive tools that can be used to implement realistic models of complex systems. By using CSIM 20 simulator, we simulated ERC, ACT and SCIT under three different conditions for evaluating performance with C++ language in Visual studio 2008.

To show comparative performance, we use the reference values from the related work [12,15] as shown in Table 2. These values are commonly used throughout the experiments. Experimental environments are determined as follows:

- Probability functions that are supported in CSIM 20 are used to provide the various request generation and processing time.
- The request generation time and processing time follow an exponential distribution.
- Every VM exposed to the external network is vulnerable to attacks. The number of attacks follows a Poisson distribution with λ.
- Attacks compromise the records of the corrupted VM. The number of compromised records depends on D_a and the remaining exposure time.
- Cleansing procedures eliminates the effects of attacks from the corrupted VM.
- The total number of compromised records is equal to the total damage of the system.

Table 2. Reference values.

Contents	Value
T_c	120 s
N_{max}	10
Average processing time	0.01 s
D_a	0.31/s

5.1. Reducing the Request Arrival Time

In this experiment, we generate requests at intervals of 0.002 seconds and set λ to 0.00048 in order to create the environment similar to the previous work. In order to evaluate QoS performance under busy condition, we reduce the request arrival time gradually. We set α to 0.7 to make ERC more sensitive to QoS.

Figure 6 shows the response time of the system based on ERC, ACT and SCIT. ACT transforms the cluster size by periodically checking the whole system's average response time. The strategy of ACT is well adapted to the condition, and the QoS performance of the system based on ACT is better than ERC and SCIT. However, the number of compromised records is 10 times larger than the other schemes as shown in Table 3, as the cluster expansion causes extension of the exposure time.

On the other hand, ERC assigned the appropriate exposure time by the scoring scheme. Since α is set 0.7, the scoring scheme gives weight to QoS performance. Although the response time of the system based on ERC is slightly larger than ACT, it maintains the average response time less than 100 ms. Additionally, the performance of QoSS is similar to that of SCIT, which only considers security. Since SCIT does not consider QoS, the architecture based on SCIT cannot respond to the increase of the incoming request rate.

Figure 6. Response time with reduced request arrival time.

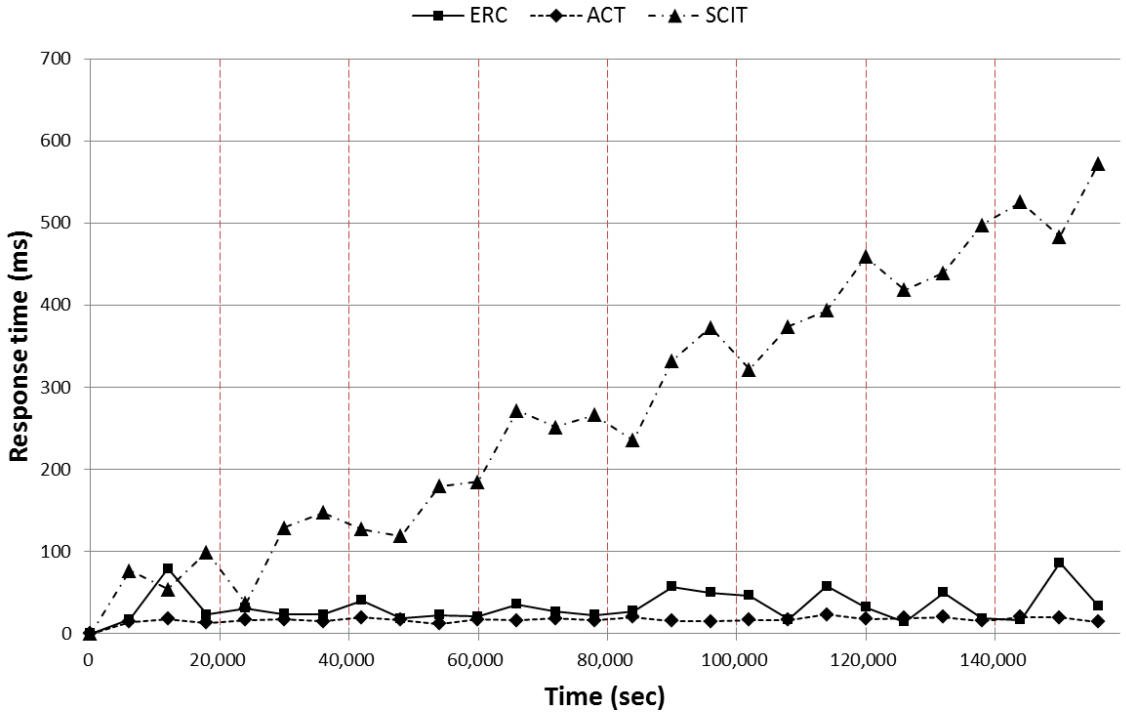


Table 3. The number of compromised records with reduced request arrival time.

Scheme	Total damage (Records)
ERC	1458
ACT	15587
SCIT	1221

5.2. Increasing the Number of Attacks

In order to evaluate QoSS performance of each scheme, we set λ to 0.00048, and gradually increase λ per unit time. The request arrival time is fixed to 0.002 seconds in this experiment, because the processing capacity of the system based on SCIT cannot handle more requests per time. We set α to 0.3 in this experiment.

Figure 7 and Table 4 show the results of the experiment. Although the request arrival time is fixed, the response time of the system based on SCIT is increased as shown in Figure 7. This is because SCIT reduces the exposure time depending on λ. Similarly, ERC can predict the expected damage by attacks and decrease the exposure time to reduce the negative effects of intrusions.

The differentiated feature of ERC and SCIT is the capability to retain QoS performance. Although the number of compromised records when adopting ERC is larger than that of adopting SCIT, it is an acceptable trade-off for the better QoS. ACT endures a denial of service (DoS) attack by predicting threatening situation and cluster substitution. However it is hard to control the exposure time depending on the attack rates. As shown in Table 4, the total damage of ACT is much larger than that of other architectures. It means that ACT cannot cope with attacks except denial of service attack.

Figure 7. Response time with increased number of attacks.

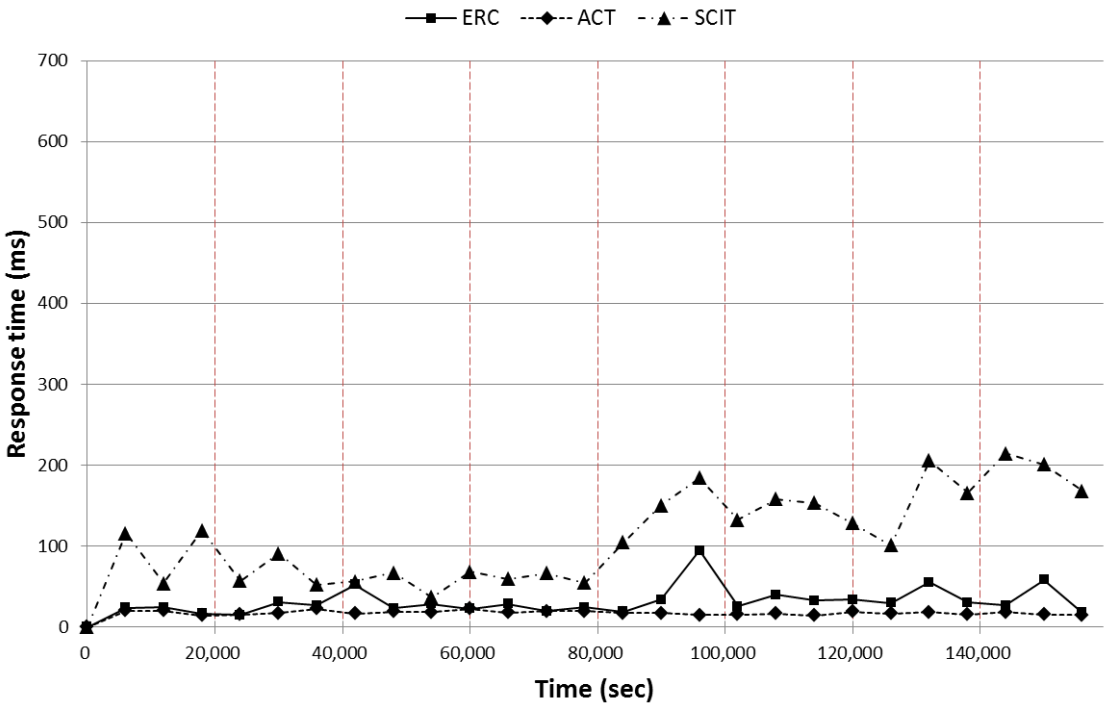


Table 4. The number of compromised records with increased number of attacks.

Scheme	Total damage (Records)
ERC	1503
ACT	28600
SCIT	1358

5.3. QoS and QoSS in the Practical Environment

Generally, attackers prefer to damage popular Web sites, to make more profit. Therefore, if the number of requests that are sent to the system is large, it means that the number of attacks that occur to the system is large, too. In this experiment, we generate requests at intervals of 0.0025 seconds and set λ to 0.00048 at initiation of the simulation, and increase the request volume and the probability of attacks per unit time gradually. We set α to 0.5.

Figure 8 and Table 5 show the results of the experiment. The experimental results indicate that the system based on ERC maintains a certain level of QoS and QoSS. ERC controls the exposure time according to the request arrival time and λ. Since the extension of the exposure time causes increment of the total damage done to the system, the number of compromised records in this experiment is larger than that of experiments in Sections 5.1 and 5.2.

Figure 9 depicts the total damages of the system based on ACT, ERC and SCIT in experiment 5.1, 5.2, and 5.3. The damage done to the system based on ERC is slightly larger than SCIT in all environments. However, the damage is much smaller than that of ACT and the QoS performance gain is large enough to make the gap negligible.

ACT and SCIT determines the exposure time in order to achieve their goals, which are maintaining specified system performance and keeping the same levels of reliability, respectively. Although ACT

and SCIT accomplish their purpose, ERC is more suitable to retain the performance of the system in the practical environment.

Figure 8. Response time in the practical environment.

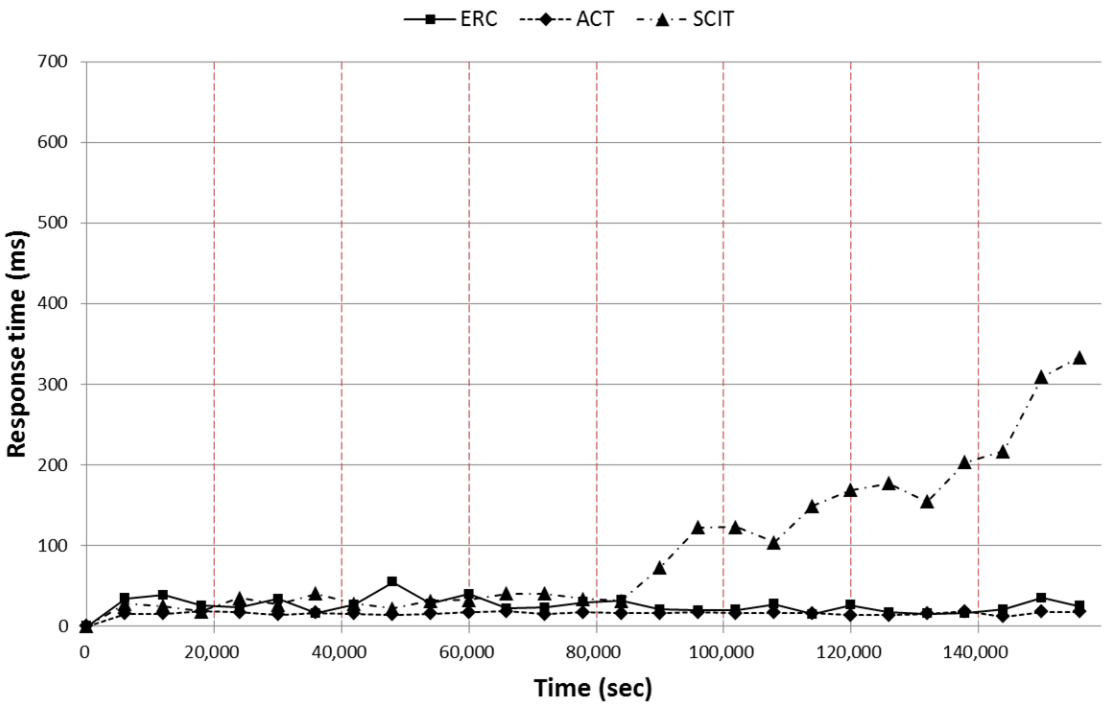
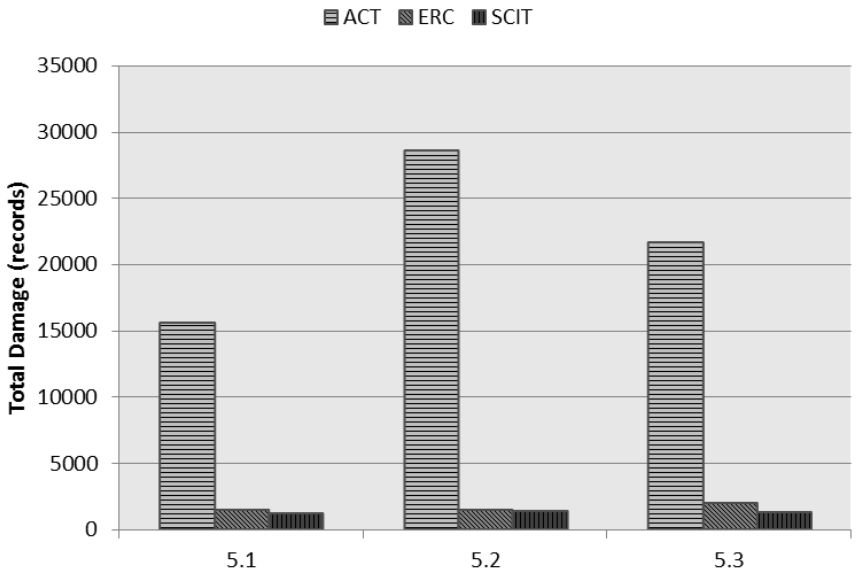


Table 5. The number of compromised records in the practical environment.

Scheme	Total damage (Records)
ERC	2008
ACT	21646
SCIT	1291

Figure 9. The number of compromised records in Section 5.1, 5.2 and 5.3.



5.4. Discussion

In this paper, we evaluate damage that can be caused by attacks by the number of compromised records. This evaluation method is very efficient to compare security performance of SCIT, ACT, and ERC. However, a successful attack does not only compromise the records in the real world.

Attackers can severely limit the exposure time by inflating λ, perhaps leading to a DoS. Still, this can be overcome by establishing a threshold for the exposure time. If a rising λ causes decrement of the exposure time under the threshold, then other defense systems will be activated to protect the system.

If VMs get corrupted, they will be restored to pristine state in cleansing period. However, any records or services hosted by back-end systems, such as database entries, can still be a potential risk. It can be overcome by assigning different priorities to VMs. Old VMs will have lower priority to request which needs to access back-end services, because they are more likely to be corrupted.

In our previous work [16], we halt a VM's operation if an attack on the VM is successful. The successful attacks decrease throughput of the system by 33%. Additionally, a malicious VM is more dangerous than external attackers, because the corrupted VM can execute any action to disturb and collapse the system.

Proactive-reactive recovery allows correct VMs to force the cleansing of a VM that is detected or suspected being faulty [9]. A limitation of this approach is a detection rate, because their detection scheme inspects VMs by only messages which are sent by VMs. Furthermore, though correct VMs have an authority to shut down an attacked VM, a recovery of the attacked VM is not perfectly guaranteed compared with the centralized architecture.

It is impossible to detect all kinds of attacks and failures, but a detection scheme that can detect a certain class of attacks, such as the internal DoS attack, is mandatory for reducing the impact of attacks. We will research on a detection scheme suitable to ERC in the next work.

6. Conclusions

Most current information systems are connected to the Internet, and it is impossible to successfully protect such systems against all the threats. In this context, an ITS has received significant attention as one of the most important security solutions.

In this paper, we introduce a novel intrusion tolerance architecture based on ERC. We present a QoS and QoSS scoring scheme, which is utilized for the resource conversion according to the internal and external factors. The central controller assigns the appropriate exposure time to VMs, and it is shown that ERC can maintain a desired level of QoS and QoSS as compared with conventional research.

With the advent of advanced threats, in order to provide services with better QoS and QoSS over open networks, research of a scheme to detect compromised VMs is required. In future work, we might consider the detection scheme as well as our scoring scheme.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MEST) (No. 2014R1A2A2-A01006957) and IT R&D program of MKE/KEIT (10041244, SmartTV 2.0 Software Platform).

Author Contributions

All the authors make substantial contributions to conception and design. Heo acquires and analyzes data. Heo participates in drafting the article. Lee, Doo and Yoon revise it critically for important intellectual content.

Appendix

A.1. Discussion Probability Functions

Double Exponential (Double Mean)

We use this function to generate a request at appropriate intervals, which are exponentially distributed with the given mean value. Each request gains exclusive use of the node, they delay for a service period (exponentially distributed with the given mean value) and then depart.

A.2. Implementations

We provide our code in CSIM20 regarding the implementations of ERC, ACT, and SCIT at the below links, respectively.

http://nslab.kaist.ac.kr/codes/ERC.cpp
http://nslab.kaist.ac.kr/codes/ACT.cpp
http://nslab.kaist.ac.kr/codes/SCIT.cpp

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Malkawi, M.I. The art of software systems development: Reliability, availability, maintainability, performance (RAMP). *Hum.-Centric Comput. Inf. Sci.* **2013**, *3*, doi:10.1186/2192-1962-3-22.
2. Jingle, I.D.J.; Rajsingh, E.B. ColShield: An effective and collaborative protection shield for the detection and prevention of collaborative flooding of DDoS attacks in wireless mesh networks. *Hum.-Centric Comput. Inf. Sci.* **2014**, *4*, doi:10.1186/s13673-014-0008-8.
3. Chung, Y.; Choi, S.; Won, D. Lightweight anonymous authentication scheme with unlinkabilty in global mobilty networks. *J. Converg.* **2013**, *4*, 23–29.
4. Kahate, A. *Cryptography and Network Security*, 3rd ed.; McGraw Hill Education: Delhi, India, 2013; p. 499.

5. Wang, F.; Gong, F.; Sargor, R.; Goseva-Popstojanova, K.; Trivedi, K.; Jou, F. SITAR: A Scalable Intrusion Tolerance Architecture for Distributed Services. In Proceedings of IEEE Second SMC Information Assurance Workshop 2001, New York, NY, USA, 5–6 June 2001; pp. 38–45.
6. Chong, J.; Partha, P.; Atigetchi, M.; Rubel, P.; Wevver, F. Survivability architecture of a mission critical system: The DPASA example. In Proceedings of the 21st annual Computer Security Applications Conference, Tucson, AZ, USA, 5–9 December 2005; pp. 495–504.
7. Saidane, A.; Nicomette, V.; Deswarte, Y. The design of a generic intrusion-tolerant architecture for Web Servers. *IEEE Trans. Dependable Sec. Comput.* **2009**, *6*, 45–58.
8. Sood, A.; Nguyen, Q. Realizing S-Reliability for Services via Recovery-driven Intrusion Tolerance Mechanism. In Proceedings of 2010 International Conference on Dependable Systems and Networks Workshops (DSN-W), Chicago, IL, USA, 28 June–1 July 2010.
9. Sousa, P.; Bessani, A.N.; Correia, M.; Neves, N.F.; Verrisimo, P. Highly Available Intrusion-Tolerant Services with Proactive-Reactive Recovery. *IEEE Trans. Parallel Distrib. Syst.* **2010**, *21*, 452–465.
10. Irvine, C.; Levin, T. Quality of security service. In Proceedings of the 2000 Workshop on New Security Paradigms, Cork, Ireland, 18–21 September 2000; pp. 91–99.
11. Schwetman, H. CSIM19: A Powerful Tool for Building System Models. In Proceedings of the 2001 Winter Simulation Conference, Arlington, VA, USA, 9–12 December 2001; pp. 250–255.
12. Lim, J.; Kim, Y.; Koo, D.; Lee, S.; Doo, S.; Yoon, H. A novel adaptive cluster transformation (ACT)-based intrusion tolerant system for hybrid information technology. *J. Supercomput.* **2013**, *66*, 918–935.
13. Kim, H.; Lim, J.; Yoon, H. A design of a novel intrusion tolerant system using virtual machine image analysis and secure exposure policy. *Telecommun. Rev.* **2012**, *22*, 904–912.
14. Stankovic, V.; Bessani, A.; Daidone, A.; Gashi, I.; Olbelheiro, R.R.; Sousa, P. Enhancing Fault/Intrusion Tolerance through Design and Configuration Diversity. In Proceedings of 3rd Workshop on Recent Advances on Intrusion-Tolerant Systems 2009, Lisbon, Portugal, 29 June 2009; pp. 600–601.
15. 2012 Data Breach Investigation Report. Available online: http://www.wired.com/images_blogs/threatlevel/2012/03/Verizon-Data-Breach-Report-2012.pdf (accessed on 10 November 2014).
16. Heo, S.; Lim, J.; Lee, M.; Lee, S.; Yoon, H. A Novel Intrusion Tolerant System Based on Adaptive Recovery Scheme (ARS). *Lect. Notes Electr. Eng.* **2013**, *215*, 71–78.