An Efficient Trojan Horse Classification (ETC)

Areej Mustafa Abuzaid , Madihah Mohd Saudi, Bachok M Taib and Zul Hilmi Abdullah

Faculty of Science and Technology (FST), Universiti Sains Islam Malaysia (USIM),
Bandar Baru Nilai, 71800 Nilai, Negeri Sembilan, Malaysia.

Abstract

For the past few years, malware or also known as malicious code is seen as one of the biggest threats of the cyber attacks. It has caused lot of damages, loss of money and productivity to many organizations and end users. Malicious code can be divided into many categories such as viruses, worms and trojan horses. Each of these categories has it owns implications and threats, and trojan horse has been chosen as the domain of this research paper. Prior to the formation of a new trojan horse detection model, an in-depth study and investigation of the existing trojan horse classification is presented in this paper. Surprisingly, not much research related with trojan horse has been done. On 16th January 2013, Troj/Invo-Zip has caused chaos by masquerading as an invoice from Europcar and spreading via email. Therefore, in this research paper, a new trojan horse classification called Efficient Trojan Horse Classification (ETC) is developed. This ETC later is used as a basis to build a model to detect trojan horse efficiently. The methods used to develop the ETC are the static and dynamic analyses. As for the dynamic analysis, cuckoo sandbox has been integrated to speed up the analysis and reverse engineering processes.

Keywords: *Trojan horse, classification, payload, static analysis, dynamic analysis, automated analysis.*

1. Introduction

Trojan horse has become a real threat to many organizations and computer users for more than a decade. Statistics taken from Cyber Security (2012) show that three types of major security incidents are often reported (i.e. fraud, intrusion and malicious code).

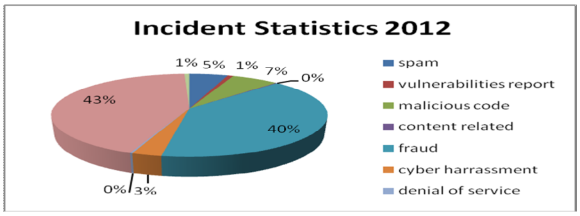


Fig. 1 Incident Statistics 2012
(Adapted from CyberSecurity Malaysia Incident Statistics (2012).

Hill [17] defined malicious code as any code added, changed or removed from a software system in order to intentionally cause harm or subvert the intended function of the system. Though the problem of malicious code has a long history, a number of recent, widely publicized attacks and certain economic trends suggest that malicious code is rapidly becoming a critical problem for industry, government, and individuals. One of the categorizations of the malicious code is known as trojan horse, which is the focus of this research paper. It is a malicious program, that must be executed in victim's computer and once it is installed, it can control the victim's computer remotely and steal any confidential information from it. It is different compared to worm and virus, since it has the capability to control the victim's computer remotely and it does not replicate itself [14].

As for the malicious code detection, classification is one of the crucial processes that must be place in order to ensure the effectiveness of the detection process. Generally malicious code can be classified based on the characteristics such as infection target and technique and other different characteristics [29]. An effective classification algorithm or technique can improved the accuracy of malicious code detection [30]. Classification method has been widely used in malicious code analysis especially in measuring the effectiveness of detection for a new or unknown sample of malicious code [31].

In this paper, a trojan horse classification called an Efficient Trojan Classification (ETC) is developed as a part and basis of a new trojan horse detection model, but the model will not be discussed in this paper. The details on how the ETC is developed are explained in this paper. Hopefully this new ETC can be used as a basis model and guidance to produce a system either to detect or protect organization from trojan horse attacks.

This paper is organised as follows. Section 2 presents the related works with trojan horse detection techniques, classification and architecture. Section 3 explains the methodology used in this research paper which consists of static and dynamic analyses and the architecture of the controlled laboratory environment. Section 4 presents the research findings which consists of a new trojan horse classification called Efficient Trojan Classification (ETC) and section 5 discusses the testing and evaluation of the proposed trojan horse classification. Section 5 concludes and summarises the future work of this research paper.

2. Related Works

Currently trojan horse attacks is considered as one of the most serious threats in cyber attacks. There are many definitions related with trojan horse such as by [3, 14]. For this research, trojan horse is defined as a program that appears as a useful and harmless, and once it has been installed in a victim computer, it begins to carry out malicious acts such as stealing important information from victim's computer. Apart from that, the victim's computer can be controlled remotely.

Though the trojan horse study was started by [16], only after 10 years later, more studies were carried out such as by [9],[12],[15],[11]. However, these works more focusing on trojan horse hardware taxonomy and hardware detection techniques instead. Each of these works has it owns strengths and gaps that can be further improved. Zhang *et al.* used timestamp-based data stream clustering algorithm to detect trojan horse theft activity [19]. The researchers used clusters to compress trojan horse communication data stream information and extracted clusters characteristics for the detection processes. Based on the experiment conducted, it produced 90% an accuracy rate and lower false negative rate. However this work is only focusing on Trojan horse with theft capability.

Apart from that, Tang presented a new trojan horse detecting method, based on Portable Executable (PE) file static attributes [20]. An intelligent information processing technique is used to analyze those static attributes in the PE files. The experiment result showed the test pass rate is 63.90%. The result can be further improved if the experiment involves bigger volume of dataset.

While Liu *et al.,* used data mining to detect the trojan horse in Windows environment [21]. This study shows that the accuracy of classification can be increased when the more relevant features are used in the data mining processes and reduces the consumption of time space. However, the more features are selected, the more time building classification cost, it responds slower in real time and it needs bigger dataset from real network environment. As for work by Dai *et al,* they presented a novel malicious code detection approach by mining dynamic instruction sequences [21]. Their result showed that their approach is accurate, reliable and efficient. But they used dynamic analyses only and when conducting their experiments, the method was not able to detect any malicious code hooked in the remaining part of the executable code. Improvement can be done if their experiment combining both static and dynamic analysis.

Based on all the previous works discussed above, the main challenges which should be considered thoroughly are the dataset types and volume, analysis and detection techniques and feature selection to detect the trojan horse efficiently. Therefore, in this research, a new trojan classification is developed by integrating static and dynamic analyses and by using bigger and standard dataset, which is further explained in Section 3 and Section 4.

3. Methodology

In order to produce a new trojan horse classification, the researchers' had conducted few experiments and researches. A controlled laboratory environment is created to conduct the experiment. The laboratory for this experiment as illustrated in Fig. 2 and Fig. 3. It is a controlled laboratory environment and almost 80% of the software used in this testing is an open source or available on a free basis. No outgoing network connection is allowed for this architecture.

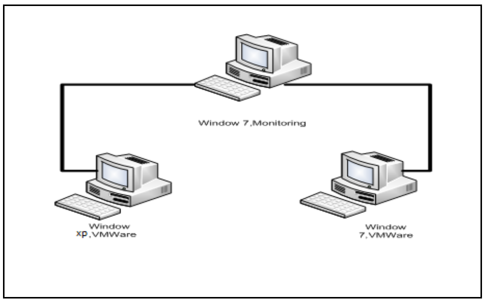


Fig.2 ETC controlled laboratory architecture

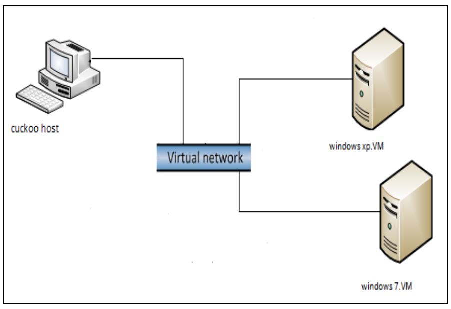


Fig. 3 Cuckoo sandbox architecture

Loading specimen: Before loading the specimen into the laboratory, the entire checklist for the analysis must be checked thoroughly. Once the preparation was done, the trojan horse datasets were loaded into the testing computer using USB memory device. In this lab, the datasets from VXHeavens were tested and analyzed. There are several reasons why this study chose to gather datasets from the VXHeavens source. Firstly, many studies have used this data for their testing. For examples those conducted by [1],[22-26]. The second reason is because the variants are more important than the quantity of the datasets, since this has already represents different types of trojan horse in VX Heavens and the third is due to the scope of this research, which only focuses on Windows platform. Lastly, it is one of largest Trojan databases freely available from the Internet. A total amount of 1640 trojan horse datasets have been tested in this lab.

Trojan horse analysis process: The analysis techniques can be divided into two techniques, which are the static and dynamic analyses. To determine the capabilities of these trojan horses, these two techniques were used in this research lab. The automated analysis, which is part of the dynamic analysis was conducted using the cuckoo open source software [27]. The architecture of the cuckoo can be referred in Fig. 3. All analysis of the trojan horse, were documented and recorded properly. This record is useful in understanding on how the trojan horse works. The detailed of the static and dynamic analyses, as the follows:

3.1 Static Analysis

The mechanism of the static analysis is by looking at the files associated with the trojan horse in the computer without running the program.

Anti-virus check: Once the dataset has been loaded into the testing computers, the file type or compression type is identified. Then, the anti-virus that has been installed inside the testing computers is run. It is used to check if the anti-virus installed can detect anything. If the anti-virus detected the trojan horse, the name of the trojan horse is checked and searched in the anti-virus website for further information.

String analysis: String tool called Strings.exe (from Sysinternal) is used to extract strings from the trojan horse codes. This is helpful in identifying the trojan horse characteristics based on the information retrieved from the strings. Examples of the strings found during the analysis are: trojan horse specimen's name, user dialog, password for backdoors, URLs associated with the codes, email address of the attacker, help or command-line options, libraries, function calls and other executables used by the trojan horse.

Looking for script: Based on the strings extracted from the trojan horse codes, the common scripting or programming languages have been identified as displayed in Table 1.

Disassemble code: Disassemble and debugger which are called as OllyDbg and Ida Pro, were used to transfer a raw binary executable into assembly language and to disassemble and debug the codes for further analysis.

Table 1: Identified common scripting languages

Scripting Language	Identifying Characteristic Inside the File	File's Common Suffix
Bourne Shell Scripting Language	Starts with the line !#/bin/sh	.sh
Perl	Starts with the line !#/usr/bin/perl	.pl, .perl
JavaScript	Includes the word javascript or JavaScript, especially in the form <Script language = "JavaScript">	.js, .html, .htm
Visual Basic Script (VBScript)	Includes the word VBScript, or the characters vb scattered throughout the file	.vbs, .html, .htm
C++	Can be standalone program or many files referenced within the language	.cpp
Active Server Page(ASP)	Can be built using Visual Basic, Jscript or Perl. Can combine HTML, scripts, Active-X server components.	.asp

3.2 Dynamic Analysis

Dynamic analysis includes executing the trojan horse and observing its actions. The trojan horse is activated in a controlled laboratory environment.

Monitoring file activities: Most trojan horse reads from or writes to the file system. It might try to write files, altering existed programs, adding new files or append itself to the file system. By using tool such as Filemon, all actions associated with opening, reading, writing, closing and deleting files can be monitored.

Monitoring process: Prcview v3.7.3.1 is a tool that is used to monitor any running program, files, registry keys and all of the DLLs in the victim's computer. For each running processes, this tool displayed its owner, personal permission, priority and its environment variables.

Monitoring network activities and registry access: Wireshark is used to sniff the network traffic and Nessus is used to monitor the listening ports. Promiscdetect.exe tool is used to determine if the victim computer in broadcast mode state of the interface. The registry needs to be monitored as it contains all the configuration of the operating system and programs installed in the computer. The registry access is monitored by using the Regmon.

Automatic analysis (malware sandbox): Sandbox is a mechanism to analyze the untrusted files or program in a system. It uses dynamical analysis approach and as an alternative of statically analyze for the binary file. It is an open source, an automated malware analysis system. The sandbox automatically run and analyze files and produces analysis results that outline what the malware does while running inside an isolated Windows operating system. The result report displays the traces of win32 API calls performed by all processes spawned by the malware, files created, deleted and downloaded by the malware during its execution, memory dumps of the malware processes, network traffic trace in PCAP format, screenshots of Windows desktop taken during the execution of the malware and full memory dumps of the testing computer.

Referring to Fig. 3, this isolated and virtual architecture consists of a host which is installed with linux (Ubuntu). It is used for guest and analysis management, analyzing, capturing dump traffic and generating reports. While another two virtual computers were setup as a guest and installed with Windows XP Professional and Windows 7 Professional. These 2 computers were used to run and analyze trojan horse files. Later, the analysis report is sent to cuckoo host to be analyzed.

4. Findings

Based on the experiment conducted, a new trojan horse classification called Efficient Trojan Horse Classification (ETC) is developed. This classification is consists of: infection, activation, payload and operating algorithm. The ETC classification is displayed in Fig. 4.

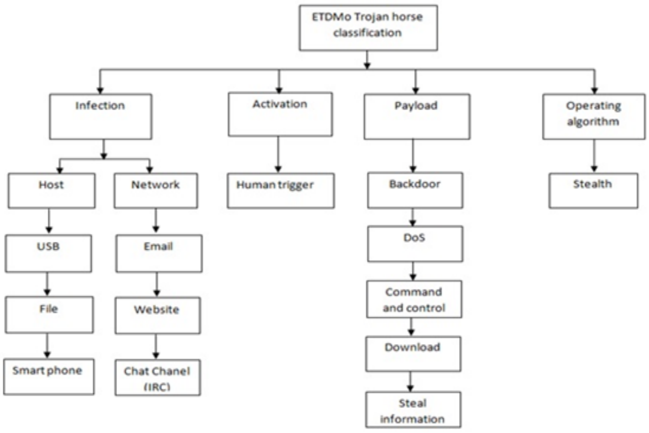


Fig. 4 ETC Classification

4.1 Infection

This term refers on how a computer becomes infected by a trojan horse[1]. The trojan horse infects the victim's computer via two ways, which are via host or network. Examples of the hosts are USB, file and smart phone. These are the most common hosts available today. As for the infection via network, the victim's computer can be infected by executing the attachment that was enclosed in the received email or by downloading and executing file from untrusted website. Apart from that, chatting channels such as Internet Relay Chat (IRC), facebook messenger and Yahoo messenger are examples of the communication channels that can be used and exploited to spread the trojan horse. By clicking the link or URL and executing file that was sent via chatting channel especially from unknown friend, it can exposed the end user of being infected by the trojan horse. Indeed the communication channel can also be used to control victim's computer remotely. To certain extend, the attacker is able to own a large number of computers which have been infected with IRC Trojan Horse and controlled them through IRC channel [1][5][3].

4.2 Activation

Trojan horse activation refers to the criteria that caused the trojan horse to become active and carry out its disruptive function [7]. For this research, based on the experiment conducted, it showed that the trojan horse can only be activated by human trigger. Once the trojan horse file has been dropped in the infected computer, it needs to be executed. Prior of the execution, there are many ways how the trojan horse can dropped itself to the victim's computer, as already explained under section 4.1.

4.3 Payload

Based on the experiment conducted, payload has been identified as one of the most important features in classifying trojan horse. Payload is defined as a destructive mechanism and is designed with malicious intention [1]. Apart from that, it might lead loss of confidential information, doubt of the information integrity (information that is sent and received by the victim) and might damage the victim's computer and caused the loss of the internet access (the availability of the computer and infrastructure). Five main destructive mechanisms have been identified which are: install backdoor, denial of service (DoS), command and control, steal confidential information and download. The details as the following:

Download: Download refers to the capability of a trojan horse to download and install executable files, software and other malware packages chosen by the attacker on the victim's computer. Once the trojan horse with the payload of download is installed, it will download any malicious files from the internet and executed it in the victim's computer without the victim's consent. Examples of the trojan horses are Trojan Horse Downloader and Trojan Horse Dropper [2][3].

Installing backdoor: In this research, installing backdoor is referred as an undercover or illegal way of entering into a computer system. It provides remote control via a LAN or the Internet. Backdoor conducts its function in the same way as legal program used by the system administrator and it is hard to be detected. Examples of the trojan horses are Rootkits and Backdoor Trojan Horse [1][3].

Denial of Service (DoS): One of the trojan horse payload capabilities is to cause Denial of Service (DoS) to the dedicated computer server or service such as web server. As a result, user cannot access the service, website or a system due to the huge of illegitimate network traffic being sent to the targeted victim's server or computer. An example of a trojan horse with DoS capability is Trojan DDoS [8][18]. Apart from causing the DoS, it also caused distributed DoS (DDoS).

Steal confidential information: Identity theft has become as one of the biggest issues in cyber attacks for the past few years. Trojan horse has the capability to do identity theft such as collecting user's username and password, operating system information, local IP address, email address and capturing credit card number. The stolen information stored in the victim computer, can be sent to a remote location where it can be accessed by the creator of of the trojan horse. Examples of the trojan horses are: PSW PSW Trojan horse, Trojan Horse-IM, Trojan Horse-Banker and Infostealer [2][4][6]. On November 2012, Pixsteal-Trojan is an example of a trojan horse who steals images files stored in victim's computer [27]. Once this trojan horse has infected the victim's computer, it copies images in the victim's computer and locate it at C drive. Then it connects to a remote FTP server and transfer the first 20,000 files. What will happen if the image trasmitted trasmitted consists of bank information such as screen capture of username and password for bank login access?

Command and control: It refers to the capability of a trojan horse to send confidential information, such as usernames and passwords, from the infected computer to the trojan horse's creator via the Internet. This allows the trojan horse's creator to control remotely any infected computer [1][2].

4.4 Operating algorithm

Operating algorithm refers to the technique used to avoid malicious code detection [1]. The trojan horse hides it codes from being detected by the anti-virus or anti-trojan software. At end user level, this type of trojan horse hides its activities from user by camouflaging itself as a legitimate program or file and consumes small percentage of system resources to avoid any suspicious [4].

5. Testing

To verify and validate the proposed trojan classification which is called as an Efficient Trojan classification (ETC), the results reports from the static and dynamic analyses is compared and verified with the cuckoo analysis results reports. Both of the static and dynamic analyses were conducted manually and were carried out before the automated analysis was conducted using the cuckoo software. In this research paper, a case study which represents on how the whole proposed ETC is evaluated, is discussed in detail under this section. The architecture of this case study as displayed in Fig. 2 and Fig. 3, where the controlled lab environment and cuckoo's architecture were used in this testing. Based on the cuckoo analysis result report, the sample has been diagnosed as Trojan-DDoS.Win32.Boxed.a. When analyzing the report, the most important part from the whole results are further discussed as the following:

Referring to the import library (*Library KERNEL32.DLL and Library ADVAPI32.dll),* the drop files, Registry Keys, Processes were created and executed (refer to Fig. 5 and Fig. 6. The Fig. 6 is highlighted with red colour), it showed showed that once this trojan executed itself, each time the computer system boots up, it created a service named *Secure Transaction Provider*. This service launches five threads, each of the threads sends TCP packets at high frequency, with SYN flags set. This has caused the network network becoming very slow due to the attacks sent. By the the time the analysis conducted the servers attacks to a certain domain has been shut down.

Based on the results (refer to Fig. 6), using the ETC trojan horse classification as a basis and guidance when doing the analysis, it can be concluded that this trojan horse has a payload of Distributed Denial of Service (DDoS). One of the ETC classifications consists of payload characteristic with five subcategories, which are: backdoor, DoS, command control, download and steal information.

Based on the result analysis, following the right method in analyzing the trojan horse and having an effective trojan horse classification, made the analysis job easier. By end of the analysis, the researcher, security analyst or virus analyst has to make her own conclusion based on the static and dynamic analyses. Therefore, with the method introduced in this paper together with the proposed ETC, have help the security analyst to do the analysis faster than the traditional way.

1) Static Analysis Results

Import

***Library KERNEL32.DLL**:*
- *0x409020 - SetThreadPriority*
- *0x409024 - ResumeThread*
- *0x409028 - LoadLibraryA*
- *0x40902c - FreeLibrary*
- *0x409030 - GetProcAddress*
- *0x409034 - GetModuleFileNameA*
- *0x409038 - CreateThread*
- *0x40903c - Sleep*
- *0x409040 - SetEnvironmentVariableA*
- *0x409044 - CompareStringW*
- *0x409048 - CompareStringA*
- *0x40904c - FlushFileBuffers*
- *0x409050 - GetStringTypeW*
- *0x409054 - GetStringTypeA*
- *0x409058 - LCMapStringW*
- *0x40905c - LCMapStringA*
- *0x409060 - MultiByteToWideChar*
- *0x409064 - GetTimeZoneInformation*
- *0x409068 - GetSystemTime*
- *0x40906c - GetLocalTime*
- *0x409070 - HeapFree*
- *0x409074 - HeapAlloc*
- *0x409078 - RtlUnwind*
- *0x40907c - GetModuleHandleA*
- *0x409080 - GetStartupInfoA*
- *0x409084 - GetCommandLineA*
- *0x409088 - GetVersion*
- *0x40908c - ExitProcess*
- *0x409090 - GetEnvironmentVariableA*
- *0x409094 - GetVersionExA*
- *0x409098 - HeapDestroy*
- *0x40909c - HeapCreate*
- *0x4090a0 - VirtualFree*
- *0x4090a4 - VirtualAlloc*
- *0x4090a8 - HeapReAlloc*
- *0x4090ac - IsBadWritePtr*
- *0x4090b0 - TerminateProcess*
- *0x4090b4 - GetCurrentProcess*
- *0x4090b8 - UnhandledExceptionFilter*
- *0x4090bc - FreeEnvironmentStringsA*
- *0x4090c0 - FreeEnvironmentStringsW*
- *0x4090c4 - WideCharToMultiByte*
- *0x4090c8 - GetEnvironmentStrings*
- *0x4090cc - GetEnvironmentStringsW*
- *0x4090d0 - SetHandleCount*
- *0x4090d4 - GetStdHandle*
- *0x4090d8 - GetFileType*
- *0x4090dc - WriteFile*
- *0x4090e0 - GetLastError*
- *0x4090e4 - SetFilePointer*
- *0x4090e8 - SetUnhandledExceptionFilter*
- *0x4090ec - IsBadReadPtr*
- *0x4090f0 - IsBadCodePtr*
- *0x4090f4 - GetCPInfo*
- *0x4090f8 - GetACP*
- *0x4090fc - GetOEMCP*
- *0x409100 - SetStdHandle*
- *0x409104 - CloseHandle*

Library ADVAPI32.dll:
- *0x409000 - CreateServiceA*
- *0x409004 - ChangeServiceConfig2A*
- *0x409008 - StartServiceA*
- *0x40900c - RegisterServiceCtrlHandlerA*
- *0x409010 - SetServiceStatus*
- *0x409014 - StartServiceCtrlDispatcherA*
- *0x409018 – OpenSCManagerA*

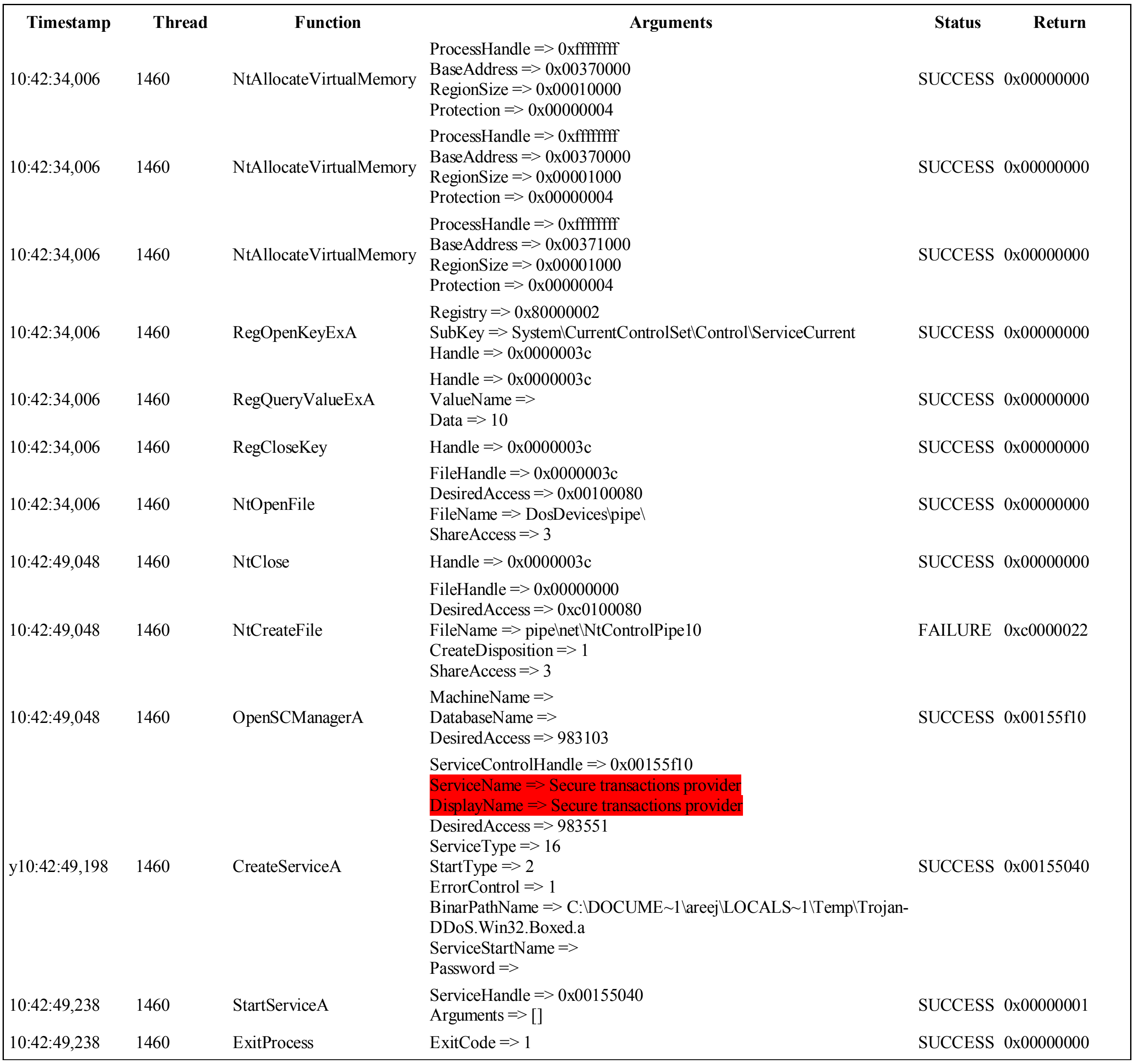
2) Behaviour Analysis Results

Drop Files:
- DosDevices\pipe\
- *pipe\net\NtControlPipe10*

Registry Keys
- *HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\ServiceCurrent*

Processes
Refer to Fig. 6

Fig. 5 Results of static and behavior analyses



Timestamp	Thread	Function	Arguments	Status	Return
10:42:34,006	1460	NtAllocateVirtualMemory	ProcessHandle => 0xffffffff BaseAddress => 0x00370000 RegionSize => 0x00010000 Protection => 0x00000004	SUCCESS	0x00000000
10:42:34,006	1460	NtAllocateVirtualMemory	ProcessHandle => 0xffffffff BaseAddress => 0x00370000 RegionSize => 0x00001000 Protection => 0x00000004	SUCCESS	0x00000000
10:42:34,006	1460	NtAllocateVirtualMemory	ProcessHandle => 0xffffffff BaseAddress => 0x00371000 RegionSize => 0x00001000 Protection => 0x00000004	SUCCESS	0x00000000
10:42:34,006	1460	RegOpenKeyExA	Registry => 0x80000002 SubKey => System\CurrentControlSet\Control\ServiceCurrent Handle => 0x0000003c	SUCCESS	0x00000000
10:42:34,006	1460	RegQueryValueExA	Handle => 0x0000003c ValueName => Data => 10	SUCCESS	0x00000000
10:42:34,006	1460	RegCloseKey	Handle => 0x0000003c	SUCCESS	0x00000000
10:42:34,006	1460	NtOpenFile	FileHandle => 0x0000003c DesiredAccess => 0x00100080 FileName => DosDevices\pipe\ ShareAccess => 3	SUCCESS	0x00000000
10:42:49,048	1460	NtClose	Handle => 0x0000003c	SUCCESS	0x00000000
10:42:49,048	1460	NtCreateFile	FileHandle => 0x00000000 DesiredAccess => 0xc0100080 FileName => pipe\net\NtControlPipe10 CreateDisposition => 1 ShareAccess => 3	FAILURE	0xc0000022
10:42:49,048	1460	OpenSCManagerA	MachineName => DatabaseName => DesiredAccess => 983103	SUCCESS	0x00155f10
y10:42:49,198	1460	CreateServiceA	ServiceControlHandle => 0x00155f10 ServiceName => Secure transactions provider DisplayName => Secure transactions provider DesiredAccess => 983551 ServiceType => 16 StartType => 2 ErrorControl => 1 BinarPathName => C:\DOCUME~1\areej\LOCALS~1\Temp\Trojan-DDoS.Win32.Boxed.a ServiceStartName => Password =>	SUCCESS	0x00155040
10:42:49,238	1460	StartServiceA	ServiceHandle => 0x00155040 Arguments => []	SUCCESS	0x00000001
10:42:49,238	1460	ExitProcess	ExitCode => 1	SUCCESS	0x00000000

Fig. 6 Processes involved in behavior analysis

6. Conclusions and Future Works

This new classification (ETC) is produced based on the research and testing that have been conducted in the controlled laboratory environment. The classification consists of four main categories: Infection, Activation, Payload and Operating Algorithm. This paper is part of a larger research project to confront the trojan horse attacks. Ongoing research includes producing a trojan horse detection model based on the ETC classification. This ETC classification can be used as a basis for other researchers in the world to build a better malware detection model.

Acknowledgments

The authors would like to express their gratitude to Universiti Sains Islam Malaysia (USIM) for the support and facilities provided. This research paper is supported by Universiti Sains Islam Malaysia (USIM) grant [PPP/GP/FST/SKTS/30/11912] and Research Management Centre, Universiti Sains Islam Malaysia (USIM).

References

[1] Saudi, M.M (2011) . A New Model for Worm Detection and Response (PHD thesis), University of Bradford, United Kingdom.

[2] Gharibi, w. 2011, Studying and Classification of the Most Significant Malicious Software.

[3] Al-Saadoon, G, Al-Bayatti, H, 2011. A Comparison of Trojan horse Virus Behavior in Linux and Windows Operating Systems, World of Computer Science and Information Technology jornal, Vol. (1), No. 3, 56-62.

[4] Thigpen,a. 2010. Infostealer Trojan horse Report, Symantec website. URL: http://www.symantec.com/security_response/writeup.jsp?docid=2003-071710-2826-99&tabid=2,[last accessed: 16th March 2013].

[5] Krishan. K. , Himanshu. U. , Ritesh. K. , 2012. Trojan horse: Infection and Precaution, BPR Technologia : A Journal of Science, Technology & Management, vol (1).

[6] Hilven, a. , Woodward a. , 2007. How safe is Azeroth, or, are MMORPGs a security risk, Australian Information Security Management Conference.

[7] Tehranipoor, h. , Plusquellic, j. ,2009. New Design Strategy for Improving Hardware Trojan horse Detection a Reducing Trojan horse Activation Time, Hardware-Oriented Security and Trust, IEEE.

[8] Liu,W (2009). Research on DoS attack and detection programming. 2009 Third International Symposium on Intelligent Information Technology Application IEEE, vol (1),:207-210.

[9] Chakraborty, R.S., Narasimhan, S. and Bhunia, S. 2009. Hardware trojan horse : threats and emerging solutions, Proceedings IEEE International High Level Design Validation and Test Workshop, San Francisco, CA, 166 - 171.

[10] Hawkins,S., C. Yen,D. and C. Chou,D. 2000. Awareness and challenges of Internet security, Information Management & Computer Security, Vol. (8): 3,131 – 143.

[11] Karri, R., Rajendran,J. and Rosenfeld,K. 2011. Trojan horse taxonomy, In: M. Tehranipoor and C. Wang(eds.), Introduction to Hardware and Security Trust,Springer, pp.325-338.

[12] Karri, R., Rajendran,J.,Rosenfeld,K. and Tehranipoor,M. 2010. Trustworhty hardware: identifying and classifying hardware trojan horse s.Computer 43(10),39-46.

[13] Saudi,M.M.and Jomhari,N.2006. Knowledge structure on virus for user education. 2006. *International Conference on Computational Intelligence and Security*, 1515 - 1518.

[14] Saudi,M.M (2008). User awareness on virus in windows platform, Journal of Information Technology and Multimedia, UKM.

[15] Tehranipoor, M. and Koushanfar, F. 2010. A survey of hardware trojan horse taxonomy and detection, Design & Test of Computers IEEE,Vol(27):1, 10-25.

[16] Thimbleby,H., Anderson,S. and Cairns, P. 1998. A framework for Modelling Trojan horse s and Computer Virus Infection, Computer Journal,Vol(41):7,444-458.

[17] McGraw,K.Hill,G. 2000. Differential effects of endoparasitism on the expression of carotenoid-and melanin-based ornamental coloration, Proceedings of the Royal Society of London, vol (267), 1525-1531.

[18] Anderson, T. Roscoe, T. Wetherall, D. 2004. Preventing Internet denial-of-service with capabilities, ACM SIGCOMM Computer Communication Review, vol(34):39.

[19] Zhang, Xi. Liu, Sh. Meng, L. Shi, Y. 2012. Trojan horse Detection Based on Network Flow Clustering, Multimedia Information Networking and Security conference IEEE: 947-950.

[20] Tang, Sh. 2009. The detection of Trojan horse based on the data mining, Fuzzy Systems and Knowledge Discovery International Conference IEEE, vol (1): 311-314.

[21] Liu,y., Zhang,l. Liang,j. Qu,s. Ni,z. 2010. Detecting Trojan horses based on system behavior using machine learning method, 2010 Machine Learning and Cybernetics conference IEEE, vol (2): 855 – 860.

[22] Dai, J., Guha, R. and Lee,J. 2009. Efficient Virus Detection Using Dynamic Instruction Sequences. *Journal of Computers.* Vol 4, No 5, 405-414.

[23] Schultz, E.E. and Shumway, Russell. (2001). *Incident Response: A Strategic Guide to Handling System and Network Security Breaches*, 1st edn., United States of America: New Riders Publishing.

[24] Henchiri, O. and Japkowicz, N. (2006). A Feature Selection and Evaluation Scheme for Computer Virus Detection. *Proceedings of the Sixth International Conference on Data Mining, 2006. ICDM '06.* Hong Kong: IEEE Xplore, pp. 891.

[25] Moskovitch,R., Stopel,D., Feher,C., Nissim,N., Japkowicz, N. and Elovici,Y. (2008). Unknown malcode detection and the imbalance problem. *Journal In Computer Virology.* Volume 5, Number 4, 295-308, DOI: 10.1007/s11416-009-0122-8.

[26] Khan,H., Mirza,F. and Khayam,S.A. (2010). Determining malicious executable distinguishing attributes and low-complexity detection. Journal In Computer Virology. 7(2), pp. 95-105.

[27] Fahmida Y. Rashid (2012), Image-Stealing Trojan Exposes Victims to ID Theft,Blackmail, URL: http://securitywatch.pcmag.com/none/304678-image-stealing-trojan-exposes-victims-to-id-theft-blackmail, [last accessed: 16th March 2013].

[28] Cuckoo Sandbox Organization, (2012), Cuckoo sandbox, URL: http://docs.cuckoosandbox.org/en/latest/installation/guest/requirements/, [last accessed: 16th March 2013].

[29] Babak, R., Maslin, B., and Suhaimi, I. (2011). Evolution of Computer Virus Concealment and Anti-Virus Techniques: A

Short Survey. *International Journal of Computer Science Issues*, *8*(1).

[30] Nguyen, V. T., Kha, V. V., and Anh, A. P. (2012). Research Some Algorithm in Machine Learning and Artificial Immune System, Apply to Set Up A Virus Detection System. *International Journal of Computer Science Issues*, *9*(4).

[30] Karbalaie, F., Sami, A and Ahmadi, M**.** (2012). Semantic Malware Detection by Deploying Graph Mining. *International Journal of Computer Science Issues*, *9*(1).

Areej Mustafa Abuzaid is currently a Master candidate in Information and Security (ISA) Programme at Faculty Science and Technology (FST), Universiti Sains Islam Malaysia (USIM).

Madihah Mohd Saudi is a senior lecturer at the Faculty Science and Technology (FST), Universiti Sains Islam Malaysia (USIM). She graduated her Bachelor degree in Computer Science from Universiti Kebangsaan Malaysia(UKM), then obtained her Master degree in Software Engineering from Universiti Malaya (UM), Malaysia and her PhD degree in Computer Security from University of Bradford, United Kingdom. She is a senior member of IEEE and IACSIT and a member of SDWIC and IAENG. Her current research interests include malware detection and response, incident response, network security, computer forensics, social engineering and machine learning. She has published numerous papers on the above and related topics. She is the corresponding author for this research paper.

Bachok M Taib is a professor at the Faculty Science and Technology (FST), Universiti Sains Islam Malaysia (USIM). He graduated his Bachelor degree in Mathematics from Universiti Kebangsaan Malaysia (UKM), obtained his Master degree in Numerical Methods from University of Reading, United Kingdom and his PhD in Numerical Fluid Mechanics from Wollongong University, New South Wales, Australia. He is a member of Persatuan Sains Matematik Malaysia (PERSAMA). His research interests are Numerical Computation and Mathematical Modelling.

Zul Hilmi Abdullah is an academician at the Faculty of Science and Technology (FST), Universiti Sains Islam Malaysia (USIM). He obtained his Bachelor degree of Computer Science from Universiti Putra Malaysia (UPM) and his Master of Information Security from Universiti Teknologi Malaysia (UTM). His research interests include information security and expert systems.