International Journal of Security and Its Applications
Vol.8, No.2 (2014), pp.291-300
http://dx.doi.org/10.14257/ijsia.2014.8.2.30

Design of Safe Internal Network with the Use of Active Tracking System

Hyung-Kyu Choi* and Seung-Jung Shin*

*Hansei University, GyeongGi-Do, Korea
Choihk2@naver.com

Abstract

Seeing the actual condition of cyber terror that is happening these days, a situation on cyber security is being emerged as serious social issue beyond damage of an individual or enterprise. Above all, it is a situation of being difficult to guess damage, which may occur due to the leakage of personal information or to the leakage of secret information in enterprise or institution. To prevent this hacking damage, a system or solution is being developed diversely. However, it is difficult to perform security in the perfect form. The real situation is that requires the technology available for perceiving hacking incident in advance ahead of this, as well as the technology of detecting and coping with hacking incident in the shortest time in the aspect of range or scale in damage, which is created by hacking incident. The purpose of this study is to suggest solution on the whole defense of system through network by grasping the whole situation on this cyber terror. For the objective dubbed the internal data loss prevention, the mechanism is needed that can preferentially analyze and monitor suspicious behavior and also that can closely analyze data of being doubtful about malicious link. The aim is to propose mechanism available for preparing for potential attack by expanding the subjects of this analysis even into the internal network and separately-divided network without setting limits to the external network.

Keywords: *Data loss Prevention, Suspicious behavior, Malicious code, Malware, Internal User*

1. Introduction

It is a situation of being continuously grown damage caused by data leakage of an individual or enterprise amidst social issue caused by cyber terror. System or solution for preventing this hacking incident is being developed variously, but is very difficult to perform security in the perfect form. The real situation is that requires the technology of safely protecting internal network by perceiving hacking incident in advance ahead of this, as well as the technology of detecting and coping with hacking incident in the shortest time in the aspect of range or scale in damage, which is created by hacking incident. Also, this countermeasure has been concentrated on external network. However, there is now a need of thoroughly preparing for internal network, too. Internal users judge that it is relatively safe. However, a case of connecting the media infected with malicious code is happening frequently through USB or laptop, which is brought inside unwillingly. In this way, all the internal networks are infected momentarily through loose route of security. A case of being happened huge damage through this took place frequently this year. In case of large bank these days, the malicious code was infected through employee of the external maintenance firm for bank computer network, thereby having been created the state that the whole of the corresponding bank was paralyzed. This can be indicated problems about which the

ISSN: 1738-9976 IJSIA

management on external business employee available for getting access to internal system was negligent, and about which there was no automated system available for inspecting and intercepting this malicious-code infection with real time. This study aims to suggest solution available for defending cyber attack by designing active tracking system through network after grasping the whole situation on this security. For the internal data loss prevention, the mechanism is needed that can preferentially analyze and monitor suspicious behavior and that can closely analyze data of being doubtful about malicious code or malicious link. The subjects of this analysis are expanded even into the internal network as well as the external network, thereby being allowed to prepare for potential attack.

The suggested system in this study offers the basic URL filtering function, malicious-code detection function (L7 filter engine), internal web-server test function (verifying integrity of bulletin board), and data loss prevention function through malicious code, and comprehensively proposes the firewall connection function, malicious URL DB and malicious-link URL DB connection as well. The basic URL filtering function monitors packet, then judges the appearance of malicious URL, and intercepts in case of malicious URL. Also, when it is not in malicious URL DB out of external URL, which is tried to be interfaced from the inside, the URL information is sent to L7 filter engine, thereby being allowed to judge the appearance of malicious URL. Even in addition to malicious URL information of being provided from malicious-link DB, L7 filter engine analyzes itself, thereby performing the function of detecting malicious URL. It delivers information on packet, which will intercept by storing it in DB and being linked to firewall. In the internal web server test, it performs the function of testing the appearance of malicious falsification on web page by testing malicious link and periodically testing integrity on web-server data as for data to which general users can get access such as bulletin board or Q&A. The data loss prevention function tracks and inspects leakage of malicious code by monitoring data of being transferred through web or e-mail.

2. Text

This study aims to design system and mechanism available for maximally defending the internally secret data loss through malicious code or other cyber terror by analyzing suspicious behavior through the detailed item. Accordingly, it is very important to well grasp characteristics of APT attack and to well analyze data loss type. The aim is to suggest a mechanism of detecting in advance cyber-attack anomaly by using diverse forms of profiles, and a system available for safely maintaining the internal network through this. The major functions in the suggested system can be divided largely into 4 functions such as URL filtering function, internal-network protection function, web-server integrity test function, and data loss prevention function. Figure 1 below is the detailed composition diagram of the suggested system.

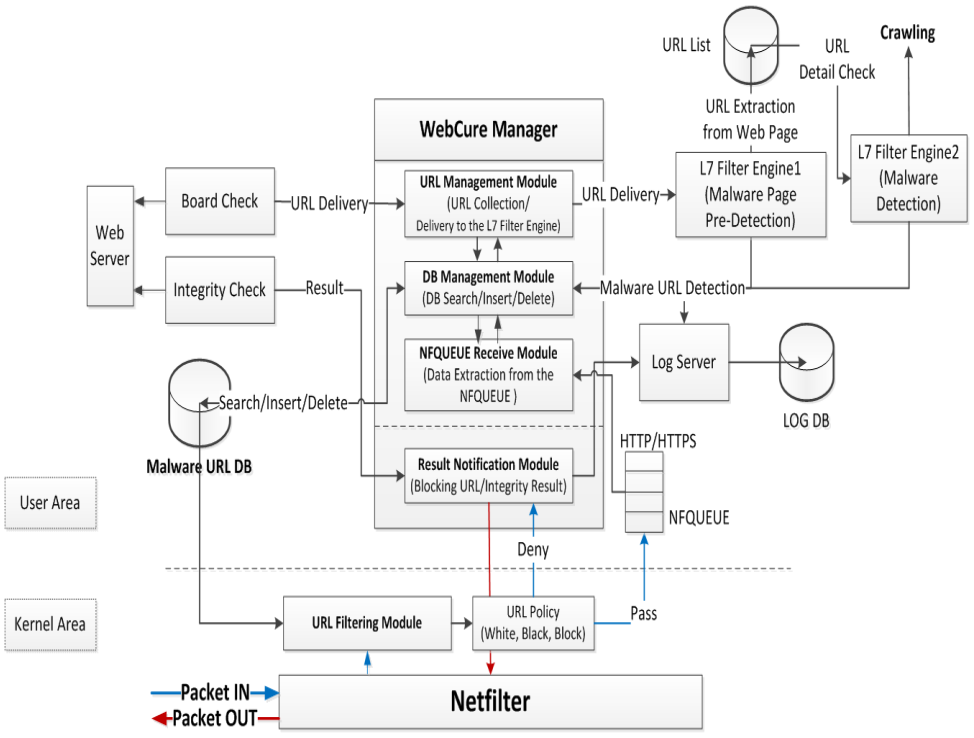

Figure 1. Detailed Composition Diagram in the Proposed System

2.1. URL Filter

As URL filter aims to offer security service on web-page connection, it intercepts malicious URL connection by comparing it with malicious URL information. At this time, it tests by mirroring packet, thereby not intercepting substantially and delivering information on interception to inline network security equipment such as the linked firewall. Even if being simple process of detecting URL in the packet level, but the proposed plan in this study re-tests the extracted URL by additionally using crawling technique, thereby offering function of making it not linked to malicious link with real time. As this function is the one that is not supplied now, it can be a plan available for making this study specialized. URL filter extracts URL information of the mirrored packet, which gets out from the outside to the inside and from the inside to the outside, thereby matching it with malicious URL information, which is uploaded into kernel memory. If packet URL was matched with malicious URL, the packet information is sent to firewall in order to be intercepted in firewall with leaving log. Given having not been matched, URL information is delivered to L7 filter by using Netlink Socket in order to be analyzed with real time.

2.2. L7 Filter Engine

L7 filter engine analyzes the delivered URL with real time and then judges the appearance of malicious URL. URL test subjects are tested URL, which isn't matched with malicious URL DB, and URL, which was found given testing web bulletin board. Given being not matched with malicious URL DB in URL filter, URL information is delivered for testing the corresponding URL test. The internal web-server test is delivered to judge whether the

corresponding URL is malicious when there is URL information in bulletin board or Q&A of the internal web server. L7 filer engine crawls itself given being inputted URL information, thereby judging the appearance of malicious URL through Hash pattern & script entropy calculation, anomaly-character counting, and heuristic pattern detection.

An analytical module, which is used in common in analyzing packet of getting out from the inside to the outside, and in testing bulletin board of internal web server, is composed of 2 processes and 1 database. Each operates organically. ANLFilterD process in Figure 2 is the process of judging the appearance of malicious URL, and downloads web page of the corresponding URL given receiving URL information.

The link, which was extracted by sequentially parsing the downloaded web-page source, is stored in L7 DB for analyzing in CRAFilterD process as Figure 3. And then, the web-page source is judged the appearance of malicious URL by using anomaly Iframe detection, vulnerable Object detection, making it normalized for generating hash pattern, script entropy calculation, anomaly-character counting, and heuristic pattern detections. Given being judged to be malicious URL, URL information is stored in malicious URL DB, thereby being allowed to perform the filtering function in kernel.

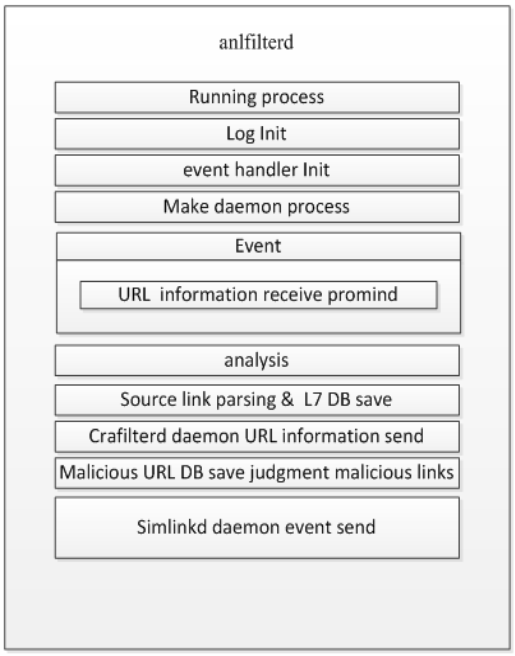

Figure 2. ANLFilterD

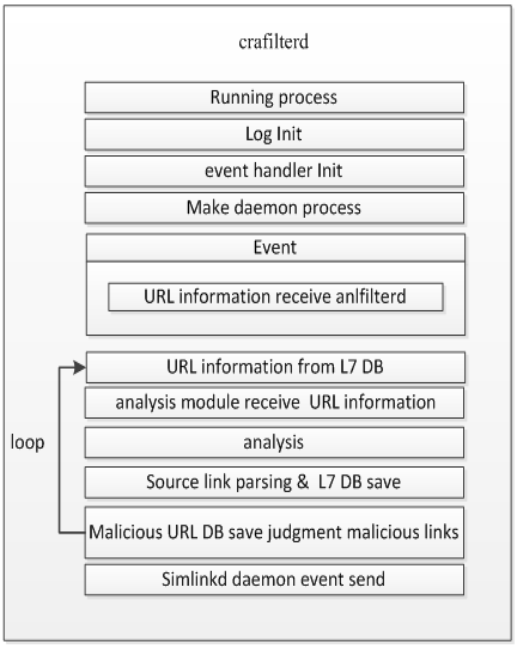

Figure 3. CRAFilterD

2.3. Internal Network Protection Function

Influx and propagation in malicious code are not what is made merely through web server, but happen much more frequently in the way of which a user of internal network surfs the web. This is being defended and cured mostly through vaccine on PC. However, in case of new malware, it cannot help being infected defenselessly. The infected PC additionally propagates in the internal network. Accordingly, this study suggests a plan for monitoring all the web protocols, which occur in the internal network by utilizing the proposed system in the above for preventing this in advance, and for possibly preventing infection originally from malicious cod by performing DB search and crawling through extracting URL out of it. Even what tests bulletin board on web server and tests integrity in addition to this function can be

said to be certainly necessary function. Most of the hackers attack web server, thereby implanting malicious code or malicious link. Thus, periodically testing safety of web server can be said to be prerequisite for safety of internal network.

2.4. Internal Web-server Integrity Test

It is the function of confirming the appearance of malicious modification by periodically carrying out integrity test on web-server data in the internal interwork for protecting web server in the internal network. This inspects web-server source in file unit, thereby performing even the function of recovering it to the original source by immediately coping with this given having been changed without notice.

2.5. Data Loss Prevention Monitoring

As the data loss prevention monitoring aims to offer security service of monitoring data leakage through mail, it can be considered with 2 kinds such as the web mail service through web browser and the service of using email-only program. The data loss prevention monitoring is divided into 2 kinds such as the test of attached file and the test of string. First of all, it confirms whether there is attached file on mail. It monitors by comparing it with the attached-file extension information, which is set in advance, when the attached file exists. When a file with extension of being included in the test subject is attached to mail and is transmitted to the outside, the mail sender information is stored to be recorded.

Also, string is tested on title and text of mail. String to be tested can be set. It sets up strings such as “address book,” “meeting,” and “organization chart” and stores sender’s information when there is consistent string among mail contents. The aim of monitoring system is to confirm sender information when data were leaked, simultaneously with inhibiting data loss.

3. Simulation

3.1. URL Filtering Performance Experiment

This experiment analyzes whether there is difference in performance between when being performed in application level and when being performed in the proposed system as for URL filtering. To measure performance, Http protocol packet is used. Filtering speed according to each of the malicious URL pieces is measured. It uses 2 units of system and kernel-2.6.32.60 in the same condition. In the application level, proxy gateway is used. URL filtering in the proposed system used hash-tree algorism for making filter speed in the targeted URLs fast. In case of forming a list only with URL, a method of needing to allow URL information, which is inputted in kernel, to be matched from the beginning to the end, gives a load to a system and comes to be lengthened even in search frequency, thereby bringing about a drop in speed. The proposed system improved speed by using a method of a hash-tree search by dividing URL into domain, host, directory, and file, respectively.

Measurement was made the filtering time in kernel according to number of malicious URL pieces. For experiment, URLs were collected randomly by crawling URLs to be matched. It divided number of URL pieces for measuring performance in URL filtering, respectively, into 1,000 pieces, 10,000 pieces, and 100,000 pieces, thereby having measured time according to number of pieces. Table 1 below is showing results on this.

Table 1. Results of Filtering Processing Speed According to Number of Malicious URL Pieces

Classification	Number of malicious URL pieces		
	1,000 pieces	10,000 pieces	100,000 pieces
Kernel search speed(sec.)	0.12 sec.	1.45 sec.	3.85 sec.
Application level search speed(sec.)	1.20 sec.	3.20 sec.	9.30 sec.

3.2. Internal Network Protection Experiment

This experiment needs to first measure performance in L7 filter preferentially for filtering URL. Accordingly, it measured performance in L7 filter, which analyzes URL with real time. For the corresponding experiment, it measured URL analytical time according to each number of the pieces in L7 filter by setting up URL according to each number of the pieces in advance in a file without applying malicious URL in URL filter. Table 2 below is showing the analytical results of L7 filter.

Table 2. Results of Analyzing L7 Filter Performance

Classification	Number of malicious URL pieces to be analyzed		
	1 piece	10 pieces	100 pieces
Time of having been analyzed in L7 filter	2.00 sec.	21.00 sec.	230.00 sec.

Seeing the experimental results in Table 2, it confirmed to have been taken 2.00 seconds when number of URL pieces is 1 piece, 21.00 seconds given being 10 pieces, and 230 seconds given being 100 pieces. As for a reason of being different in time without being increased by 10 times in time every piece number, it confirmed that the analytical time is different according to number of lower domains to be derived because even the links to be derived in upper domain are analyzed with real time. The analysis in the current L7 filter is being progressed in order of being put in Queue without priority. In the wake of being progressed without priority, there is a case that URL with large traffic is processed first compared to URL with small traffic. It can be known that there is a need of policy available for analyzing preferentially by selecting priority according to traffic in L7 filter.

3.3. Web-server Integrity Test Experiment

This paragraph progressed experiment on function of testing integrity of web server. To test internal network, firewall was located in the middle, thereby having been allowed to show the same effect as the actual internal network.

The corresponding test system environment was used Linux (CentOS 6.3) iptables v1.4.7 version as firewall and Apache 2.2.14, PHP 5.2.12, MYSQL 5.1.39 as web server. Three kinds of functions that will be tested in the web-server integrity test function include test cycle, test object, and file recovery. The test cycle is set up on when the integrity test will be executed and is available for setting by minute, by time, and by day. The test object can be set a subject in a file to be tested integrity. Whole files within system can be set. Directory or single file can be set. The appearance of recovery is set up on whether recovering it to the

existing backup file given being discovered falsification and modulation as a result of testing integrity. Table 3 below is scenario that was set for performing this experiment.

Table 3. Function Test Scenario of Integrity Test

Classification	Test cycle	Test object	Recovery appearance
Set up ①	5 min.	Whole	Non-recovery
Set up ②	10 min.	test.txt(unit file)	Non-recovery
Set up ③	Manual(instantly)	Test(directory)	Recovery

Figure 4 below is the outcome of integrity test targeting anomaly file. Figure 5 below is the outcome of integrity test targeting anomaly directory.

```
PROGRESS: validating integrity of    /home/sijung/Test
STATUS:
        WARNING: [csrc]    /home/sijung/Test/test.txt
        [Inodes: 17586166-17586165, Sizes: 26-29, Times: Nov 04 04:10 2013 – Nov 04 04:132013]
```

Figure 4. Anomaly File Test Results

```
PROGRESS: validating integrity of    /home/sijung/Test/
STATUS:
        WARNING:    [csrc]    /home/sijung/Test/test1.txt
        [Inodes: 17586166-17586170, Sizes: 6-8, Times: Nov 04 04:47 2013 – Nov 04 04:48 2013]

        WARNING:    [csrc]    /home/sijung/Test/test2.txt
        [Inodes: 17586164-17586159, Sizes: 6-9, Times: Nov 04 04:47 2013 – Nov 04 04:48 2013]

        WARNING:    [csrc]    /home/sijung/Test/test3.txt
        [Inodes: 17586167-17586166, Sizes: 6-8, Times: Nov 04 04:47 2013 – Nov 04 04:49 2013]
```

Figure 5. Anomaly Directory Test Results

It confirmed that the test on integrity of web server is available for inspecting safety in web server and for testing on and off as for malicious user. In selecting test object, the files of being changed normally need to be processed in order not to be happened error of integrity through the integrity update function. System-based configuration is judged to be needed in order to possibly set efficiently the classification of a file with high importance and a file with low importance in selecting test object.

4. Conclusion

The proposed system in this study is judged to be effective mechanism for reconstructing the existing mechanism for coping with cyber hacking or cyber terror and for preventing hacking or data loss more efficiently. The existing cyber hacking or cyber terror showed a character of attack for stopping service or aggravating confusion in enterprise or institution. However, the recent attack showed lots of behaviors for additionally capturing the internally corporate secret or personal information.

Also, data loss is happening frequently in the face of having invested much in network boundary because internal user makes malicious code flow in that can be used in hacking,

through USB or laptop from the outside. Accordingly, a plan for preventing internal data loss is in a situation of needing to closely examine even internal traffic as well as network boundary. This study designed in order to progress interception and prevention on suspicious behavior in the network boundary for blocking this completely and also to possibly analyze and intercept with real time as for malicious link, which may occur in internal traffic, with recognizing seriousness of this data loss damage. Through this, it supplemented a blind point on the existing internal network and suggested a general measure of hacking and data loss prevention.

In addition, it designed in order to possibly prevent malicious code or malicious-code link from being infected with web server through the periodic integrity test on web server, which is served from the inside. Also, to optimize this function with real time, even a flow was designed and proposed. It is a situation that perfect solution or system on malicious code is nonexistent now. What defends this malicious code with single plan will be unreasonable. Accordingly, the best defensive measure will be what prepares omni-directionally by complexly designing useful functions that were proposed in a study. Still, the experiment was carried out as what monitors data loss type by setting limits to a case of being leaked through e-mail or web. Thus, this part will be available for defending more efficiently given being equipped with the internal data loss prevention based on behavior through additional research.

This study suggested complex system for solving social issue on the current malicious code. Each of the proposed core modules is available for operating even independently and may perform complexly organic function. This organic combination is confirmed to be likely able to be conducive to arranging fundamental countermeasure against cyber hacking with escaping from characteristic of UTM (Unified Threat Management), which had been the existing aggregate in simple function.

Acknowledgements

This article is a revised and expanded version of a paper entitled [A Study of Data Loss Prevention System with Suspicious Behavior Trace] presented at International Symposium on Advanced and Applied Convergence held on November 14-16, 2013 at Seoul, Korea.

References

[1] H.-K. Choi and S.-J. Shin, “A Study of Data Loss Prevention System with Suspicious Behavior Trace”, ISAAC 2013/ICACT 2013, AACL 01, **(2013)**, pp. 1-4.
[2] L. Seung-won, N. Yeong-seop and H. Chang-u, “A Study on the Design and Implementation of an Digital Evidence Collection Application on Windows based computer”, Korea Institute of Information Security & Cryptology Thesis Paper, vol. 23, no. 1, **(2013)**.
[3] H. Geon-il, P. Chan-uk, P. Won-hyeong and G. Gwang-ho, “A Study on Improvement in Evidence Collection Module of Malicious Code Based on Windows”, Information Security Paper, vol. 10, no. 3, **(2010)** September, pp. 61-68.
[4] Guidebook of Intrusion Accident Analysis Procedure, Korea Communications Commission . Korea Internet & Security Agency, **(2010)** January.
[5] WebSense TRITON, “Threat Analysis and Defense Strategies for SMB, Mid-size, and Enterprise Organizations”, Websense, Inc., REV 2, **(2012)**, pp. 1-14.
[6] WebSense TRITON, “Miercom v7.7.3”, Websense, Inc., DR130214, **(2013)**, pp. 1-35.
[7] M. K. Rogers, “Computer Forensic Field Triage Process Model”, Conference on Digital Forensics, Security and Law, Las Vegas, Nevada, USA, **(2006)** April 20-21, pp. 27-40.

Authors

Hyung-Kyu Choi
Year in 1984 Bachelor of Computer Science and Engineering, Chungbuk National University
Year in 2003 Master of Information Security, Soongsil university
2009~2013Doctorate course for the Dept. of IT, Hansei University
1986~Present Working at IT section of public institution

Seung-Jung Shin, he received the Ph.D. degree from Kookmin University in 2000, Korea. Currently, he is a professor at Department of Information & technology, Hansei University. His current research interests include security and privacy.