A Study on the Live Forensic Techniques for Anomaly Detection in User Terminals

Ae Chan Kim[1], Won Hyung Park[2] and Dong Hoon Lee[3]

[1]*Dept. of Financial Security, Graduate School of Information Security, Korea University, 145 Anam-dong 5-ga, Seongbuk-gu, Seoul, 136-713, Korea*
[2]*Dept. Information Management, Far East University, Wangjang-ri, Gamgok-myeon, Eumseong-gun, Chungbuk, 369-700, Korea*
[3]*Graduate School of Information Security, Korea University, 145 Anam-dong 5-ga, Seongbuk-gu, Seoul, 136-713, Korea*
[1]*holytemple@korea.ac.kr*, [2]*whpark@kdu.ac.kr*, [3]*donghlee@korea.ac.kr*

Abstract

Digital forensics techniques that have been used to analyze system intrusion incidents traditionally are used to detect anomaly behavior that may occur in the user terminal environment. Particularly, for the method to analyze user terminals, automated live forensics techniques that are used as supporting tool for malicious code (malware) detection. We suggest a way to take advantage of the live forensic techniques for the anomaly detection of malware.

Keywords: *Live Forensics, Anomaly Detection, User Terminals, Malware*

1. Introduction

Recently, Malware have been created in order to commit a theft users permission via C&C botnet and stopover server. It makes information leakage by arbitrarily modifying or creation of system files. It defines the limitation of conventional pattern matching and signature detection of vaccine. A diverse range of malware has made user terminals ever more vulnerable. This situation gets worse as user terminals quickly evolve [1].

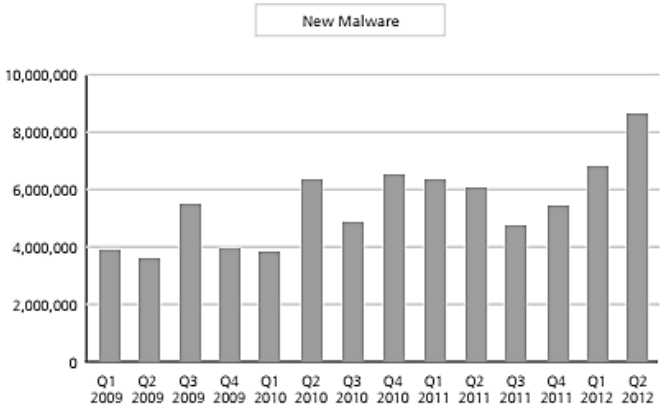


Figure 1. The New Malware

[1] First Author
[2] Corresponding Author

In this paper, we refer to the live forensics techniques; it will be used to analyze user terminal environment to protect normal users from malicious programs.

2. Related Work

Traditionally digital forensics meant the methodology or techniques to figure out causes of security incidents in the legal and investigatory prospects. However, the application range of digital forensics is expanded recently, and it is recommended for detecting malicious programs [2, 3].

"The information obtained from network device which plays an essential role in network and information security device which monitors and protects network." The information for log/state of server system itself or the information for security log is excluded. That is, information in network forensic is defined as the information for the system which plays a role as network device or information security device [4].

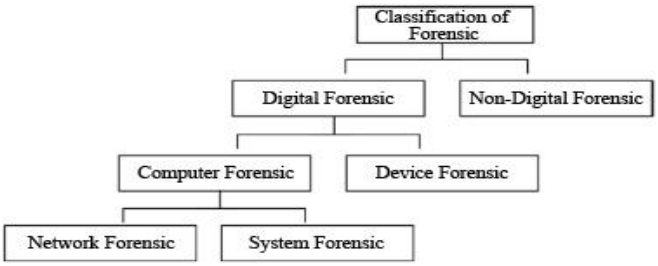


Figure 2. Classification of Forensic

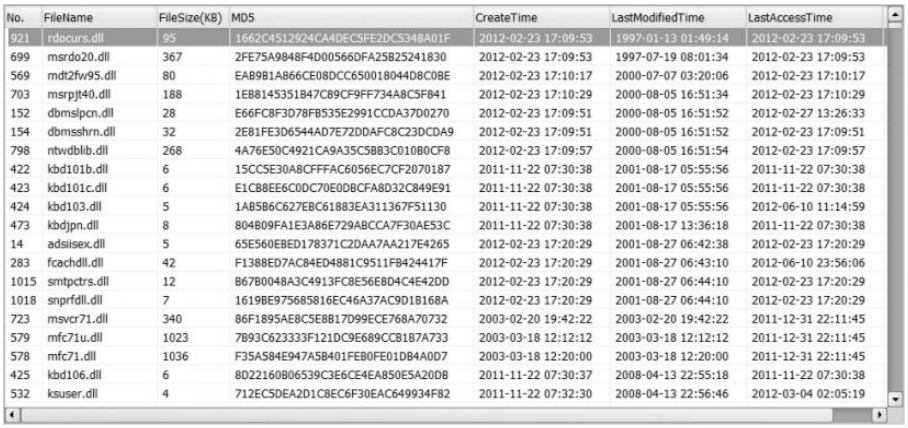
To name a forensics and analysis tool developed previously, "FRED (Frist Responder's Evidence Disk)" of the security incident response team of US Air Force is the windows based tool collecting malicious program proofs developed by AFOSI (Air Force Office of Special Investigation) in last 2000 to respond internal security incidents more promptly and effectively. This tool was open to the public by Digital Forensics Research Workshop on August 8th, 2002. Another one is Microsoft's "COFFEE (Computer Online Forensic Evidence Extractor)" which is the closed live response tool for government offices. Currently Law Enforcement Agencies distribute this program free of charge. Unlike to FRED or COFFEE just introduced, Ahnlab's "AhnReport" is GUI based, which enables correlation analysis on malicious program more easily than CLI based. Recently, the correlation analysis techniques of "AhnReport" is more enhanced to develop the program named "SysTrace", which is used for security incident response of national and public organizations [5].

3. Methodology

3.1. Timeline Analysis of System File

Most of malicious programs discovered up to now are installed in Windows system folders (C:\WINDOWS, C:\WINDOWS\SYSTEM32, C:\WINDOWS\SYSTEM) and temporary folder (C:\WINDOWS\TEMP). Commercial program file extensions such as swf, pdf, hwp and ppt are increasingly used to distribute malicious programs recently. However, in order for malicious programs to infect the system, ultimately they should reside in the above paths in the forms of DLL or EXE mostly. Therefore, it is necessary to set system files in those paths as analysis targets. One of typical attacks by malicious programs is downloading additional malicious programs (EXE, DLL) in the paths specified as above and executing

them or attacking existing DLL files by DLL injection [6]. For such reason, executable files (EXE) and libraries (DLL) within the Windows and system folders will be collected in priority. As shown in Fig. 3, the SysTrace analysis tool outputs information on collected system files for the forensics investigator able to intuitively analyze MAC Timeline.



No.	FileName	FileSize(KB)	MD5	CreateTime	LastModifiedTime	LastAccessTime
921	rdocurs.dll	95	1662C4512924CA4DEC5FE2DC5348A01F	2012-02-23 17:09:53	1997-01-13 01:49:14	2012-02-23 17:09:53
699	msrdo20.dll	367	2FE75A9848F4D00566DFA25B25241830	2012-02-23 17:09:53	1997-07-19 08:01:34	2012-02-23 17:09:53
569	mdt2fvv95.dll	80	EAB9B1A866CE08DCC650018044D8C0BE	2012-02-23 17:10:17	2000-07-07 03:20:06	2012-02-23 17:10:17
703	msrpjt40.dll	188	1EB8145351847C89CF9FF734A8C5F841	2012-02-23 17:10:29	2000-08-05 16:51:34	2012-02-23 17:10:29
152	dbmslpcn.dll	28	E66FC8F3D78FB535E2991CCDA37D0270	2012-02-23 17:09:51	2000-08-05 16:51:52	2012-02-27 13:26:33
154	dbmsshrn.dll	32	2E81FE3D6544AD7E72DDAFC8C23DCDA9	2012-02-23 17:09:51	2000-08-05 16:51:52	2012-02-23 17:09:51
798	ntwdblib.dll	268	4A76E50C4921CA9A35C5B83C010B0CF8	2012-02-23 17:09:57	2000-08-05 16:51:54	2012-02-23 17:09:57
422	kbd101b.dll	6	15CC5E30A8CFFFAC6056EC7CF2070187	2011-11-22 07:30:38	2001-08-17 05:55:56	2011-11-22 07:30:38
423	kbd101c.dll	6	E1CB8EE6C0DC70E0DBCFA8D32C849E91	2011-11-22 07:30:38	2001-08-17 05:55:56	2011-11-22 07:30:38
424	kbd103.dll	5	1AB5B6C627EBC61883EA311367F51130	2011-11-22 07:30:38	2001-08-17 05:55:56	2012-06-10 11:14:59
473	kbdjpn.dll	8	804B09FA1E3A86E729ABCCA7F30AE53C	2011-11-22 07:30:38	2001-08-17 13:36:18	2011-11-22 07:30:38
14	adsiisex.dll	5	65E560EBED178371C2DAA7AA217E4265	2012-02-23 17:20:29	2001-08-27 06:42:38	2012-02-23 17:20:29
283	fcachdll.dll	42	F1388ED7AC84ED4881C9511FB424417F	2012-02-23 17:20:29	2001-08-27 06:43:10	2012-06-10 23:56:06
1015	smtpctrs.dll	12	B67B0048A3C4913FC8E56E8D4C4E42DD	2012-02-23 17:20:29	2001-08-27 06:44:10	2012-02-23 17:20:29
1018	snprfdll.dll	7	1619BE975685816EC46A37AC9D18168A	2012-02-23 17:20:29	2001-08-27 06:44:10	2012-02-23 17:20:29
723	msvcr71.dll	340	86F1895AE8C5E8B17D99ECE768A70732	2003-02-20 19:42:22	2003-02-20 19:42:22	2011-12-31 22:11:45
579	mfc71u.dll	1023	7B93C623333F121DC9E689CCB1B7A733	2003-03-18 12:12:12	2003-03-18 12:12:12	2011-12-31 22:11:45
578	mfc71.dll	1036	F35A584E947A5B401FEB0FE01DB4A0D7	2003-03-18 12:20:00	2003-03-18 12:20:00	2011-12-31 22:11:45
425	kbd106.dll	6	8D22160B06539C3E6CE4EA850E5A20DB	2011-11-22 07:30:37	2008-04-13 22:55:18	2011-11-22 07:30:38
532	ksuser.dll	4	712EC5DEA2D1C8EC6F30EAC649934F82	2011-11-22 07:32:30	2008-04-13 22:56:46	2012-03-04 02:05:19

Figure 3. MD5-MAC Timeline Analysis of System File

For automated analysis on the collected system files, the detection model being proposed generates MD5 and checks if MAC Time values are applicable of conditions in Table 1.

Table 1. Detection Conditions of System File

Condition 1-1: Creation time > LastModified time Condition 1-2: LastModified Time > LastAccess Time

Conditions in Table 1 correspond to the MAC timeline analysis method of traditional digital forensics when investigating any security incident in the system [7]. If there is a system file satisfying one or both conditions above, the applicable file will be determined as highly probable that it is a malicious program or the file infected by a malicious program. The MD5 hash value will be generated and the malicious program information (DB) in the personal firewall program will be searched to additionally check if there is matching information. If there is an identical MD5 value in the malicious program information (DB), it is confirmed infected by the malicious program.

3.2. DNS Analysis: Hosts file, Phishing and Pharming Inspection

Since Hosts file is a text file, anyone can use a tool such as Notepad to open and modify it. If this fact is used to modify Hosts file, it can be a potential threat of attack. A recently occurred malicious program targeted of internet banking actually showed ‘pharming’ of manipulating Hosts file [8], and in a previous case of 3.4DDoS attack, Hosts file was manipulated to block downloading vaccine programs [9]. For automated ‘pharming’ detection, the detection model being proposed checks if applicable of the condition in Table 2.

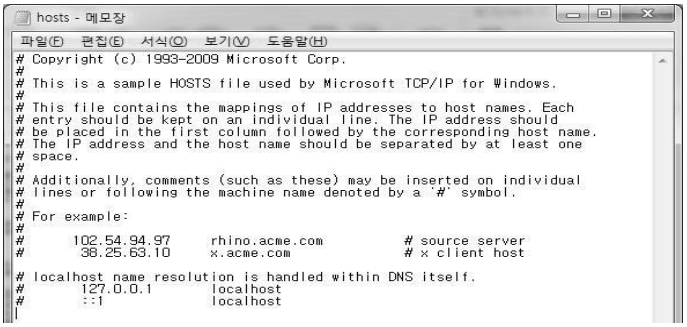


Figure 3. Hosts File in the Normal (Default) State (Windows 7)

Table 2. Character String Comparison for Hosts File

Condition 2-1: MD5 hash-values are calculated for Hosts file, and then compared with MD5 values of normal(default) Hosts file of the Windows operating system

Depending on kinds of Windows operating systems, as contents of Hosts files vary little by little, MD5 values vary too. Table 3 shows default MD5 hash values by operating systems. At this time, if the hash values are different from the default MD5 values, the default comparison targets, Hosts file is modified and it may be considered as 'pharming' attacked. However, with only that Hosts file is modified, it cannot be concluded as the 'pharming' attack symptom. Therefore it will additionally check if applicable of Condition 2-2 in Table 4. Use of "Strings" can be considered additionally as a character string extracting tool for analysis.

Table 3. MD5 Value of Normal (Default) Hosts File by Operation System

Operation System	MD5 (128 bit)
Windows XP	de1cbfe6c3086010af115a1f00909b01
Windows Vista	01505bb3f7004537f4f2c0fbba349a1f
Windows 7	9559da711c2abf477e95eeb41cebf637

Table 4. Character String Comparison for Hosts File

Condition 2-2: It is performed only if identified that Hosts file is modified by Condition 2-1 if Hosts file contains character strings of web site addresses of domestic financial companies.

For example, if web site address information of a public site (www.google.com, www.nate.com, banking.nonghyup.com and etc.) is modified or entered in Hosts file, it can be judged that a malicious program or an attacker deliberately attacked to induce 'pharming'.

Next is the DNS analysis method using 'ipconfig /displaydns'. 'ipconfig /displaydns' command outputs information saved in DNS Resolver Cache of the user system to show domain addresses of external hosts the system has accessed along with their IP addresses. For DNS Resolver Cache maintains not only the access point that the user tries to access normally by a web browser but also the access point that was tried to access in the anomaly method by operation of malicious program, DNS Cache of the user system can be used to check if infected by DNS Cache Poisoning attack. By DNS Cache Poisoning attack over the user

system, attackers or malicious programs manipulate its DNS Cache to induce ‘pharming’ attack.

For automated detection of any ‘DNS Cache Poisoning’ attack over the user system, the detection model being proposed checks if applicable of Condition 2-3 in Table 5.

Table 5. Detection Condition for DNS Cache Poisoning

Condition 2-3: By the result of ‘ipconfig /displaydns’ executed, if an electronic financial transaction (financial company) domain information in the ‘data name’ raw, its mapped IP (‘<host name> record’ value) is checked if matching with the normal financial transaction IP.

For example, when its execution result is as shown in Figure 4, the normal IP for ‘banking.nonghyup.com’ should be ‘61.37.254.31’. If not matched, it can be judged ‘pharming’ attacked.

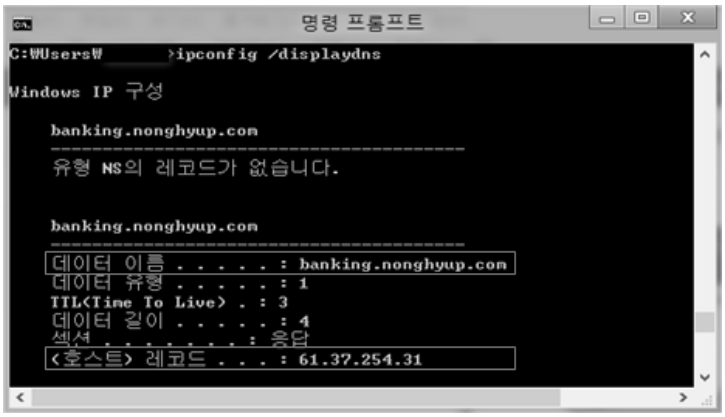


Figure 4. ‘ipconfig /displaydns’ Executed Window

3.3. Correlation Analysis between Network Connection and Processes

‘netstat’ was largely used as the existing tool to identify network connection information. However, even though the forensics investigator identified a suspicious address (overseas IP information of such as China, etc.) by the command, additional analysis was required to identify the domain address mapped on the IP address. Because it is difficult to figure out correlation only by use of ‘netstat’ generally known, for network connection information is not mapped with application programs.

Therefore, it is necessary of the method integrating network connection information and process information. In this paper, it is recommended to use ‘–navb’. Option -n identifies the port number connected to other host by IP address, option -a identifies all of the ports connected to other hosts or on standby, and option -b outputs the process name involved in each connection and the list of DLL files loaded by applicable processes only for well-known processes. Option –v is used with option –b, and outputs the list of DLL files loaded by respective processes for not only well-known processes but whole processes [10]. When the analysis tool (command) is executed, the protocol, the local address, the foreign address, the state and the process id (PID) are output. Figure 5 illustrates ‘netstat –navb’ executed window. The collected data is ‘the protocol information, the local IP address, the foreign IP address, presence of the connection state (listening and established), and the process name under execution associated with active connection’.

By mapping network connection information with program execution file (process) and its file list, the detection model being proposed can perform correlation analysis, and for automated analysis, it checks if applicable of Conditions 3-1 and 2 in Table 6.

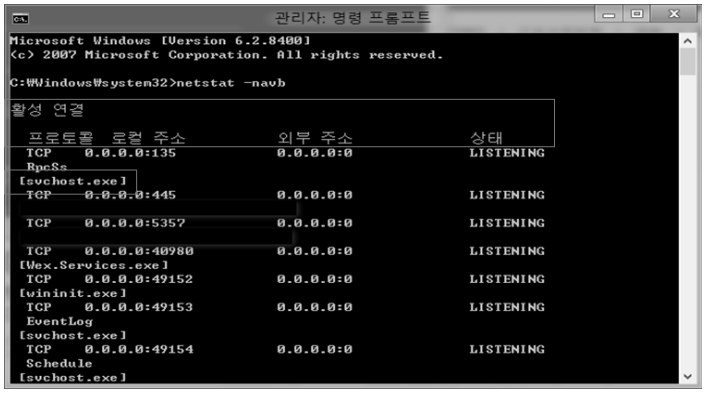


Figure 5. 'netstat –navb' Executed Window

Table 6. Detection Conditions for Network Connection and Process

Condition 3-1: If the value of the 'foreign address' item is within the foreign IP range (China, Taiwan, etc.) Condition 3-2: If the 'name of process under execution' matches with information in the known malicious program list.

If applicable of Conditions 3-1 and 2 in Table 6, it can be judged highly probable that attacks of information leakage or hijacking the administrator right (root) are under progress by network connection of a malicious program or an attacker.

3.4. ARP Analysis

ARP is the protocol used to map an IP address to the MAC address on the IP network. Here, MAC address means the 48 bit address uniquely assigned by network cards for Ethernet or token ring network. On physical communication, the logical address of IP is not used, but the physical network address is used. There is the 'ARP Cache Poisoning [11]' attack among the typical ARP Spoofing attacks abusing such ARP protocol vulnerability.

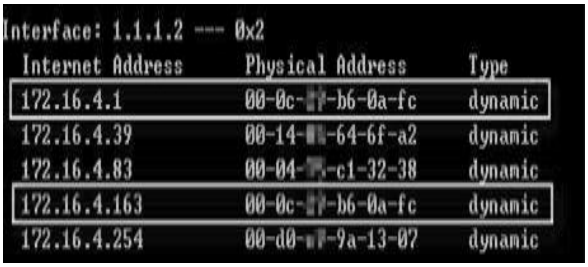


Figure 6. ARP Cache Information of the System Attacked by ARP Spoofing

Above Figure 6 shows ARP Cache information of the system attacked by ARP Spoofing, on which table IP addresses different each other are mapped on the same MAC address. If attacked, both hosts transfer all of the traffics to the attacker for they recognize MAC addresses of each other as the MAC address of the attacker. For the attacker is able to retransmit to both hosts, all of the packets are sniffed. As the conclusion, if ARP table is modified anomaly by an attacker, it can be judged that there is an intrusion threat. By this reason, for automated detection of ARP Spoofing attack in the same sub network the detection model being proposed checks if applicable of Conditions 4-1 and 2 in Table 7.

Table 7. Detection Conditions of ARP Spoofing

Condition 4-1: If there are two or more of a same MAC address in ARP table information.
Condition 4-2: After Condition 4-1 is met, and if the MAC address of the network gateway is changed into two or more of MAC addresses in ARP table information.

4. Conclusion

We refer to live response techniques for anomaly detection in user terminals. It was expected to improve limitations of existing forensics analysis methods and vaccine based on signature; these tools are based on correlation analysis for forensics investigators able to detect malicious programs more promptly and correctly. That is, the method based on live forensics will be used to detect the behavior of malware.

Acknowledgements

This work is supported by the Korea Information Security Agency (H2101-12-1001).

References

[1] McAfee, “McAfee Threats Report: Second Quarter 2012”, **(2012)**, http://www.mcafee.com/us/resources/reports/rp-quarterly-threat-q2-2012.pdf?cid=BHP010.
[2] K. Karen, C. Suzanne, G. Tim and D. Hung, “Guide to Integrating Forensic Techniques into Incident Response”, NIST SP 800-86, **(2006)**.
[3] D. Brezinski, “Guidelines for Evidence Collection and Archiving”, IETF RFC 3227, **(2002)**.
[4] ASEC Team, ASEC Report, Ahnlab Inc., **(2010)** September.
[5] Y. H. Lim, H. R. Ryu, K. S. Choi, C. W. Park, W. H. Park and K. H. Kook, “A Study on Malware Detection System Model based on correlation analysis Using Live Response techniques”. In: Proceedings of 2012 International Conference on Information Science and Applications (ICISA), **(2012)**, pp. 1-6.
[6] H. H. Park and D. W. Park, “A Study on New Treatment Way of a Malicious Code to Use a DLL Injection Technique”, Journal of the Korea society of computer and information, vol. 11, no. 5, **(2006)**, pp. 251-258.
[7] J. K. Kim, “Timeline Analysis”, **(2011)**, http://forensic-proof.com/archives/ 2323.
[8] INCA-CERT, “Internet Banking Malware, Google Code Spread through the attempt to bypass the Google Code hosting”, **(2012)**, http://erte am.tistory.com/313.
[9] Ahnlab, “3.4 DDoS Analysis Report”, Ahnlab Inc., **(2011)** March.
[10] Netstat, **(2012)**, http://en.wikipedia.org/wiki/Netstat.
[11] D. H. Jang, “ARP Spoofing Attack and Countermeasures”, Hanseo University, **(2007)**.