

Forensic Analysis of a Botnet System: Architecture and Capabilities

Sultan Bandr Almutairi

a thesis submitted to the faculty of design and creative technologies
Auckland University of Technology
in partial fulfilment of the
requirements for the degree of
Masters of Forensic Information Technology

School of Computing and Mathematical Sciences

Auckland, New Zealand
2014

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the qualification of any other degree or diploma of a University or other institution of higher learning, except where due acknowledgement is made in the acknowledgements.

.....
Sultan Bandr Almutairi

Acknowledgements

First and foremost I would like to thank Allah for making this possible to achieve. This thesis completed at the Faculty of Design and Creative Technologies in the school of Computing and Mathematical Sciences at Auckland University of Technology. I would like to express my sincere gratitude to everyone that has supported me throughout the last 2 years of completing my thesis. First, I would like to thank my father Bandr, my mother Aiyzah, my brothers and sisters for the amazing support throughout my study. In addition, I would like to thank my best friend Laura for the incredible support throughout my MFIT program. Also I would like to thank Masaed for the support and the guidance throughout my thesis.

Likewise, I would like to thank my thesis supervisor, Dr. Brian Cusack who has provided me with valuable advice on my thesis project and Dr, Brian Cusack whom helped me to achieve so much this far. Also I would like to thank Dr Alastair Nisbet who helped me get through my postgraduate year with a valuable advice.

In addition, I would like to thank my course mates for the memorable moments that we shared in the classes. I would like to thank Muteb and Scottie for sharing their ideas with me that helped me throughout my thesis. Furthermore, The English language is my second language; therefore, I would like to thank all the people who have provided proof reading advice to help the presentation of this thesis.

In addition, I would like to thank all the staff of the Saudi Mission Culture for supporting me throughout my studies. I would like to send special thanks to Dr Sattam Alotaiby and Dr Samir Aljabri for providing the support to all the Saudi Students in New Zealand.

Abstract

The botnet is one of the biggest threats to computer machines and systems. The main challenge of the botnet is that this type of malware has developed to avoid detection. Many of the computer users use anti-virus tools that do not detect the botnet existence in the computer system. The botnet infects a computer then connects the computer to the command and control server to join. The botnet runs in the background and communicates with the (C&C) server to receive instruction that typically involves being part of malicious activities performed against other organisations. The malicious activities typically performed without the knowledge of the owner of the computer machine is being part of the malicious activities. The victims of the botnet are usually in the millions of infected hosts.

A secure laboratory environment made this research to be able to examine actions close to a real botnet event. The Dionaea honeypot used to be able to collect the samples of the malware including the botnet samples. Then, the downloaded botnet samples submitted into two external sandbox services to be able to analyse the samples. After that, the samples were analysed by a malware analysis tool to be able to have a clear picture of the botnet malware samples. In addition, the downloaded botnet samples by Dionaea then used to infect the Virtual machine (VM) host in the experiment. Each botnet sample used to infect the host, then, the host formatted to its original status for fresh infects on with other bots downloaded by Dionaea. The focus of this research is to be able to find the possible evidence in the infected host as well as the communication of the host with the C&C server.

The findings from the laboratory experiment show evidence that related to a botnet event. The research was able to locate the evidence of the existence of the botnet in the infected host in the registry, file system, network and the physical memory of the infected host. The research found that there were a large number of changes, which have performed to the infected host. The research was also able to find that the infected host was communicating with the suspicious C&C server. The infected host connects to the suspicious C&C straightaway after the infection of the bot sample. The infected host by the IRC bot was requesting more than 200 domain names and IP addresses within a short period of the infection of the bot.

The sniffer tools were able to show the domain names and the IP addresses that have requested by the infected host. The research was able to find the instructions sent to and from the suspicious C&C server. The research was able to find that the instructions of the IRC bot usually sent in a plain text using the TCP protocol. However, the checking of the status of the bot in the infected host performed by using the ICMP checked channel that encrypted.

The research recommendations discuss the cross-border-issues as one of the challenges that stop the international effort to track down the botnet master. The botnet master is difficult to locate due to the complexity of the techniques they use to hide their location. Furthermore, the detection of the botnet needs to be improved as the current detection techniques of a botnet are still evolving. However, this research recommends that in order to shut down the C&C server future work should also consider the destruction of the C&C server. The contribution of this research is on better understanding of the C&C communicating and hence evidence that can be used to disrupt a botnet.

Table of Content

Declaration	i
Acknowledgements	ii
Abstract	iii
Table of Contents	v
List of Tables.....	ix
List of Figures	x

Chapter 1: Introduction

1.0 BACKGROUND.....	1
1.1 MOTIVATIONS	3
1.2 THE RESEARCH APPROACH	5
1.3 THE RESEARCH FINDINGS.....	6
1.4 STRUCTURE OF THESIS	6

Chapter 2: Literature Review

2.0 INTRODUCTION.....	9
2.1 DEFINITION OF BOTNETS.....	10
2.1.1 Definition of Term.....	10
2.1.2 Botnets in Cybercrime	11
2.1.3 Botnets Feature	12
2.1.4 Building Of Botnets And Its Lifecycle.....	16
2.1.5 Brief History Of The Botnets	18
2.2 BOTNETS ARCHITECTURE	21
2.2.1 Centralized C&C	21
2.2.2 IRC Internet	22
2.2.3 HTTP	23
2.2.4 P2P.....	24
2.2.4 DNS	24
2.3 BOTNETS COMPONENTS	25
2.3.1 Botnet Master	25
2.3.2 Command And Control (C&C) Channel.....	26
2.3.3 Bot	28
2.3.4 Victim.....	29

2.4	BOTNETS ACTIONS	29
2.4.1	Propagation.....	29
2.4.2	Bot Terminated Process.....	32
2.4.3	Compromised Machines.....	33
2.5	BOTNETS COLLECTION AND ANALYSIS	34
2.5.1	Collecting Botnets	35
2.5.2	Detecting Botnets	36
2.5.3	Honeypot	37
2.5.4	Analysis Of Botnets Malware	38
2.5.5	Live And Static Forensic	41
2.6	REVIEW OF ISSUES AND PROBLEMS.....	42
2.7	CONCLUSION	44

Chapter 3: Research Methodology

3.0	INTRODUCTION	45
3.1	REVIEW OF SIMILAR RESEARCH.....	45
3.1.1	Visualization Of Invariant Bot Behaviour.....	46
3.1.2	Real-Time Botnets Command And Control Characterization At The Host Level	47
3.1.3	Collaborative Architecture For Malware Detection And Analysis	49
3.1.4	Insight From The Analysis Of The Mariposa Botnets	51
3.2	RESEARCH DESIGN	52
3.2.1	Summary Of Similar Studies.....	53
3.2.2	Review Of Problem Areas.....	54
3.2.3	The Research Question And Hypotheses	54
3.2.4	Research Phases.....	56
3.2.5	Data Map	57
3.3	DATA REQUIREMENTS.....	59
3.3.1	Data Type	59
3.3.1.1	Malware Signature	59
3.3.1.2	Digital Evidence	60
3.3.2	Data Collection.....	60
3.3.2.1	Laboratory Environment.....	60
3.3.2.2	Laboratory Component.....	61
3.3.3	Data Processing	62
3.3.4	Data Analysis.....	63
3.3.4.1	Memory Analysis	64
3.3.4.2	Static Analysis	64

3.3.4.3 Analysis Tools	64
3.4 LIMITATIONS	65
3.5 CONCLUSION	66

Chapter 4: Research Findings

4.0 INTRODUCTION	68
4.1 VARIATIONS ENCOUNTERED IN THE EXPERIMENT	68
4.1.1 Data Collection	68
4.1.2 Data Processing	69
4.1.3 Data Analysis And Presentation	69
4.2 MALWARE COLLECTION AND THE ANALYSIS OF THE MALWARE	70
4.2.1 Low Interaction Honeypot (Dionaea)	70
4.2.2 Threat Expert	74
4.2.3 Anubis.....	82
4.2.4 Wireshark.....	88
4.2.3 Live Monitoring.....	94
4.3 ANALYSIS	97
4.3.1 Propagation	98
4.3.2 Infection.....	100
4.3.3 Connecting To The Botnets Server	100
4.3.4 Summary Of The Analysis	100
4.4 CONCLUSION	102

Chapter 5: Research Discussion

5.0 INTRODUCTION	104
5.1 ANSWERING THE RESEARCH QUESTION	104
5.1.1 Sub-Question Answers	105
5.1.2 Hypothesis Tests.....	111
5.1.3 The Research Question Answer	116
5.2 DISCUSSION	119
5.2.1 Discussion Of The Infected Host Environment.....	119
5.2.2 Discussion On Data Acquisition And Extraction From The Infected Host	121
5.2.3 Discussion On Reconstruction & Analysis	121
5.2.4 Command And Control Communication.....	123
5.2.5 Recommendation On Tracking Botnets	124
5.2.5.1 Cross Border Issues.....	124

5.2.5.2	Tracking The Botnet Master.....	125
5.2.5.3	Improving The Detection Of The Botnets.....	125
5.3	CONCLUSION	126

Chapter 6: Conclusion

6.0	CONCLUSION	127
6.1	LIMITATIONS OF RESEARCH	128
6.2	FUTURE RESEARCH.....	129

REFERENCES	131
------------------	-----

Appendices

Appendix 1	137
Appendix 2	140
Appendix 3	142
Appendix 4	150
Appendix 5	151
Appendix 6	152
Appendix 7	153

List of Tables

Table 3.1 Analysis Tools.....	65
Table 4.1 The Scan Result Of The IRC Bot From ThreatExpert.....	78
Table 4.2 Shellcode Downloaded By Dionaea Honeypot.....	98
Table 4.3 Summary Of The Analysis.....	101
Table 5.1 Sub-Question 1 And Answer.....	105
Table 5.2 Sub-Question 2 And Answer.....	106
Table 5.3 Sub-Question 3 And Answer.....	106
Table 5.4 Sub-Question 4 And Answer.....	107
Table 5.5 Sub-Question 5 And Answer.....	108
Table 5.6 Sub-Question 6 And Answer.....	108
Table 5.7 Sub-Question 7 And Answer.....	109
Table 5.8 Sub-Question 8 And Answer.....	110
Table 5.9 Sub-Question 9 And Answer.....	110
Table 5.10 Tested Hypothesis 1	111
Table 5.11 Tested Hypothesis 2	112
Table 5.12 Tested Hypothesis 3	114
Table 5.13 Research Main Question and Tested Hypothesis	116

List of Figures

Figure 1.1 The Distribution of TDL4-infected computers by country.....	3
Figure 1.2 Number of publications on botnetss per year	4
Figure 2.1: The official report from the New Zealand National Cyber Security Centre (NCSC) that shows the target for the cybercrime in 2012 in New Zealand..	12
Figure 2.2 The infection phase	14
Figure 2.3 The attack phase.....	15
Figure 2.4: Building a botnets. The 5 phases that have been mentioned in this section in details	17
Figure 2.5 The evaluation of the botnets	19
Figure 2.6 The configuration option for the IRC	27
Figure 2.7 The configuration option for the HTTP	27
Figure 2.8 The Known Vulnerabilities Commonly Exploited by Rbot Variants.	30
Figure 2.9 Vulnerabilities Exploited by Spybot Variants to Help it Propagate ...	31
Figure 2.10 The File Extensions Known to Be Commonly Targeted by Mytob for Harvesting E-mail Addresses	31
Figure 2.11 Mytob Eliminates Harvested E-mail Addresses with the Following Domains.....	32
Figure 2.12 A Sample of Processes Sometimes Terminated by RBot	32
Figure 2.13 A Known Filenames Used by Backdoor for SDBot	33
Figure 2.14 Cumulative distribution function (CDF) duration	39
Figure 2.15 The spam e-mail enticing the victim to click and become infected..	40
Figure 2.16 Malicious encoded JavaScript code to lead the victim to download ecard.exe.....	40
Figure 2.17 The decode JavaScript with shell-code and instruction to install ecard.exe.....	41
Figure 3.1 Three phases of botnets life-cycle considered for Invariant bot behaviour identification.....	46
Figure 3.2 Architecture overview of our proposed approach.....	48
Figure 3.3 Cooperative architecture for malware detection and analysis	50
Figure 3.4 Overview of Mariposa Bot.....	51
Figure 3.5 Research Phases	57

Figure 3.6 Proposed Research Data Map	58
Figure 3.6 Laboratory Component	61
Figure 4.1 The scan result of the Rbot from Virustotal.....	72
Figure 4.2 The scan result of the IRC bot from ThreatExpert	75
Figure 4.3 The scan result of the IRC bot from ThreatExpert	76
Figure 4.4 The scan result of the IRC bot from ThreatExpert	77
Figure 4.5 The scan result of the IRC bot from ThreatExpert	77
Figure 4.6 The scan result of the IRC bot from ThreatExpert	81
Figure 4.7 The scan result of the IRC bot from Anubis	82
Figure 4.8 The scan result of the IRC bot Dependency from Anubis	84
Figure 4.9 The list of loaded libraries from Anubis	85
Figure 4.10 The list of loaded libraries	86
Figure 4.11 The list of the Registry affected by the IRC bot	87
Figure 4.12 The list of files affected by the IRC bot.....	88
Figure 4.13 The scan result of the IRC bot from Anubis	89
Figure 4.14 The DNS indicates of a suspicious IRC bot server from Wireshark	89
Figure 4.15 The search result of the IRC bot from Windows	89
Figure 4.16 The TCP Traffic of the suspicious IRC bot from Wireshark.....	90
Figure 4.17 The DNS indicates of a suspicious IRC bot server from Wireshark	90
Figure 4.18 The checking statues of the bot from Wireshark	90
Figure 4.19 The DNS result of the IRC bot from virusTotal	91
Figure 4.20 The DNS result of the IRC bot from virusTotal	92
Figure 4.21 The running process in the physical memory of the infected host ...	93
Figure 4.22 The DNS requested by an IRC bot a650c67e14cfb27879999036741478d5.....	95
Figure 4.23 The DNS requested by an IRC bot 0a278f8d72e4d3d2d44485764398c84d	96
.....	96
Figure 4.24 Sensitive information sent to the suspicious C&C server.....	96
Figure 4.25 The list of loaded external libraries	97

Chapter 1

Introduction

1.0 BACKGROUND

Malware in general is pre-programed and designed in order perform an activity in an infected machine. The main aim for all types of malware is to disrupt damage or perform activities in the infected machine that is unwanted by the owner of the machine. The “mal” word in Latin means “bad”, this means that by replacing the “mal” with “bad” will make the word as “badware” that harm the computer system. There are many type of malware that area a threat to the computer machines and internet security, for example Viruses, Worms, Trojans, Horses, Spywares and Botnets (TechTerms, n.d.). The botnet is one specfic threat to the computer machine or internet security and is a high level of threat. The botnet is a most significant threat in network security. The botnet is the collection or a large network of compromised computers (Ullah, Khan, & Aboalsamh, 2013) that increases each time a new host has been infected joins. The increasing of the number of the infected hosts creates an army of compromised computers that are called “Zombies”. These compromised computers and systems (infected hosts) are used by the botnet-master to perform suspicious activities on other computer systems (Ullah et al., 2013). In addition, the botnet itself can be called as a network of hosts that are infected by malware (bots) that are controlled by a botnet master (Stone-Gross et al., 2009).

The botnets control the infected host by a bot, which is a program that runs in the infected host to control it remotely without the owner of the machine being aware of it. In order for the bot to infect the machine, the bot finds its way into the machine by using the vulnerabilities of the machine. Most of the malware are designed for the Windows based operating system for many reasons such as known vulnerabilities and the high number of users (Bächer, Holz, Kötter, & Wicherski, n.d.). The purpose of using a botnet is for the bot to create a large network of compromised computers that connected to hosts. The host that is the agent between the bot and the botnet-master is typically a large number of hosts that indicates the high number of possible machines that could be infected by the botnets (Stone-Gross et al., 2009). The botnet usually aims to find vulnerabilities

throughout the internet for more victims by many methods such as scanning the IP addresses, and the broadband users are preferable for the botnets to perform their malicious activities against other computer systems. One of the interesting studies on the botnets shows that more than 4 million computers have been infected by TDL4 botnets in just a 3 month period which indicates that extent of the threat that faces the security vendors to fight this type of malware (Greengard, 2012)

There are many types of malwares that identified by previous researchers. The most popular botnets are IRC botnets, which are the first developed, these followed by HTTP, and P2P designs. The easiest type of botnet to detect is the IRC bot whereas the most difficult one to detect is the P2P bot. The P2P bot is most difficult one to detect, however, the botnets-master has limited control in the design, as the botnets-master cannot locate the entire infected host. In addition, the IRC bot is easier to detect and sniffer tools can sniff the traffic easily. However, it is still one of the favourite bots for attackers as it is easy to set up as well many attackers have been using this type of bot for years and have great experience with it (Bächer, Holz, Kötter, & Wicherski, n.d.).

The bot infects the machine then it connects to a large network of hosts through what is called the command and control (C&C) channels. The botnets C&C server is the agent that receives the instructions from the botnet master and sends it to the bot that are controlling the infected host. The infected hosts is then used as an army to perform an attack to other computer systems such as banks and other organisations (Greengard, 2012). The attractive part for attackers about the C&C server is that when one of C&C server hosts is taken down and blocked, then, the botnet master (attacker) needs to register the new domain list to take back the control of the bots that belong to the C&C server that has been taken down. This means that the bots in the infected hosts can be re-control by the new C&C server without losing the control of these hosts (Stone-Gross et al., 2009).

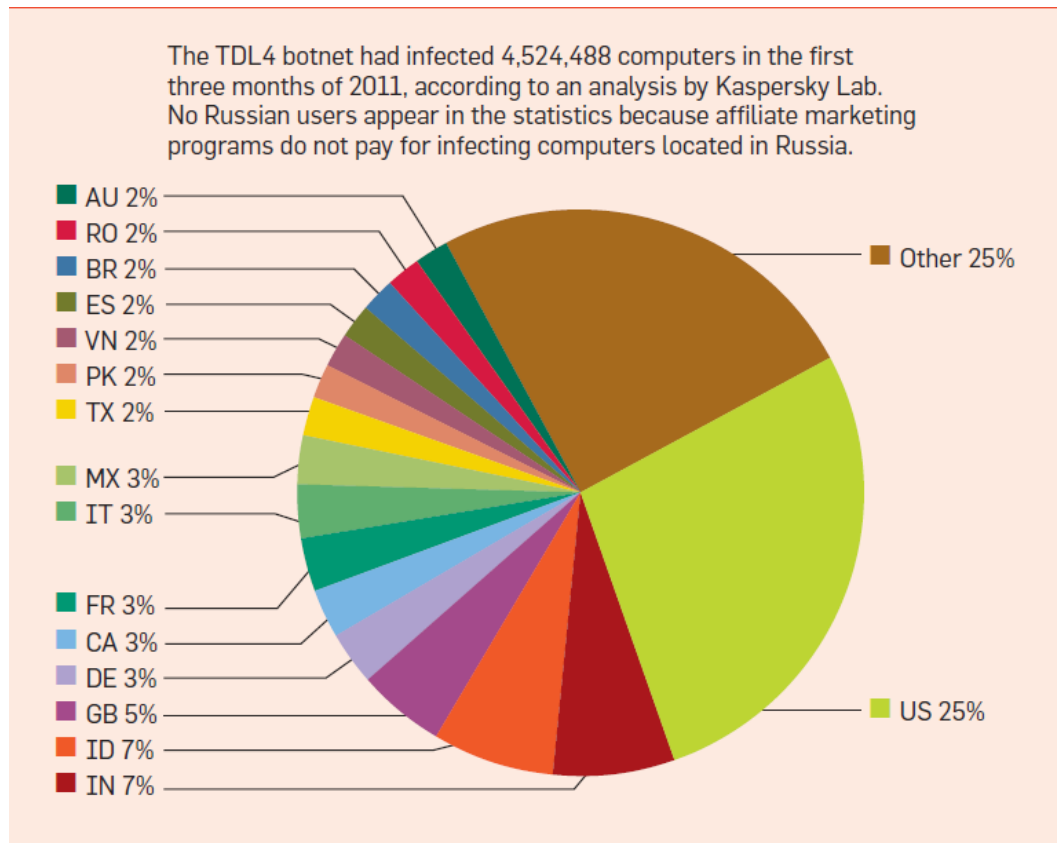


Figure 1.1 The Distribution of TDL4-infected computers by country
(Greengard, 2012, p. 16)

The Figure 1.1 shows the challenges that face the internet security when protecting against botnet threats. It is not a national specific problem but the issue of the is an international issue which the figure 1.1 shows when the TDL4 botnets infected more than 4 million computers around the global. The 4,524,448 computers that were infected by the TDL4 became the army of the TDL4 botnets (Zombies) that perform different types of malicious activities and most of the owners of these infected machines are probably not aware that their machines are infected by the botnets (Greengard, 2012).

1.1 Motivation

The motivation for botnet activities is typically to gain a financial benefit out of these activities for different purposes. A botnets survey shows that the damage from this malicious activities have cost US \$35 billion in 2007. This indicates that the botnet is driven by cybercrime organisations for financial gain. In addition, the majority of the spam email are sent by botnets. This is one of the malicious malware that spreads spam as well as is responsible for DoS and DDoS attacks using the bots (Greengard, 2012). The botnets steal sensitive information from the

infected host and send the sensitive information back to the server, then, this sensitive information is on sold by the attackers. The sensitive information could be personal bank details, private information or other information that an unauthorised person should not have the right to (Stone-Gross et al., 2009).

The botnet event is one of the fastest growing malware threats that faces the security community (Li, Jiang, & Zou, 2009). Surveys show that the botnet events have been growing rapidly since 1993 when the botnets first noticed and detected such as Eggdrop found in December 1993. After that there were many types of botnet that have been detected such as PrettyPark 1999, Agobot 2002, SDbot 2002, SpyBot 2003 and many other botnetss (Li et al., 2009). It is estimated the number of botnet members on the internet are between 16 to 25% of the users (Silva, Silva, Pinto, & Salles, 2013). Figure 1.2 also shows that the number of publications from research is also growing progressively over a ten year sample period (Silva et al., 2013).

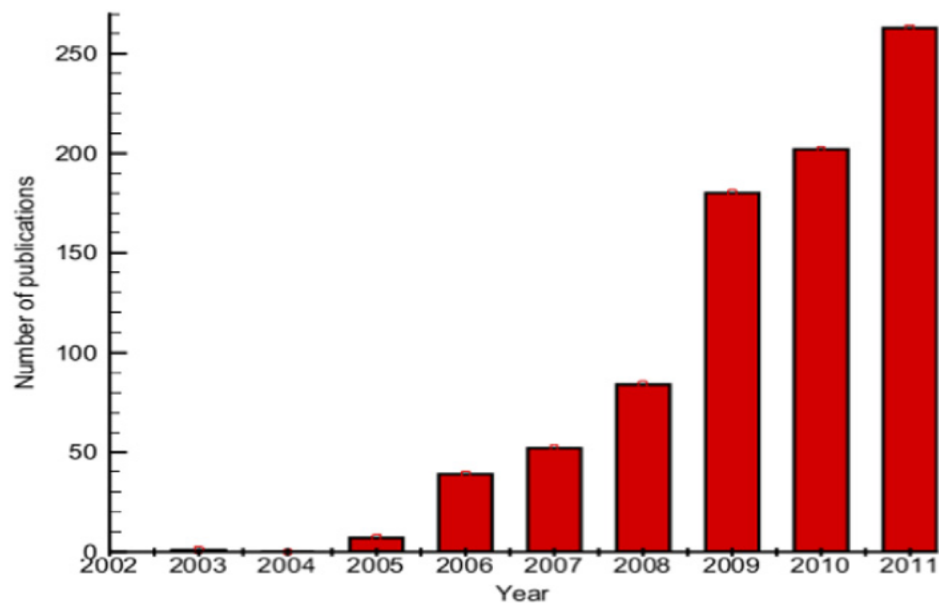


Figure 1.2 Number of publications on botnetss per year (Silva et al, 2013, p. 379)

The Figure 1.2 shows the number of publications on botnets started to grow in 2005. The Number of publications on 2009 almost doubled the number of publications in 2008. In addition, many publications have been focusing on the botnet behaviour especially inside the infected host that includes the computer

machines and the network system (Shahrestani, Feily, Masood, & Muniandy, 2012).

The motivation for this research is to focus on the forensic evidence from botnets that collected from the infected host machine. The evidence that looked for in this research is from the infected host where the unauthorised changes to the infected host are located and ascribed to the botnet attack. Furthermore, the contribution is to focus on the C&C server communications between the infected hosts (bots) and the C&C servers as there are not many publications in this area. The communications between the bots and the C&C server could provide information in regards to the activities of the botnets. The lack numbers of publication in this area encouraged the researcher to study the C&C communications as well as the evidence on the infected host.

1.2 THE RESEARCH APPROACH

This research has reviewed and analysed previous publications on the botnet area to be able to perform research in a new research area. This research reviewed four previous studies on botnets to achieve the goal of having an effective research methodology. The research has developed seven sub-questions to guide the direction of the research. Furthermore, there are four asserted hypothesis that the sub-questions will support.

In addition to the sub-questions and the asserted hypothesis, the research designed into five phases. Phase one, is the stage of building the database of the malware binaries signature using the Dionaea honeypot. Phase two, is the infection of the host by the binaries of the bots that have been collecting using Dionaea in phase one. The forensic investigation will performed in this phase to be able to locate the forensic evidence in the infected host. Phase three is the stage when the research will analyse the malware binaries by extracting and to be able to understand the behaviour of the bots using malware analysis tools. Phase four, of the research will identify the involvement of the botnets in the infected host incident. In addition, the infected host will monitored to be able to identify any C&C server communication in the infected host using sniffer tools. In phase five the research will present the binary evidences, the C&C communications as well as the report of possible evidences that have been found in the infected host.

1.3 THE RESEARCH FINDINGS

The research will be setting up a secure and isolated laboratory environment of machines running Linux operating systems. In addition, the machine will be running a Virtual Machine (VMs) that has a Windows operating system installed in it to be able to perform the testing. The VMs environment is the most efficient, safe and flexible method to test the botnets in a laboratory. In addition, the cost of this research is limited and must keep within the budget for this research. The main operating system, which is Linux, runs the Dionaea honeypot. The VM uses a Windows operating system to test the behaviour of the botnet including the C&C communications inside the Windows operating system. Furthermore, another VM uses Windows to analyse the binaries of the botnet.

This research will be running a Dionaea honeypot to be able to download malware signatures as well as building a database that contains the information about the malware. The research will then identify the botnet binary signature from the malware binaries downloaded to be able to study the botnet behaviour. Identifying the botnets will be performing by using an external service such as sandboxes which also provide an analysis of the malware. This research will be using malware analysis tools and sniffer tools in top of the sandboxes to be able to understand the behaviour of the botnets. The existence of the botnets in the infected host would found. The activities of the botnets would found in the registry, file systems, network activity as well as the physical memory of the infected host. The evidence that would found in the infected host must preserve including the physical memory of the infected host.

In addition, one of the most important parts of the findings in this research is to capture the C&C communications between the bot in the infected host and the C&C server. The sniffer tools will be installing in the infected host prior to the infection to be able to listen to the communication and capture them. The communications that this research is aiming for is at the infection time as well as the steps of joining the C&C server. This research is to observe and record behaviours and not to involve attacks. It is conducted in an isolated and safe network environment with no harm to other people.

1.4 STRUCTURE OF THE THESIS

This research organized into six chapters starting with the introduction and conclusions in chapter 1 and 6 respectively. Chapter 1 is the introduction that

gives a brief introduction to the botnet and the threat that the botnets have caused to the internet community. What is more, chapter 1 describes the motivation of this research on studying the botnets in depth.

Chapter 2 provides a comprehensive literature review in the research area to be able to gather more knowledge about botnet research. The previous publications studied look at the botnets in different areas of impact, which means that looking, and reviewing them is one of the most important parts of this research to be able to gain previous learning. Chapter 2 also reviews the problems and issues in the botnets area in order to develop researchable questions.

Chapter 3 is the chapter where the methodology of this research is developed. Reviewing the relevant previous research is the starting point of understanding and developing successful methodology. Furthermore, the data requirements identified and presented in this chapter along with the limitations of the research.

Chapter 4 is the presentation of the findings of this research. The experiments that have performed, reported in chapter 4. It begins with the review of the data collection, data processing, data analysis and presentations. Then the information about the honeypot that has used during this experiment listed. A brief description of the Dionaea honeypot in regards to the malware binaries collections during the 22 days that it was running. The information that has gathered from the external sandboxes and some of the analysis results that have provided by these sandboxes is given. Chapter 4 then will present the capture of the suspicious C&C server data captured from the infected host as well as the live monitoring of the infected host that locates the domain names and IP address that have requested by the infected host to the suspicious C&C server.

Chapter 5 is the discussion of the findings presented in chapter 4. Chapter 5 presented the answers of the sub-questions that have presented in chapter 3. In addition, the chapter 5 discusses the asserted hypothesis with the arguments for or against, and, conclusions. Then chapter 5 will discuss the Environment of the infected host, Data Acquisition and Extraction from the infected host, Reconstruction & Analysis, Command and Control Communication and the recommendations for tracking botnets. The recommendation for the tracking of botnets including the cross border issues, tracking the botnet master and improving the detection of the botnets are noted.

Chapter 6 draws the conclusion and suggests the future research directions for researchers to understand the botnet in depth. The suggestion for future works includes the improvements of the detection techniques, and disrupting the C&C server. In addition, the tracking of the botnet master to be able to locate the botnet master. In addition, the references and the appendix presented after chapter 6. The appendix research allows the readers to be able to view the additional results that have gathered from the experiments in this research.

Chapter 2

Literature Review

2.0 INTRODUCTION

The botnets are a growing threat to the Internet since the first known botnets that found in the early 90s. The IRC channel established in 1988 to support the owner of the computer and allow the owner to perform something in the computer while busy doing something else. The botnet has taken advantage of this channel to use it with the communication between the botnet master and the bot in the victim's machine. This research will focus on understanding botnets in regards to the architecture, components and actions.

The reason for studying the botnets in depth is that the cybercrime has grown in the last few years to obtain confidential information that stored in the target's machine or more important is to obtain a financial profit. The botnet has taken a lead in cybercrime because of the functionality and the difficulty of detection. The challenge for the researchers is that the botnet is developing hiding capability along with the development of the technology, and this leads to the difficulty of the detection of the botnets. The previous researchers have addressed different types of techniques that have used to detect the botnets. The challenge is that botnets usually avoid these techniques to find new techniques to avoid detection. In addition, the high standards of the code of botnets are another challenge. The reason for that is that some parts of the botnet hidden and it are difficult for the researchers to access and analyse.

This chapter introduces the definition of the botnets with the features and history in section 2.1. In section 2.2 gives an overview of the architectures that have used in botnets. In section 2.3 introduces the components of the botnets, section 2.4 illustrates the action of the botnets that include the propagation and the defence mechanisms. Section 2.5 illustrates the information security and the forensic investigation against the botnets attacks. Section 2.6 outlines the issues and challenges that faces the researcher and finally the conclusion for this chapter is in section 2.7.

2.1 DEFINITION OF BOTNETS

This section designed to define botnets. The botnet is a serious threat to the computer environment and that is why it is important to understand it. This section will define the botnets, explain the motivation for making them and then elaborate the botnet role in cybercrime.

2.1.1 Definition Of Term

A Botnet is a collection of computers or a large network of compromised computers (Ullah, Khan, & Aboalsamh, 2013). A bot refers to malicious software that runs on an infected computer. A bot provides a control to the attacker (Rajab, Zarfoss, Monroe, & Terzis, 2006). A bot is also known as a virus of viruses (Clark, Chaffin, Chuvakin, Paladino, Dunkel, Fogie, Gregg, Grossman, Hansen, Petkov, Rager, & Schiller, 2008).The attacker has control of the bots by using the C&C command channel, which exchanges the command between the attacker and the bots that receive instructions from the attacker (Correia, Rocha, Nogueira, & Salvador, 2012). The attacker usually uses one or more servers in order to allow the attacker to control the bots (Zahid, Belmekki, & Mezrioui, 2012). The command received through the C&C server executed autonomously and automatically without the end user's consent. The botnet is also known as an army of zombies and the reason for that is that they hide themselves until they become activated by an instruction (Choo, 2007). In addition, the bot network and botnets referred to as connection of networks for communication with each other. Also the attacker who controls the C&C server is called the botnet master (Rajab et al., 2006).

A bot is a completely different from other type of malicious software that harms the computer or a network. A bot is program that acts like an agent for any type of illegitimate activity (Rouse, 2005). Therefore, a bot can be independent software. A bot software executes the commands without making any communication with its operator (Grizzard, Sharma, & Nunnery, 2007). Some of the researchers have defined the botnet as a collection of bots that connected to each other through a malicious network which are using a computer technology resources for their criminal activity (Grizzard et al., 2007). There are many types of botnet communications such as C&C and Peer-to-Peer (P2P). However, with

the continued development of computer technology new types of botnet can also be developing.

Botnets started to appear to the world in the late nineties. Botnets started to target ecommerce as well as the government's websites (Heron, 2007). The botnet started with Internet Relay Chat (IRC). Also, The first botnets that is well known is an Eggdrop which was published in 1993 (Silva, Silva, Pinto, & Salles, 2013). The growth of the computer capacity, storage, and high process speed, has supported the growth of the Botnets. In addition, the increase of the internet speed have made the communication between bots and the botnet master even easier and faster (Heron, 2007). Nevertheless, the IRC channel established in 1988 and the reason for inventing this channel is that to support the owner of the computer multitask. As mentioned above this channel was first started to be used in a malicious software in 1999 (Clark et al., 2008).

2.1.2 Botnets In Cybercrime

The Crime-space for cybercrime is a communication between two places using the internet communications. (Britz, 2009) has defined computer crime as “any criminal act committed via computer”. The computer related crime defined as “any criminal act in which a computer is involved”. The cybercrime usually attempts to find out a way to get unauthorised access to any computer or system to steal any sensitive data for different type or purposes such as stealing money.

The motivation of the cybercrime has been totally changed from been just for curiosity to being a financial purpose (Choo, 2007). In addition, cybercrime includes any criminal activity against any type of data or the content of that data as well as breaching any copyright infringement (Gordon & Ford, 2006). In New Zealand a study by the New Zealand National Cyber Security Centre (NCSC) has been shown that the purpose of the attacks that occurred in 2012 has been targeting the private sector which means that the cybercrime motivations have been focusing on the financial gain ((NCSC), 2012).

One of the main issues that face the governments and its agencies is that the response to the cybercrime is slow. What is more, the governments have asked its agencies to develop a software that meets the requirement of the legal compliance and forensic investigation (Britz, 2009).

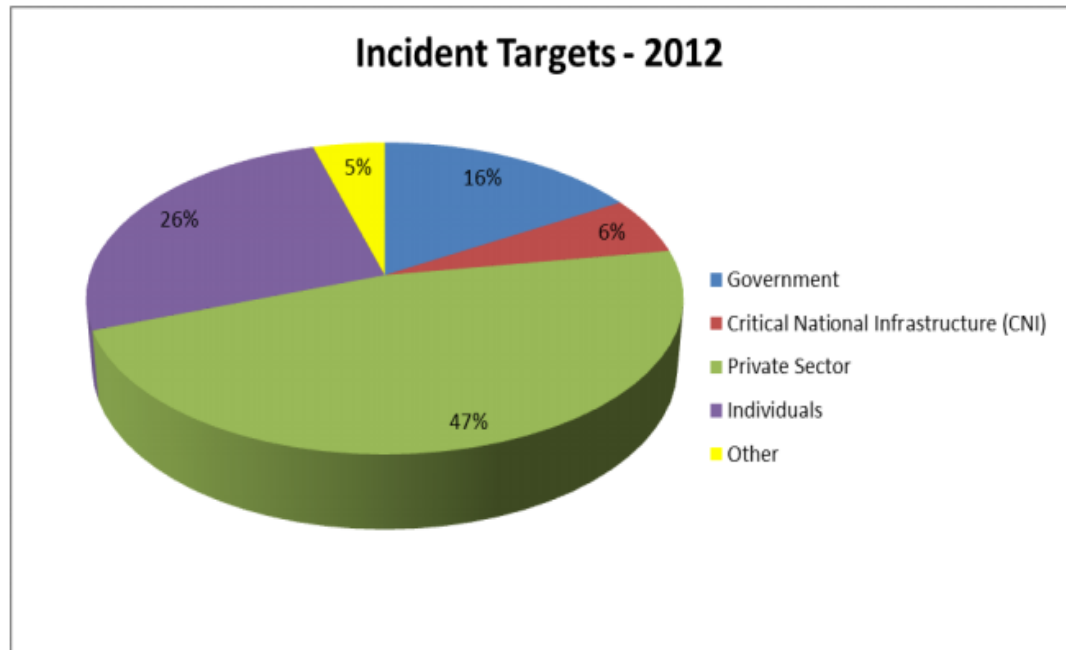


Figure 2.1 The official report from the New Zealand National Cyber Security Centre (NCSC) that shows the target for the cybercrime in 2012 in New Zealand (NZSC, 2012, p. 3)

The figure 2.1 above shows that the main target of cybercrime in New Zealand was targeting the private sector with the percentage of 47% of the incident reported in 2012. The second target for the cybercrime in New Zealand in 2012 was individuals with the percentage of 26%. It can be seen that the target for the cybercrime in 2012 was making up to 73% against private sector and individuals ((NCSC), 2012). The intention of the cybercrime organisation is the purpose of financial gain, which the graph above shows. The botnet cybercrime in New Zealand NCSC has showed that there is an increase number of the botnet related incidents reported in New Zealand. The report shows that the highest incident that occurred in 2011 related to computer incidents was Botnets and related incidents. This means that the botnet is one of the biggest threat to the internet community. The percentage of the Botnets and related incidents in New Zealand in 2011 made up to 23% ((NCSC), 2011).

2.1.3 Botnets Feature

Botnets defined in the previous section as malicious software that installed into the victim machine has and send information without the consent of the owner. Therefore, this section will discuss the feature of the botnets in term of the

network features and the software features. The best way to have a clear picture about botnets is to compare bots that are running on the infected system with the other malicious elements such as virus and worms. First, this section will discuss the network feature of the botnets. Botnets have different types of software that develops a more botnets. The software that develops a botnets is called the “killer Web App” that allows management and propagation (Clark et al., 2008). One of the features is that after an attacker programs a large number of bots they will give an opportunity to an attacker to transmit thousands of spam emails in a short time. Each bot will be sending only a few emails (Xie et al., 2008).

One of the features that make botnets more powerful is that botnets can propagate based on a mathematical algorithm modelling. This feature support a botnet attack by spreading bots through a specific network (Rrushi, Mokhtari, & Ghorbani, 2011). Botnets are able to have propagation just like other types of malware and may be self-replicating. The attack effect will be dependent on the measures of network infection rates and network susceptibility either directly or indirectly (Rrushi et al., 2011). Bots usually achieve control of the target’s computer without having the attacker to log into the target’s computer. In fact the bots communicate with each other using the C&C server to receive an instruction from the attacker to achieve the same goal (Schiller et al., 2007). The main challenge with a botnet server is that one or more servers could be linked to each other to control a few hundreds if not thousands of bots client (called zombies), for the previous reason this research will classified the botnets depends on the C&C (refer to section 2.1.5) channel.

The second part of the botnet features is the botnets software features. One feature is that the attacker cannot be reachable meaning that the attacker is hidden by using many IP addresses flooding a single target (Heron, 2007). This feature does not only hide the identity of the attacker it also hides the way that the attacker comes to the target’s machine (Heron, 2007). The bots receives the instruction from the botnet master through the C&C control channel, and then the bots will perform the task that the botnet master asked for and report the result through the C&C control channel. This means that the bots are able to adapt with any environment as well as being accurate and targetable (Dietrich, Rossow, & Pohlmann, 2013). Botnets classified into two phases that required being consider.

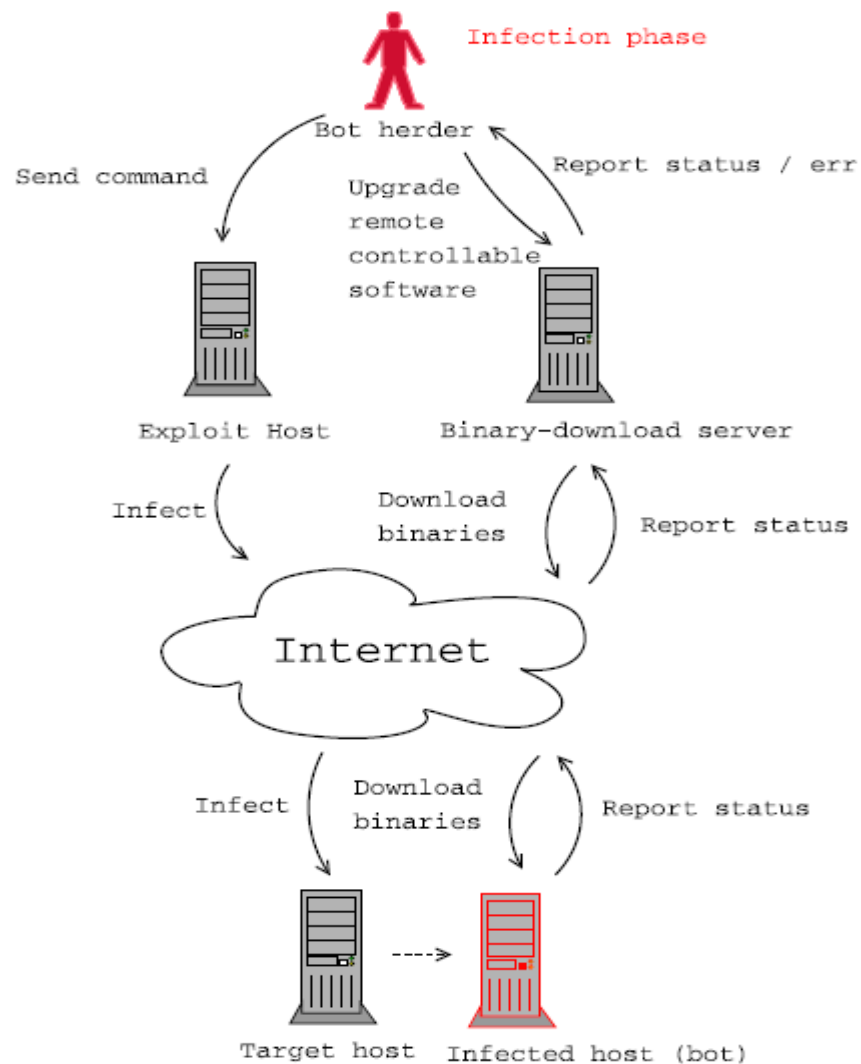


Figure 2.2 The infection phase (Wang, Huang, Lin, & Lin, 2011, p. 3277)

The first phase is the infection phase, which is the process of spreading bots over networks and machines. The infection phase is including the collection of the different kind of malicious code, trying to expand the army of the bots to get more and more victims. Expanding bots army has different techniques such as looking for software vulnerabilities as well as scanning for open ports. Once the machine of the target has successfully compromised then the remote controllable software will be downloading from the botnet server to the target's machine.

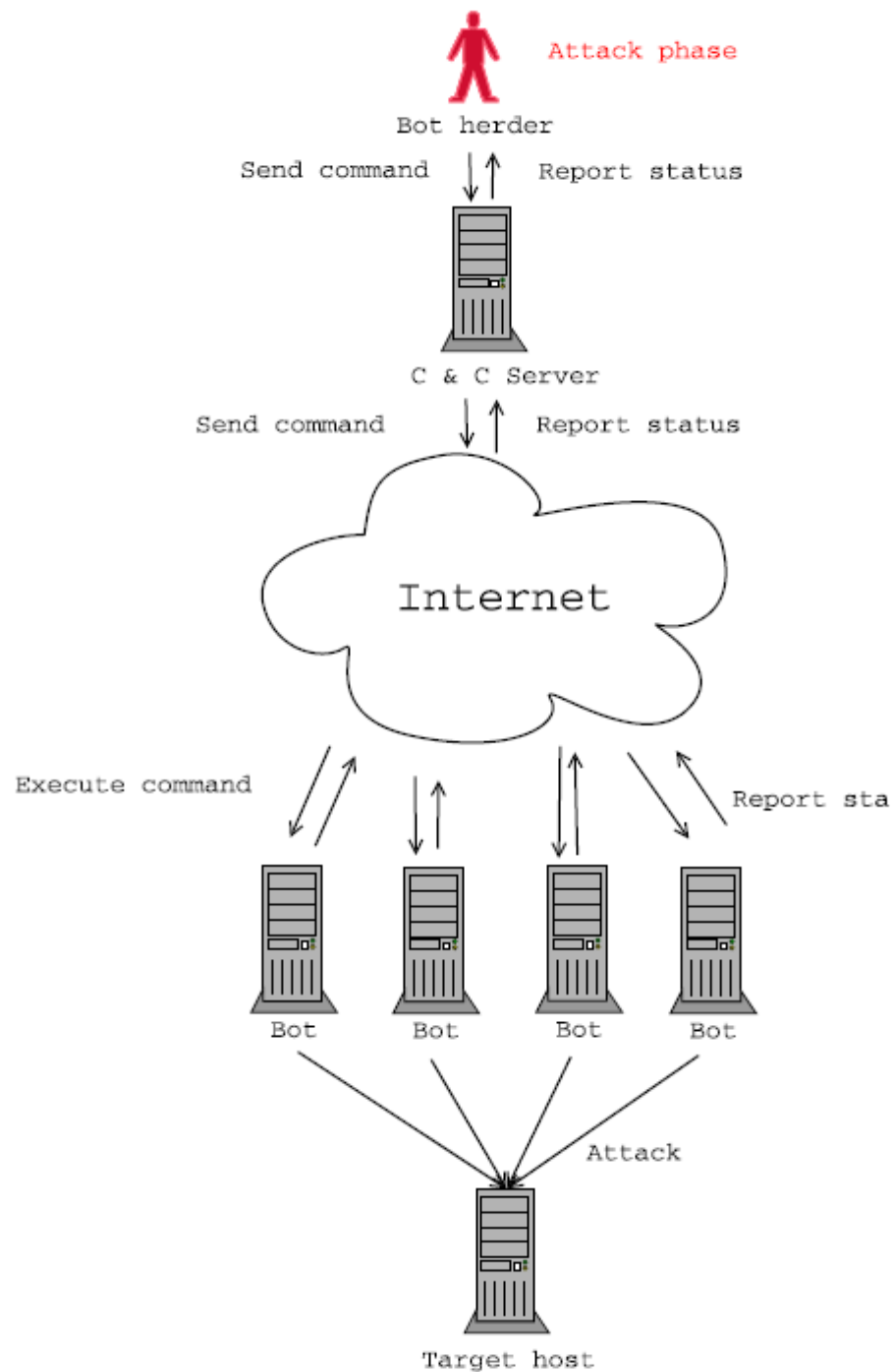


Figure 2.3 The attack phase (Wang, Huang, Lin, & Lin, 2011, p. 3277)

The second phase is the attacking phase, in this phase the attacker will send a command to bots in the compromised machine to perform a specific task that the attacker send through the C&C. In this stage the attacker is able to collect any type of data that the attacker is valuable for the attacker using bots which will report any data that the attacker collects (Wang, Huang, Lin, & Lin, 2011).

In addition, bots in the target's machine will have a different type of mission to do which means that bots in the target's machine are unemployed to

have a similar purpose. For example there will be a module that looks for vulnerabilities in the target machine, another one stops any type of security in the target's machine that would detect bots such as firewall and antivirus (Song, Jin, & Sun, 2011). What is more, bots missions do not stop in the target's machine. After installing the botnets software and compromising the machine of the target, bots look for a new target to increase the number of compromised machines and having more and more bot agents to increase bot numbers as well as having bots survive longer (Schiller et al., 2007). Botnet attacks can target into a specific sector such as an organisation or a business. In addition, botnets are targeting private enterprises such as businesses and individuals for a financial gain, therefore, botnets can be customised to target these sectors in order to achieve their goals (Schiller et al., 2007; Song et al., 2011). In addition, bots take an advantage of a backdoor left by other types of malicious code (Bailey, Cooke, Jahanian, Xu, & Karir, 2009).

2.1.4 Building Of Botnets And Its Lifecycle

In this section, the review will be looking at building the botnet in order to understand the process of building. The creation of the botnets as well as maintained them can typically classify into 5 phases. The five phases are initial infection, secondary injection, connection, malicious command and control, update and maintenance (Feily & Shahrestani, 2009). The first phase of building a botnet is the most important phase as an attacker tries to take an advantage of known vulnerabilities to infect the target's machine. In this phase an attacker will scan the target's machine for known vulnerabilities in order to infect the target's machine (Feily & Shahrestani, 2009). There are many ways for the bots to be installed in the victim's machine such as opening a malicious attachments through a spam email or connecting to a malicious server (Lu, Rammidi, & Ghorbani, 2011). In addition, the bots will be looking for another new victim as part of the propagation process by looking for known vulnerabilities in a new machine. Bots look for a new host or target randomly or by looking for a specific host and scan it in order to achieve an advantage of one of its vulnerabilities to expand their activities by downloading the malicious bot code using social engineering techniques as well as Trojan insertion (Feily & Shahrestani, 2009). When the initial injection successfully achieved then the secondary injection starts by

executing shell-code that known as a script. The shell-code fetches the image that contains the bot binary form an exact location through a File Transfer Protocol (FTP), HTTP or Peer to Peer (P2P) (Feily & Shahrestani, 2009). Once the bot binary (software) is installed in the victim's machine, then, the machine of the victim becomes an army of Zombies that will run a malicious code in the target's machine (Bailey et al., 2009). The attacker then launch the C&C server and the reason for that is that this connection channel will communicate the attacker with the bots and the attacker will have the control of these zombies' armies. After that, the fourth phase begins when the attacker is able to send commands to the bots through the C&C server to execute it in the victim's machine when the bots received the commands. This channel will enable the attacker to control the large number of bots (Feily & Shahrestani, 2009).

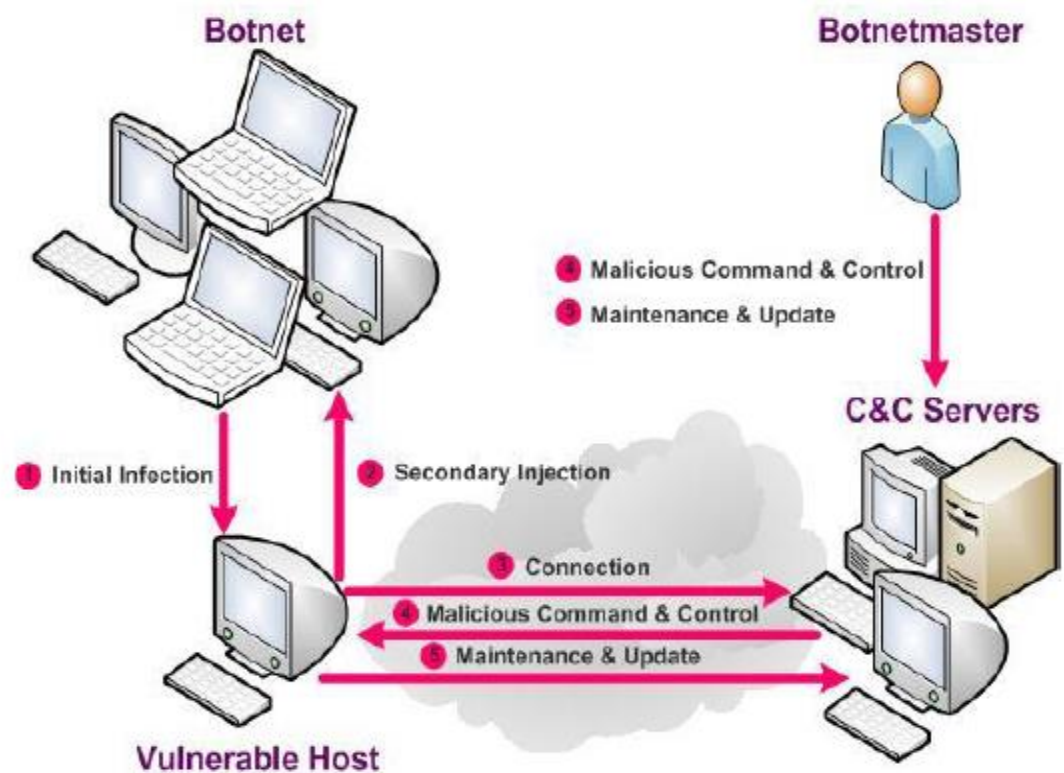


Figure 2.4 Building a botnets. The 5 phases that have been mentioned in this section in details (Feily & Shahrestani, 2009, p. 269)

The last phase of the building a botnet server is the update and maintenance phase when the attacker is able to update its software (bots in the victim's machine) for

different types of reasons. For example, the botnet master needs to update the binary of the bots to avoid the detection of the bots.

In addition, the botnet master may need to adopt or add new features to the functionality of the bots as well as relocate the C&C server that connects to the bots. This means that the IP address of the server that controls a number of bots will change but the server will keep the same name. The bots will be updated with the IP address for the new server as soon as the server launches as the short time-to-live (TTL) values the name sets by the DDNS provider. The main purpose of changing the C&C server to a new one is that this approach will keep the server and the bots alive as the IP address of the server can be blocked due to the detection. Therefore, in order to keep them alive the C&C server and the IP address need to be changed so they last longer (Feily & Shahrestani, 2009). Figure 2.4 summarises the botnet phases that have been mentioned in this section. It can be seen clearly that the botnet master communicates with the bots through the C&C server.

2.1.5 Brief History Of The Botnets

As the machines and internet have developed overtime and made it even easier for people to use the machine and the internet. Botnets have also incredibly developed over time just like the machines and internet. People are relying on the machines and the internet for almost everything in their lives such as shopping and online banking. The uses of the machines and the internet have encouraged the attackers to use their ability to develop malicious software that would assist them to steal sensitive information such as credit card numbers. Surprisingly botnets have used new techniques in the recent years as well as the functionality of the botnets has developed to a higher standard to prevent them from detection.

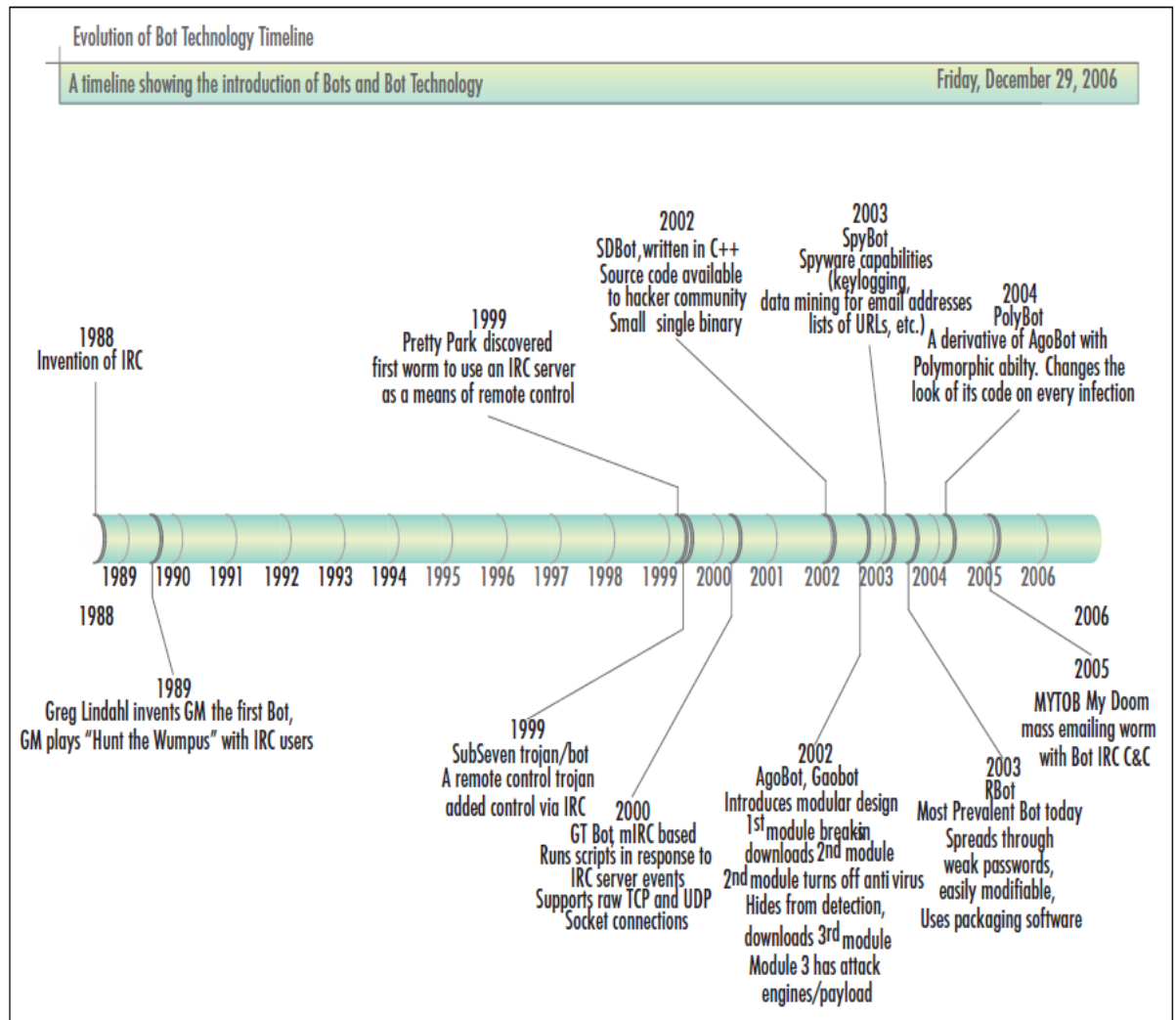


Figure 2.5 The evaluation of the botnets (Schiller, Binkely, Harley, Evron, Bradley, Willems & Cross, 2007, p. 6)

One of the malicious botnets that used the IRC discovered in May 1999. The name of the botnet was Pretty Park, according to (Schiller et al., 2007). The botnet Pretty Park used to have many of the functions and concepts that most of the botnets have. The Pretty Park botnet written in the Delphi that provides so many capabilities to the botnets. The Pretty Park provides many features including retrieval of the operating system of the target's machine as well as the version of the operating system, user information and other basic information. In addition, the Pretty Park provides the capabilities of searching through the email, retrieves username and password, upload and download files as well as update the functionality of the bots in the target's machine. What is more, the Pretty Park provides the capabilities of launching many DoS attacks, redirects the traffic as well as incorporation of its own IRC client (Schiller et al., 2007).

Just a few months later in May 1999 there was another botnet that discovered which called The SubSeven. The SubSeven was also written in Delphi. The version 2.1 of the SubSeven was discovered in June which enabled the attacker to remote control the bots via the IRC server connection (Schiller et al., 2007). The main challenge for this botnet was that it created a backdoor in the victim's machine by running the SubSeven server. This type of botnet received a command via an IRC channel which was popular at that time and many botnets has taken advantage of the SubSeven design (Lee, 2009).

In 2000, another botnet was discovered which is called a Global Threat (GT) Bot (Lee, 2009; Schiller et al., 2007). The Microsoft Internet Relay Chat (mIRC) can run scripting interface that respond to IRC events. In addition, it supported raw TCP and UDP socket connection, which allowed a variety of spoofing for an open port as well as denial-of-service (DDoS). The GTBot has high functionality in regards of the bots age that perform a port scanning, packet flooding, an IRC cloning as well as enabled the botletclient to access an IRC server anonymously (Schiller et al., 2007).

In addition, another bot appeared in 2002 called SDBot, which written by a Russian programmer. The program was written in C++ program and was a huge step up for the botnets history (Schiller et al., 2007). The SDBot written in 2000 lines of the C++ and released to the internet by the Russian programmer that, made it easy for the attackers to access to it. The source of code gave the ability to create a web page and provide e-mail and ICQ contact information. The easy access to the code let other attackers add modification and maintenance. Similar to other botnets, this botnet provided a remote control backdoor just like the rest of other malicious software. However, the SDBot code was not really modular or clean, even though the code was released to the attackers community (Lee, 2009; Schiller et al., 2007).

AgoBot (Aka Gaobot) that arrived to the internet community in 2002. AgoBot has increased the botnets performance due to the modular design and the significantly high build functionality with around 20,000 lines of C/C++. The AgoBot does not infect a system with only one bot as the infect phase has three modules to be performed. The first module is to contain the IRC bot client to remote access backdoor. The second module is stopping the antivirus of system from working. The third module is to stop the victim from accessing a list of

websites which are usually antivirus websites (Lee, 2009; Schiller et al., 2007). This type of bot can contain various components for different purposes such as propagation, communication, harvesting sensitive information and attacking targets (Lee, 2009). In addition, some other types of bots related to this bot include Phatbot, Forbot, Polybot and XtremBot. Phatbot uses WASTE P2P file sharing protocol to extend the control of botnets (Lee, 2009; Schiller et al., 2007).

Many other types of malicious codes have used different kinds of techniques to obfuscate the binary payload in order to avoid signature-based IDS system (Lee, 2009). For instance, the Rbot introduced the use of runtime software package encryption tools such as Morphine, UPX, ASPack and PESpin (Schiller et al., 2007). Polybot, which appeared in 2004, used the code base of the AgoBot and its name for its use polymorphism. This bot modified its code whenever there is a new machine infected by the bot. In addition, Mytob discovered in 2005 and used source code from My Doom. This type of code used social engineering and spoofed email addresses. As the source of botnet, code became modular as they use different open source licences for their code. The antivirus vendors are attempting to identify the botnets by its functionality such as spam e-mail and launching a DDoS attack rather than identifying the overall bot (Schiller et al., 2007).

2.2 BOTNETS ARCHITECTURE

This section will review the previous works that have published on botnet architecture. The activity and the behaviour acts differently depend on the type of the botnets. This section will discuss some of those popular botnet architectures.

2.2.1 Centralized C&C

Control and command C&C server is what makes the botnets more powerful than other type of malicious malwares. The botnets protocols based on the C&C server can be classified into IRC, P2P, HTTP and TCP (Correia et al., 2012; Lu et al., 2011). The centralized topology is classified as a central point which is responsible for the communication between the clients by forwarding the messages between them (Bailey et al., 2009). The communication between the bots and the centralized server requires a password, therefore, the bots would be programmed to have this password in order to authenticate and communicate with

the server (Schiller et al., 2007). The centralized system use a low latency that means the transition only needs a few recognised hops.

However, the centralized C&C has some major issues. The first main issue is that the detection of it can be easy as many clients connect to it. The second main issue is that the discovering of the central location means it can be shut down all the clients simply by blocking the whole server (Bailey et al., 2009).

In next few sections (2.2.2, 2.2.3, 2.2.4 and 2.2.5), this research will discuss different types of protocols that are used for communication between the C&C server and the bots. In addition, it describes how the Doman Name system (DNS) used with these protocols.

2.2.2 IRC Internet

IRC is the most popular protocols used between the bots and the C&C server. The Internet Relay Chat (IRC) originally designed to for large users and network of servers to support them in case of any failures by providing scalability and resilience. The IRC protocol provides a communication between the botnet master and the bots via either a private message or a broadcast. The IRC has been used since 2001in cybercrime, however, recent studies shows that the IRC protocols has been used in many botnets which means that this protocol still exists in the botnets as a source of communication between the botnet master and the bots (Kharouni, 2009; Zhuge, Holz, Han, Guo, & Zou, 2007).

The bots connects to the IRC server channel using a unique nickname in order for the server to identify each bot. This makes sure that the bots are an authentic member of the botnets server (Lu et al., 2011). The IRC provides an encryption communication between the botnet master and the bots (Choi, Liu, & Seo, 2010). After setting up the nick name and successfully authenticating the communication between the bot server and the bots, then, each bot will be waiting and listening for the command to be received from the botnet master to execute them on the victim's machine (Lu et al., 2011). However, there is a disadvantage for the IRC based protocol which is that the IRC server can be affected by shutting down due to the vulnerable based on highly centralized architectures (Zhao et al., 2013). Furthermore, detecting and blocking the IRC server is not difficult as it can be filtering and a list of them in the blacklisting of the filter of the firewall of the machine or firewall (Lu et al., 2011).

2.2.3 HTTP

HyperText Transfer protocol (HTTP), The C&C server has moved to use HTTP protocol to communicate with the bots (Chiang & Lloyd, 2007). The reason for that is that it allows more flexibility. In addition, using the HTTP protocol allows the C&C server to avoid any weakness if having a single point of failure (Lu et al., 2011). For example Social engineering is being involve with a tempting of people interest in order to encourage people to open the malicious attachment or link that contain malicious website (Lu et al., 2011). One of the main reasons for the botnets to use such a HTTP protocol is that the http protocol is a firewall friendly protocol which means that the chance of the detection is more difficult than the previous protocol (IRC) (Jang, Kim, Jung, & Noh, 2009).

In addition, the bots on the victim's machine can focus to run within the applications' process of the victim's machine including web browsers such as Internet Explorer (IE) (Daswani & Stoppelman, 2007). A previous case study shows that the http communication classified into two phases based on HTTP POST form. The two phases are key exchange and instruction. The key change is similar to the C&C server communication, this phase use the POST form of the HTTP protocol. The second phase is the instruction include a valuable number from the client's side which respond to the server instruction (Chiang & Lloyd, 2007). The POST form allows joining messages send between the server and the bots by identifying the operating system of the victim's machine in order use a specific port for this communication (Chiang & Lloyd, 2007). This communication known as Rustock shows the backdoor rootkit that Chiang and Lioyd explains in their case study.

The main issue with the HTTP and IRC protocol is that they can be detected even though they both are able to provide a great communication tool to the botnet master (Lu et al., 2011). Attackers who use IRC and HTTP protocol for their bots have noted that there are a drawback on C&C communication (Grizzard et al., 2007). The HTTP and IRC detected by systems, however, the IRC is easier to detect than HTTP. The main thing that the botnet master will lose if the C&C is blocked is the central point of control as it contains most of the client's information that the botnet master need (Wang et al., 2011).

2.2.4 P2P

Peer-to-peer (P2P) is a network when any node in the network can act as both a client and a server. This research mentioned in the section 2.2.3 that the IRC and HTTP detected and blocked by the system. However, the P2P protocol is really difficult to detect (Wang et al., 2011). One of the features in using P2P protocol is that the botnet master does not have to rely on a central server as the P2P communication is able to manage the communication between the botnet master and the bots without using the central server. In addition, the botnet master will be able to manage to upgrade and control the bots on the client's machine while not being detected (Grizzard et al., 2007).

What is more, the botnet master does not have to worry about the control server. This means if one of the servers related to a specific bot is blocked, then the bot will be able to connect to another server as each bot act as a client and a server. The drawback and the centralized network gives more features and flexibilities to encourage the attacker to use the P2P protocol rather than other types of protocols (Grizzard et al., 2007). The C&C communications are really difficult to detect and the reason for that is that the design for the P2P is complex which makes it undetectable in the network layer (Yan, Ha, & Eidenbenz, 2011).

2.2.5 DNS

Domain Name system (DNS), the botnet master uses the DNS to have more flexibility of controlling the bots. The DNS is not a communication protocol to the botnet and master uses the DNS to avoid detection (Lu et al., 2011). The DNS deals with the domain name with a set of IP addresses while the URL shortening service (USS) deals with the domain name with the URL. The USS does not allow the registration of a domain to modify as well as the USS allows only one URL for each alias. On the other hand, the DNS allows the registration of a domain name to be modified as well as several IP addresses for each domain name (Lee & Kim, 2013). An example of that is that when a host is willing to connect to google.com, then, it has to obtain the IP address for google.com for instance XX.XX.XXX.XXX from the DNS.

When a bot tried to connect to the C&C server, it looks for an IP address which can be obtained by using the DNS query (Silva et al., 2013). The bots try to hide the IP address of the DNS server so that the server is not blocked. This

means that the bots will take the advantage of the compromised sever such as phishing website, therefore, the IP address of the bots' server does not get blocked. In addition, the botnet master will have the advantage of hiding his identity, as the phishing website will work as a proxy between the bots and the botnet master. There is one disadvantage of redirecting the communication between the botnet master and the bots which is tracking down the command pathway can be difficult (Lee & Kim, 2013).

The main challenge for the botnet master is to keep the C&C server to last longer as many security vendors block down and track the IP addresses of these servers. Therefore, the botnet master creates a new method that assists them to keep the C&C last longer by changing the IP address of the C&C server frequently. One of the methods that the botnet master uses called fast flux domain flux (FFDF) which is changes a set of IP address of the C&C server frequently. The main purpose of this method is to hide the real IP address of the C&C servers. The FFDF makes it difficult to detect the IP address of the C&C server as the bots connect to a server that change frequently which means shutting down or blocking the C&C server is not an appropriate solution of fighting against the botnets (Lee & Kim, 2013).

2.3 BOTNET COMPONENTS

This section will discuss the botnet components and describe the work of botnets. The botnet components can be categorised as botnet master, command and control channel, bot and victim.

2.3.1 Botnet Master

A botnet master (attacker) is the person who drives the collection of bots and responsible for all the operations going between the server and the bots. A botnet master is responsible for all the social communications between the bots as well as the communication between the bots and the server. Botnet master components, which, classified into three main categories: a botworker, a botupdater and a C&C engine. In the first category, a botnet master builds and maintains the bots to be able to infect different types of machine as well as the communication between them (Boshmaf, Muslukhov, Beznosov, & Ripeanu, 2013). The developer of the botnets can be a person or a group of people who build and design the botnet.

However, the developer of the botnet does not have to be the owner (botnet master) of the botnet as the code and the hackers' community can deploy the design of the botnets. In addition, the developers of the botnet could gain a financial support to implement and design a particular botnets. There are many of malware tool kits that are available online for anyone interested on this type of malware to download that allows them to build and administer the botnets (Gomez, Andez, & Garc'ia-Teodoro, 2013).

Botupdater is the second category; in this category, the botnet master updates the bots, updating the bots to new software, updating the bots with new software, updating the bots with a new command from the botnet master. In addition the botnet master can update the bots with a new C&C engine due to the blocking of the previous C&C engine or for detection purposes (Boshmaf et al., 2013).

Finally, the C&C engine, C&C engine (channel) works like a warehouse of the botnet master command as well as is responsible of the controlling the bots by the botnet master. The C&C server will forward all the messages from the botnet master to the bots. In addition, authenticate the new bots to the zombies' army (Boshmaf et al., 2013). The next section will discuss more about the command and control (C&C) channel.

2.3.2 Command And Control (C&C) Channel

The command and control (C&C) channel is the interaction in the botnets. The C&C server grouped into three characteristics: Type of messages, direction of the information and communication protocols. Type of the message classified as a command sent by the botnet master that could be an order for the bots to perform a particular action in the victim's machine. In addition, the type of message could be controlling the bots by the botnet master. Controlling provides information about the botnets such as the number of bots that are active in a particular botnet. What is more, the direction of the message classified by either pulls or pushes. The bot usually requests information in pull C&C messages. However, the information gets received in passive manner, and the bots do not send the previous request as is the case in the push case. The communication protocols play an important part of the communication between the bots and the server. The most

common protocols that are used in the C&C communication is the IRC, HTTP and the P2P protocols (Gomez et al., 2013).

IRC C&C
server : String
port : int
channel : String
admins : String[]
callback : function

Figure 2.6 The configuration option for the IRC (Lee, 2009, p. 51)

Firstly, the IRC-based control model is central to a particular IRC server or channel. The IRC listens to the command that the botnet master sends through a post in a chat room. The bots listen to the messages in the chat room and perform the action. The configuration parameters of the IRC component shown in the figure 2.6. The server has the IP address of the IRC server, port means the port number of the IRC server, channel means the chat room to exchange the messages, admin means botnet master that may include nicks to accept the command and the callback means function that process each line of the input by the admin (Lee, 2009).

HTTP C&C
url : String
interval : int
callback : function

Figure 2.7 The configuration option for the HTTP (Lee, 2009, p. 51)

Secondly, the HTTP-based control model is continuously accessing a website to obtain new instructions. When the page successfully downloaded then the page would contains a callback function. the configuration of the HTTP component would be similar to figure 2.7 url containing the URL for the web site to obtain new instructions, interval is the rate time to polls of the web page and the callback function is called whenever a new web page is downloaded. Finally, the Peer-to-Peer (P2P) protocol control model are complex, however, P2P is popular to use in how they maintain their network. The P2P grouped into three groups: Peer Management, Message Passing, Search/Publish and Presentation. Peer Management has the responsibility of tracking the active connection and select different peers for different task. Message Passing is responsible for passing the messages between the botnet master and the bots. Message passing usually

contains commands; however, Message Passing does not carry any stolen information or updates. Search/Publish, this handles the searching for resources in the P2P network, and publishes handles searching for links to update the spam. Presentation is responsible for formatting the messages and request from and to the other three subcomponent and the network (Lee, 2009).

As mentioned in the section 2.2.5 (DNS) the C&C server uses the method fast flux domain flux (FFDF) to change the server the IP address of the sever frequently. In the C&C server uses the alias flux method that is similar to DNS fast flux method but the difference is that the alias flux method in the C&C servers is that it changes the alias of IP addresses of C&C servers instead of their domain name (Lee & Kim, 2013).

2.3.3 Bot

The bot is the software that installed in the victim's machine without the awareness of the owner. The bot is usually malicious software that is capable of performing an action in the victim's machine. The bot usually installed in the victim's machine in a different way such as opening an email attachment or accessing untrusted or malicious website. Usually the configuration of the bot is to be launch whenever the victim's boots their machines. After the bot launch in the victim's machine then the bot will be ready to receive an instruction from the botnet master through the command and control channel (Silva et al., 2013). Then the bot can act as an agent for the botnet master in the victim's machine (Schiller et al., 2007).

The bot group depends on the protocol they are using to communicate with the C&C server. Botnets usually use IRC, HTTP and P2P, which is the most popular protocols for botnets. For example, the IRC bot use ping-pong mechanism to stay alive. The ping-pong is usually a small size file as the C&C server will not be able to handle a large size file. The IRC bot usually customized to wait between 30 and 600 seconds. On the other hand, the HTTP bot is different from the IRC bot as the HTTP bot does not use any persistent TCP connection to stay alive in the C&C sever. The HTTP bot connects to the C&C server from time to time to get new commands or new instructions. This means that the HTTP bot does not receive the command instantly as the HTTP bot connects to the server from time to time. Another difference between the IRC bot and the HTTP bot is

that unlike IRC bot, the HTTP sends a command in a very large file. The size of the file is about 1000 bytes and the reason for that is that the HTTP bot has to establish a new TCP connection that contains a three way handshake (Zhao et al., 2012).

2.3.4 Victim

The victim is the target of the botnet where the intention is to infect the victim's machine with the bots to be able to control the machine via the C&C server. The victim could be the system, person or network, which is the object where the attack executed. The victim is vary and depends on the purpose of the attack, and the botnets. For example a user who receives a spam or confidential information that has been stolen. Another example is that the company who loses several millions dollars due to the DoS attack (Gomez et al., 2013).

The victim usually selected for different reasons but mainly to gain a financial profit. When the bot installs itself successfully in the victim's machine, then, the bot will destroy the entire program that will defend the victim's machine. The reason for that is that the bot usually stays hidden in the victim's machine. The botnet master is able to distinguish the victim's as each bot is uniquely identified so the botnet master knows which one of the bots can be used in the victim's machine (Schiller et al., 2007).

2.4 BOTNET ACTIONS

This section will discuss the action that the botnets perform including the propagation of the botnet to increase the army size. This section reviews the propagation, Bot Terminated Process and compromised machine.

2.4.1 Propagation

There are different methods of the propagation method of the botnets. This section will discuss some of the propagation methods of well-known botnets. The SDBot usually counts on the vulnerabilities of the security on the target system. In addition, the SDBot can take the advantage of the user to connect to with other network resources. The right of the access and privilege of the user who logged into the system assumes by the SDBot to be the same. The SDBot will the take advantage of default administrative to make the connection and spread the bot.

The results can be found in a typical windows system such as PRINT\$, C\$, D\$, E\$ or ADMIN\$. In addition, the SDBot is known to scan the SQL server installation looking for any vulnerabilities in the administrator password or a security issues in the configuration of the SQL server (Schiller et al., 2007).

Another botnet propagation method is the RBot, which scans the windows network for an open port in either 139 or 445 to connect to it. The aim of the scan is to get the IPC\$ administrative share on that system. Then the RBot will try to get a list of the usernames and passwords if the connection to the IPC\$ administrative share is successful. The interest of the list of the usernames and the passwords in the systems is to get access to it. If the list of the usernames and passwords is not successful, then, it will simply try a preconfigured list of usernames that is in the malware (Schiller et al., 2007).

-
- Microsoft Windows LSASS buffer overflow vulnerability (TCP port 445)
 - Microsoft Windows ntdll.dll buffer overflow vulnerability (Webdav vulnerability) (TCP port 80)
 - Microsoft Windows RPC malformed message buffer overflow vulnerability (TCP ports 135, 445, 1025)
 - Microsoft Windows RPCSS malformed DCOM message buffer overflow vulnerabilities (TCP port 135)
 - Exploiting weak passwords on MS SQL servers, including Microsoft SQL Server Desktop Engine blank sa password vulnerability (TCP port 1433)
 - Microsoft Universal Plug and Play (UPnP) NOTIFY directive buffer overflow and DoS vulnerabilities (TCP port 5000)
 - DameWare Mini Remote Control buffer overflow (TCP port 6129)
 - Microsoft Windows Workstation service malformed message buffer overflow vulnerability (TCP port 445)
 - Microsoft Windows WINS replication packet memory overwrite vulnerability (TCP port 42)
 - RealSystem Server SETUP buffer overflow vulnerability
 - Microsoft SQL Server 2000 Resolution service buffer overflow vulnerability
 - Microsoft Windows Plug and Play service buffer overflow vulnerability

Figure 2.8 The Known Vulnerabilities Commonly Exploited by Rbot Variants (Schiller et al., 2007, p. 110, 111)

The figure 2.8 Shows that thee known vulnerabilities commonly exploited by Rbot Variants. The Rbot used this list of vulnerabilities to propagate itself. If one of the vulnerabilities found in one of the target machines, then, the RBot executes a small program that instructs the target machine to go to the remote server and download the full code of the RBot. The connection back to the RBot source might use a different port for connection such as port 81 which indicates HTTP and port 69 which indicates TFTP (Schiller et al., 2007).

Another example of the botnet propagation is the Agobot family. The Agobot family spread the bot army in P2P network using WASTE. AOL designs

P2P protocol. WASTE designed to use an encryption algorithm during the transfer of the file in P2P for a security reasons. However, there is a disadvantage for the WASTE as the WASTE only manages between 50 and 100 clients' nodes, which means that the bot army can be limited in P2P network. The Agobot spread the bot through open network shares until the target machine infected, then, the bot will seek to get the username and password. In addition, the bot will attempt to search for administrative information to be able to get the username and the password. The Agobot is preconfigured and the bot has a list of common usernames and passwords.

Vulnerability	Port(s)	Microsoft Security Bulletin
• DCOM RPC vulnerability	TCP 135	MS03-026
• LSASS vulnerability	TCP ports 135, 139, 445	MS04-011
• SQL Server and MSDE 2000 vulnerabilities	UDP 1434	MS02-061
• WebDav vulnerability	TCP 80	MS03-007
• UPnP NOTIFY buffer overflow vulnerability		MS01-059
• Workstation Service buffer overrun vulnerability	TCP 445	MS03-049
• Microsoft Windows SSL Library DoS vulnerability		MS04-011
• Microsoft Windows Plug and Play buffer overflow vulnerability		MS05-039
• Microsoft Windows Server Service remote buffer overflow vulnerability		MS056-040

Figure 2.9 Vulnerabilities Exploited by Spybot Variants to Help it Propagate (Schiller et al, 2007, p. 122, 123)

In addition, the Spybot has similar propagation method to rest of the bot families. The Spybot looks for open or poorly secured networks, which able to spread and compromised other systems. Figure 2.9 show the vulnerabilities that the Spybot is looking for. The Spybot usually scans the target systems to be able to identify one of the vulnerabilities shown in the figure 2.9. The Spybot aims to achieve the username and the password in the targets systems with a bot that is preconfigured with a list of common usernames and passwords and similar to the other bots.

wab	php
adb	sht
tbb	htm
dbx	txt
asp	pl

Figure 2.10 The File Extensions Known to Be Commonly Targeted by Mytob for Harvesting E-mail Addresses (Schiller et al, 2007, p. 126)

The figure 2.10 Shows the files extension that the Mytob is interested to find in order to execute the bot in the target's machine. Mytob known as an email bot because it spreads the bot through an email attachment. Mybot find the extension shows in the figure 2.11 to infect the target machine by those extensions.

.gov	gov.	mydomai
.mil	hotmail	nodomai
abuse	iana	panda
acketst	ibm.com	pgp
arin.	icrosof	rfc-ed
avp	ietf	ripe.
berkeley	inpris	ruslis
borlan	isc.o	secur
bsd	isi.e	sendmail
example	kernel	sopho
fido	linux	syma
foo.	math	tanford.e
fsf.	mit.e	unix
gnu	mozilla	usenet
google	msn.	utgers.ed

Figure 2.11 Mytob Eliminates Harvested E-mail Addresses with the Following Domains (Schiller et all, 2007, p. 126, 127)

The figure 2.11 shows the domain that Mytob targeting to spread the bots and execute it in the target's machine. Mybot uses those domains to propagate the bots through an email attachment (Schiller et al., 2007).

2.4.2 Bot Terminated Process

The botnets goal is to stay undetected all the time and in order to achieve that the botnets need to be able to remove any program that detected its bot.

regedit.exe	MSBLAST.exe
msconfig.exe	teekids.exe
netstat.exe	Penis32.exe
msblast.exe	bbeagle.exe
zapro.exe	SysMonXP.exe
navw32.exe	winupd.exe
navapw32.exe	winsys.exe
zonealarm.exe	ssate.exe
wincfg32.exe	rate.exe
taskmon.exe	d3dupdate.exe
PandaAVEngine.exe	irun4.exe
sysinfo.exe	i11r54n4.exe
mscvb32.exe	

Figure 2.12 A Sample of Processes Sometimes Terminated by RBot (Schiller et all, 2007, p. 106, 107)

The botnets first attention is to use the bot to be able to remove any program that would detect the bot such as antivirus program. The reason for that is that program such as antivirus might remove the bot after detecting it. The Figure 2.12

shows the processes sometimes terminated by RBot. For example, regedit.exe indicates the registration editor in windows, which enables the user to modify registry entries, and msconfig.exe is executable file or a program in the Microsoft windows operating systems. In addition, some of the botnets families such as Agobot target some of programs and services to terminate. Mainly, the programs and the services, which Agobot targets are associating with antivirus and other security software. Also botnets can shut down any process that associates with computing malware (Schiller et al., 2007).

2.4.3 Compromised Machines

There are many signs of compromised machines that affect the machine of the victim. The botnets is like the rest of the malicious software, which means that there will be many signs that would indicate the machine of the victim's is under a threat. Botnets can place a file of itself in the system folder. The botnets such as SDBot can use the variable %System% to be able to place the system folder then place a file of itself in the system folder. The name of the botnets can be different and hard to keep track of it. Some of the well-known botnets files name that used for backdoor shown in the figure 2.13

Aim95.exe	service.exe
CMagesta.exe	sock32.exe
Cmd32.exe	spooler.exe
Cnfgldr.exe	Svchosts.exe
cthelp.exe	svhost.exe
Explorer.exe	Sys32.exe
FB_PNU.EXE	Sys3f2.exe
IEXPLORE.EXE	Syscfg32.exe
iexplore.exe	Sysmon16.exe
ipcl32.exe	syswin32.exe
Mssql.exe	vcvw.exe
MSsrvs32.exe	winupdate32.exe
MSTasks.exe	xmconfig.exe
quicktimeprom.exe	YahooMsgr.exe
Regrun.exe	

Figure 2.13 A Known Filenames Used by Backdoor for SDBot
(Schiller et al, 2007, p. 100, 101)

In addition, other botnets will have different names for the system files. For example, the RBot will have different filenames such as wuamgrd.exe; most of the botnets will use the %System% directory to copy the file into the file system with a read only hidden. What is more, the botnets uses some of the machine files system details such as the timestamp and the date of the files botnets files system

to match the details of the system's files. Details such as matching the timestamp and date of the explorer.exe file so the victim thinks that the files have installed with the system files and have not changed since then. Other botnets such as Agobot can use following filename syschk.exe, svchost.exe, sysmgr.exe, and sysldr32.exe (Schiller et al., 2007). Likewise, botnets can modify the registry entries of the targeted machine. The main point of modify the registry of the target's machine is that whenever the victim turns the machine on such as the machine the botnets will automatically started whenever the operating system such as windows started. In addition, some of the botnets are preprogramed to check the registry value in case the value has changed, deleted or changed. Also some of the botnets run only one copy of the registry value which has different value from one machine to the others.

In addition, botnets can have additional files in the system files in order to improve the functionality of the botnets. For example there are two files have been noticed in the SDBot, which are SVKP.sys and msdirectx.sys. The first file SVKP.sys is a protection of the software of the machine that give a prevention to the software from being revers-engineered. The botnets uses some techniques that prevent the security program from identifying it. The second file is msdirectx.sys, which provides a higher functionality for the botnets that guarantee a full control and access to the victim's machine. In addition, some of the botnets such as Agobot prevent any attempt to access any of the security and antivirus website in order to prevent detection of it. The Agobot redirect any attempt to access to this website to a different website that set by the botnets developer(s).

Unexpected traffic is another sign of compromised machines that uses open port to access and communicate. For example, the SDBot uses the port 6667 that uses TCP and 7000. Usually the bot tried to connect to the IRC server and uses some open port in the victim's machine, which identified by analysing the traffic of the network. The IRC configured to connect to the IRC server that requires channel, port number and password along with other information that the server required (Schiller et al., 2007).

2.5 BOTNET COLLECTION AND ANALYSIS

This section will discuss the collection and the analysis of the botnets performed by the previous researchers. This section will discuss collecting malware,

detecting botnets, honeypots used to collect malware, analysis of botnets malware and live vs static forensics.

2.5.1 Collecting Botnets

With the increasing of the botnets researchers become interested to analysis the bot and collect as many botnets as they can so they can analysis, and understand how it works. The botnets have developed in a high standard and quality especially the large botnets. The researchers have faced a difficulty analysing the code of the bot due to the professional way of writing the code as well as some of the code is hidden to make it complicated for the researchers to analysis it. The main challenge that faces the researchers is that the researchers do not want the code of the botnets to get into a wrong hands, therefore, the researchers find it difficult to collect the full version of the botnets as part of it is hidden by the developer(s) of the botnets (Lee, 2009).

In addition, collecting the binaries code of the bot is the main goal of collecting the botnets as well as collecting as many bots as possible. The main issue that may appear when collecting the botnets is that developing scalable and robust infrastructure. Any malware collection infrastructure must be able to support the wide array of data collection endpoints as well as it should be highly scalable. Therefore, the research was facing a challenge of finding a special implementation to avoid participating in malfeasance. Therefore, the research has found an approached that uses a honeypot that developed by professionals, which they found it safe to collect malware. The approach is basically based on automated malware collection, the reason for that is that the automatic way of collection assist the research of reducing the overload of deploying and maintaining honeypots. The result of the research has come with result that improves the understating of the botnets and the result will help the researchers to have better information about the attack patterns, attack trends and attack rates of malicious network traffic. The research has also shown that it is possible to investigate each piece of malware. The malware collected can be identified as possible evidence and also provide a fingerprint of the attack which can be useful for the investigation (Rajab et al., 2006).

In addition, other research shows that as understanding the botnets code can be quite difficult to understand as the developers of the botnets attempts to

hide part of the code to make it complicated for the researchers to examine. Lee (2009) has found a way of understanding more about the code of the botnets. After a few attempts the researcher managed to find a botnet master community website that provides a hint about some parts of the code sample (Lee, 2009).

2.5.2 Detecting Botnets

There are several methods, which, have introduced by previous researchers to detect the botnets. The detection of the botnet mainly based on monitoring the traffic. Many open source tools support monitoring of the network to detect any malicious traffic. For example, the netflow-based tools (Schiller et al., 2007). The network traffic has different type of traffic content and monitoring the traffic may result of finding different type of botnets such as HTTP-based botnets and IRC-based botnets (Lu et al., 2011).

Lu et al (2011) have classified the detection of the botnets into two categories, the first one is the supervised botnets detection and the second one is unsupervised botnets detection. The supervised botnets detection uses a labelled dataset to create the profiles of system or network. The drawback of these detection techniques is that it needs to label the training data, which means there could be an error-prone. In addition to the error-prone, there will be a cost involves as well as the time consuming to label the training data. On the other hand, unsupervised botnets detection uses the unlabelled data to identify the behaviours of the bot. This means that the drawback of the supervised botnets detection does not appear in this techniques as the training base on unlabelled dataset which improve the detection accuracy of the botnets (Lu et al., 2011).

Most of the existing botnets uses a centralized architecture, where the decentralized botnets have identified and detected more and more recently. Usually in botnets, one or a few compromised machines would configure as a command and control server such as IRC server. However, the main issue with the centralized architecture is that it can be easily detected as the C&C server is a central point of failure (Silva et al., 2013). One of the tradition ways of detecting a botnet is the signature-based detection, as Lu et al (2011) experiment discussed that 40% of the network flow cannot identified in a signature-based approach. This means that 60% of the traffic in the public WIFI can be identified based on the signature. Therefore, the new researchers are focusing on unknown signature

detection to improve the chance of detecting the botnets. Honeypot detection is one of the popular detection techniques that have used in detecting the botnets. The next section 2.5.3 will be discussing this detection technique. However, this detection technique will only work for the existing botnets and will not detect a new botnet which is known as a zero-day attack (Silva et al., 2013).

In addition, there is a host-based detection where each host is monitoring for any suspicious traffic activity including accessing files. However, the important thing about this approach is that all machines in the network have to have a tool installed in them called monitoring tool for this detection technique to be effective. What is more, other detection techniques are network-based techniques. These techniques involve monitoring the traffic either if it's active or passive. Packets to see the active network response such as a botprobe can inject the network. The downside about network-based techniques is that the traffic would increase which means that more traffic will be added to the network traffic (Silva et al., 2013). The main challenge for the botnets detection is that the botnets traffic does not have any difference than a normal traffic, which means that it requires time consumption to analyse it. In addition, the botnet traffic uses encryption techniques in order to hide itself from detected, which means that the analysis of the traffic as well as have knowledge about the different type of encryption algorithm is required to analyse the traffic. In addition, the botnet uses a fast-flux method to avoid detection mechanisms as mentioned in section 2.2.5. Also with a lot of load of traffic data passing through in a real time, it is really difficult to analysis the traffic data on a real time, this means a delay of detecting a malicious traffic passing through in a real time (Silva et al., 2013).

2.5.3 Honeypot

As mentioned in the section 2.5.2, the honeypot is one of the most popular detection techniques that used by many of the recent researchers. The main reason that many of the researchers uses the Honeypot and HoneyNet is that they are effective detection techniques as well as the cost of setting them up and running the tests is reasonable. The most important point of using the Honeypot and HoneyNet is that they do not have a false positive, which gathers more researchers to use it. One consists of the Honeywall and honeypot network that deployed the architecture. There is Chinese version of honeypot called HoneyBow that a tool

that collects malware in high interaction. In addition, Tang and Chen presented a novel “double-honeypot” the main interaction about this honeypot is that it detects internet worm attacks effectively. Bothunter that has a set of communication flows that exchanged between the internal host and the external entities has introduced another modelled. The main use of this modelled is to point out the difference between the suspected infection event (Li, Jiang, & Zou, 2009).

Pham and Dacier (2011) have presented another technique of using a honeypot. Their approach is to identify the botnets, however, the focus of their approach is to identify and study the size of the botnet army or zombies as well as the lifetime. Therefore, their focus is the size and lifetime of the botnet army. They are assuming that there could be different bots that related to different botnets, and in another words there could be more than one botnet in the dataset. They are interested in promoting their approach without any complication in use so it used widely. Their approach is a bit different from the other honeypot approaches as the honeypot usually installed in the network to detect the botnets whereas their approach targets of the attack. The datasets were located in different countries and have been running for more than 800 days with the maximum of 10 times that the dataset has downloaded. They have noticed that there were 2 attacks from the same botnet and they realised that by analysis the IP address as they have so many IP addresses and they looked really similar (Pham & Dacier, 2011).

2.5.4 Analysis Of Botnets Malware

The analysis of the botnets malware should include two major ways to analyse the malware, examining the code and its behaviour. The botnets developer(s) try to avoid the detection of the botnets as well as the analysis of the botnets code. Therefore, the botnets developer(s) use some techniques to avoid analysis of the botnets such as hiding part of the botnets code as well as uses encryption techniques. The issue that increases when the researchers tries to decrypt the botnets is that the decryption form is stored in the memory of the bot and only for a short period of time (Lee, 2009).

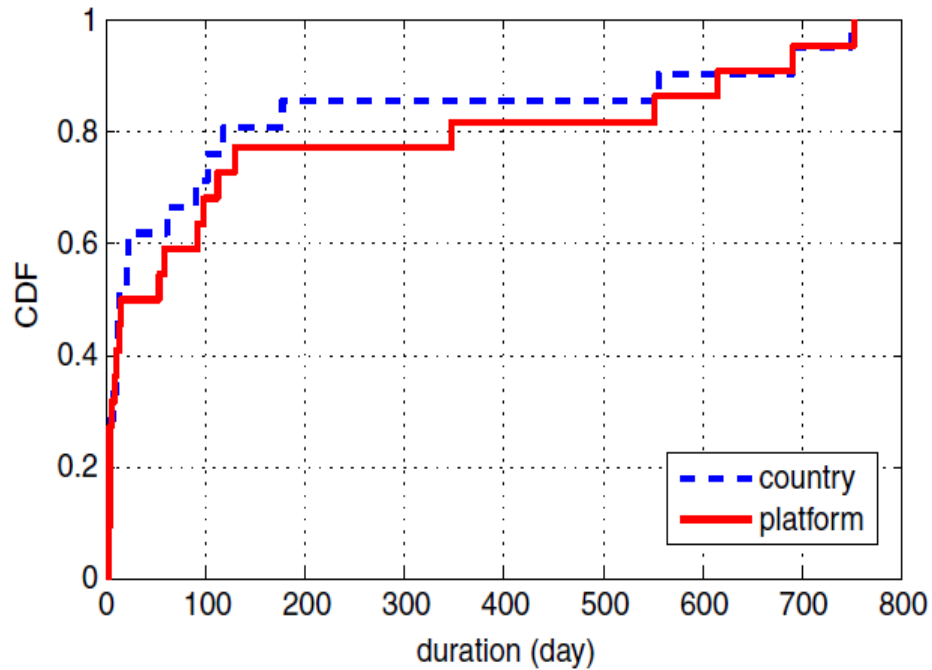


Figure 2.14 Cumulative distribution function (CDF) duration (Pham & Dacier, 2011, p. 543)

The bot can stay active in the victim's machine for a long period without detected in some cases. Some of them stayed active for more than 200 days and other stayed active for almost a 700 days. The figure 2.14 shows that cumulative distribution the lifetime of the botnets army and the country. The long-time indicates that the bot takes long time to compromised machines or the botnets army are able to stay active for a long time. This means that the victim's machine will infect with a new bot and that when one of the army becomes inactive another bot replaces it with an active status. In addition, the number of attacks gives a chance to the botnets malware to propagate, this means that the botnets will launch more than one attack to the same machine to increase the chance of increasing the number of infected machines. One of the main jobs that the bot is asked to do is to focus on the vulnerabilities as well as test for other vulnerabilities, however, testing for another vulnerabilities is a small subset that is asked by the botnet master to the bot to do (Pham & Dacier, 2011).

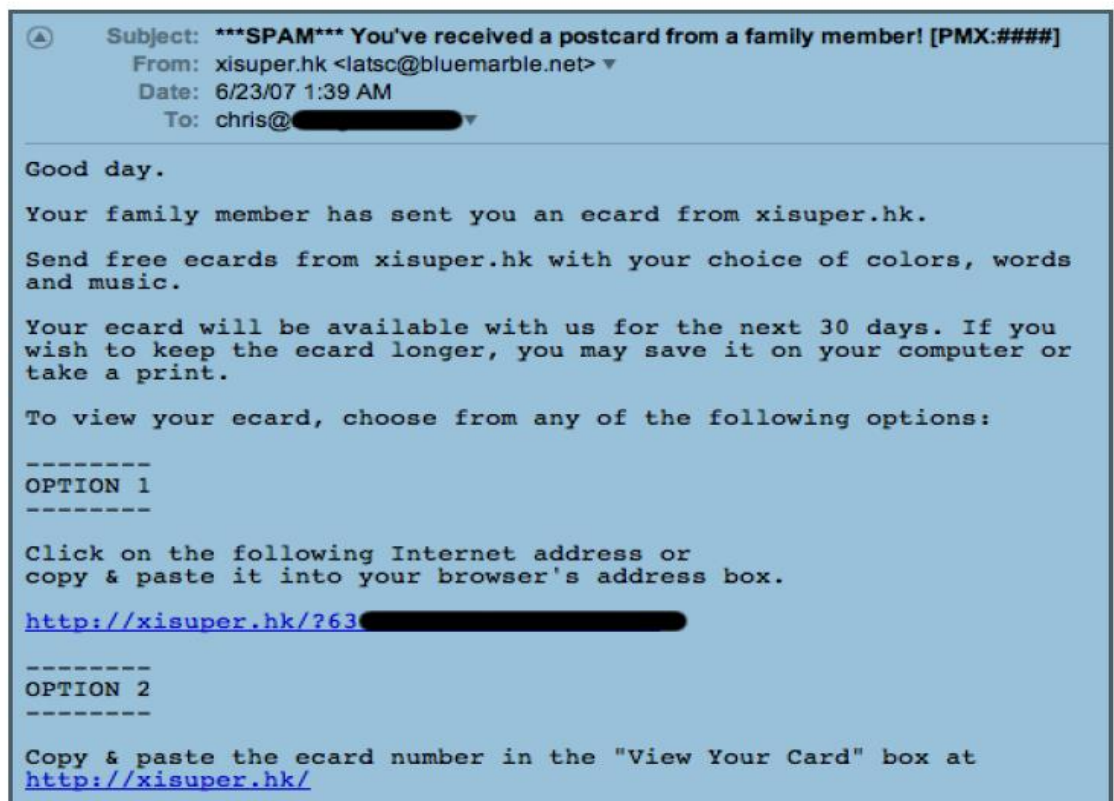


Figure 2.15 The spam e-mail enticing the victim to click and become infected (Lee, 2009, p. 26)

The figure 2.15 Shows an example of a storm in Europe that sends a link to the victims to click on it, then, the link will direct the victim to the website that contain a malicious JavaScript. The malicious JavaScript shown in figure 2.16

[illegible]

After that, the victim's machine will forced to download the primary infection binary as shown in the figure 2.17

```
<script>
try{x=unescape("%u9090%u9090%u9090%u9090%u00e8%u0000%u5d00%ued81%u11ce%u0040%ucce8%u0000%u8d00%u5e85%u4012%ue
800%u0007%u0000%u7275%u6d6c%u6e6f%ue800%u011e%u0000%uc389%u858d%u131e%u0040%u13e8%u0000%u5500%u4c52%u6f44%u6e7
7%u6f6c%u6461%u6f54%u6946%u656c%u0041%ue853%u00f8%u0000%u9090%u8d8d%u127f%u0040%u006a%u006c%u09e8%u0000%u6300%
u5c3a%u7e74%u6e69%u0078%u6a51%uff00%u8dd0%u6b85%u4012%u6a00%ue800%u0099%u0000%u3a63%u745c%u692e%u786e%ue800%u0
0be%u0000%u858d%u1273%u0040%u006a%ub1e8%u0000%u4c00%u616f%u4c64%u6269%u6172%u7972%u0041%u6957%u456e%u6578%u006
3%u7845%u7469%u7250%u636f%u7365%u0073%u7468%u7074%u2f3a%u722f%u7466%u7568%u2e64%u6f63%u2f6d%u616d%u2e6e%u7865%
u0065%u0000%u0000%u0000%u0000%u0000%u0000%u0000%u6000%u8b64%u301d%u0000%u8b00%u0c5b%u5b8b%u8b1c%u8b1b%u085b%uda89%u9
d89%u132d%u0040%u7b8b%u013c%u03d7%u785f%u4b8b%u8b18%u2073%u7b8b%u0124%u01d6%ufcd7%u01ad%u51d0%u9657%ubd8d%u131
e%u0040%u0fb9%u0000%uf300%u96a6%u595f%u0674%u4747%ue4e2%uc4eb%uc031%u8b66%uc107%u02e0%u738b%u011c%u01d6%uadc6%
ud001%u8589%u1331%u0040%uc361%uff50%u2db5%u4013%uff00%u3195%u4013%uff00%u47e0%u7465%u7250%u636f%u6441%u7264%u7
365%u0073%u0000%u0000%u0000%u0000");y=unescape("%u0d0d%u0d0d");while(y.length<0x40000)y+=y;y=y.substring(0,0x3
ffe4-x.length);o=new Array();for(i=0;i<450;i++)o[i]=y+x;z=Math.ceil(0xd0d0d0d);document.write('<object
classid="CLSID:EC444CB6-3E7E-4865-B1C3-0DE72EF3983F"></object>');z=document.scripts[0].createControlRange().
length;}catch(e){}
</script>
<iframe src="exp1.htm" width="1" height="1"></iframe>
<iframe src="exp2.htm" width="1" height="1"></iframe>
<iframe src="exp3.htm" width="1" height="1"></iframe>
<style> * {CURSOR: url("123.htm")} </style>
```

Figure 2.17 The decode JavaScript with shell-code and instruction to install ecard.exe (Lee, 2009, p. 27)

The botnets uses the encryption method to hide the information and the code from being analysed as mentioned earlier which makes it hard to analysis the code (Lee, 2009).

2.5.5 Live And Static Forensic

Live and static forensic is the process of the investigation of any type of cybercrime. There is advantages and disadvantages for both steps of the forensic investigation. The important thing about the forensic investigation is preserve the evidence of any type of cybercrime for prosecution. Firstly, the live forensic is imaging the infected machine when the machine is up and running. The forensic investigator will be collecting the evidence from the machine without making any changes to the data. The live forensics includes documenting all possible steps that have taken during the image of the infected machine. The important step in live forensics is to take as much information as possible from the memory of the infected machine. However, in some cases the machine could be shut down which means that the forensic investigator has only one option which is running the machine using a live DVD or USB so there will not be any changes to the hard disk of the infected machine (Yen, Yang, & Ahn, 2009).

On the other hand the static forensic also called (traditional static analysis techniques) is another important process of the forensic investigation. The static forensics means that after taking the part of the live forensics and imaging the infected machine, then, the forensic investigator will take the imaging to the forensic lab for the examination. The static forensics is usually analysis in the forensic lab which means taking the evidence away from the scene (Yen et al., 2009). The static forensic investigation is where the evidence stored in the media storage and analysis using forensic software. The most known forensic software is Encase and FTK which have the ability to analysis the data an advance way. The analysis includes electronic document, browsing history, email records, installed program and more importantly the files that the users have deleted and they think it has gone from their machines. The most important part of the forensic investigation especially the static forensic investigation is that during the imaging of the infected machine the blocking devices need to be used in order to prevent any changes of the evidence. The reason for that is that any changes of the evidence means that the evidence will not be accepted in court for prosecution (Hay & Nance, 2008).

2.6 REVIEW OF ISSUES AND PROBLEMS

The botnet is one of the biggest threats to the information security and the Internet. The botnet hides its army in the target's machine. Unfortunately, botnets have become a threat since 1991; researchers have been studying the botnets and focus on this threat only in the last few years. This means that the botnet has increased in regards to advanced coding as well as the functionality of performance. There are issues and challenges that faced the forensic investigation during the investigation of the botnets. This section will address some of the issues and challenges that make it hard to explain and understand.

The propagation is one of the challenges in botnets as the propagation of the botnets can change its plan and behaviour. The botnets have advanced rapidly with its propagation techniques. For example the push-based model and pull-based model, this causes a significant issue of the infection phase which means that the different type of social engineering techniques is used in order to increase the number of victims and machines infected (Gomez et al., 2013). The other issue that faced the forensic investigator is that the attacker sends a link to random

number of people that direct them to a particular website that download a malicious software into the victim's machine with the permission or the awareness of the owner of the machine (victim). In addition, the process scan of the scanning the victim's machine for known or unknown vulnerabilities could be another challenge, the reason for that is that each bot pre-programed to scan the target's machine for vulnerabilities. The main issue that faces the botnet researchers is that the botnet army (bot) is unable to be analysed as the developer of the botnet destroys the bot if the code of the bot has been breached to prevent the researchers from analysing the bot (Lu et al., 2011).

In addition, the detection of the botnets is another challenge that the researchers faced. The reason for that is that the botnets main goal is hide its identity. Therefore, the botnets change the IP addresses of the bot continuously as well as change the IP address of the server using the method fast-flux that changes the nickname of the server as well as changing the IP address of the server continuously (Lee & Kim, 2013). In addition, the signature-based techniques does not seem to work for the current botnets, the reason for that is that the botnets can easily change the signature of its bot which means that the antivirus cannot detect the new signature. A bundle software called Rootkit that hide the botnets from detection, implementation to modify the data flow and identifying the operating system of the victim's machine. Identifying the operating system will assist the botnets to hide the bot activity and existence the victim's machine. As mentioned in the section 2.4.2 the botnets has the ability to modify the infected host including the registry of the victim's machine without displaying the modification date on the victim's machine. What is more, the botnets has the ability to disappear some of the important information that the forensic investigation is willing to find the victim's machine.

Likewise, the forensic investigation is the most important part of any botnets attack. The regular process of the forensic investigation is to gather information from the machine's disk, memory when the machine is up, and running. In some cases, this is not possible because the machine has switched off for different reasons that mean that the information in the memory of the machine has gone and it is nearly impossible to retrieve. In addition, the live forensic investigation in the infected machine is not repeatable as the forensic investigator has only one go at the infected machine. The reason for that is that the forensic

investigator will change the state of the machines after examining the infected machine in the first time. Therefore, the forensic investigator can only rely on the read only software that guarantees there will not be any changed made to the targeted machine.

2.7 Conclusion

In chapter 2, the context of botnets, the place of the botnets in the cybercrime, botnets feature, building the botnets and lifecycle as well as the history of the botnets has reviewed. The section 2.1 the overview of the botnets in cybercrime, that has reviewed from previous research. Then this chapter shows the different types of botnet architectures that have used in the internet either for a communication between the botnet master and the botnets's army through the botnets's server or managing the communication between the botnet master and the botnets's army. In section 2.3, the components of the botnets that are involved in the botnet attack. The components of the botnets include the botnet server, botnet master, bots and the victim of the botnet attack. Then this chapter gives an overview of the capabilities of the botnets in propagation, defence mechanisms that assist the botnets from detection and fighting against the security software as well as the compromised machines that the botnets controls after infecting them. After the overview of the botnet and having a better understanding, this chapter addresses also the botnet actions in how they propagate the defence mechanism and the action taken in the compromised machine. Then this chapter point out the previous work of the detection, collection and analysis of the botnets malware. In addition, the forensic investigation that taken when a machine is being attacked by the botnets is reviewed.

The next chapter 3 will present the methodology that this research will use. The chapter 3 will establish methodology from previous researchers, presenting the research questions, sub-questions and the hypothesis. The laboratory environment will be defined to be able understand how this research is to be performed.

Chapter 3

Research Methodology

3.0 INTRODUCTION

The botnets are a serious threat and require investigation. The forensic investigation of botnets is part of the requirement for investigating cybercrime. Therefore, the evidence of the incident that involves botnets is to examine forensically to preserve the evidence in an infected host that contains valuable information about the incident. The focus for the forensic investigator in an incident is not letting the system or the network to shut down until the valuable information gathered from the infected host. Therefore, a live forensic investigation needed in order to perform memory examination. The forensic investigator has only one chance to perform and it is not a repeatable step. The forensic investigator will be responsible to investigate the cause of the incident and the reason of how the incident happened that includes the vulnerabilities in the system. This research will address the issues found during the investigation of the botnet events and the difficulties that the forensic investigator will face during the investigation of a malicious activity.

In Chapter 3 the method to be used in this research will be developed in order to perform a forensic investigation in the infected host. The research question and the sub questions will defin. In chapter 2, the research has reviewed previous works that have done by researchers who have researched botnets. The literature published has reviewed in order to focus on specific issues as the botnet field is a large area and this research will not be able to cover it all. Therefore, the research will focus on an incident involving botnets on the infected host side.

3.1 REVIEW OF SIMILAR RESEARCH

This section reviews previous studies on the botnets to observe how did they perform their studies as well as report the achievements they have made. This section will review four previous studies to identify the methodologies used.

3.1.1 Visualization Of Invariant Bot Behaviour

This study has done by Shahrestani, Feily, Masood, Muniandy, (2012). It focuses on the botnet behaviour in an infected host. According to their survey, the botnet has five phases, which are initial infection, secondary infection, connection, malicious command and control and update and maintenance. However, they have concentrated on the Connection, Malicious Command and Control, Maintenance and update.

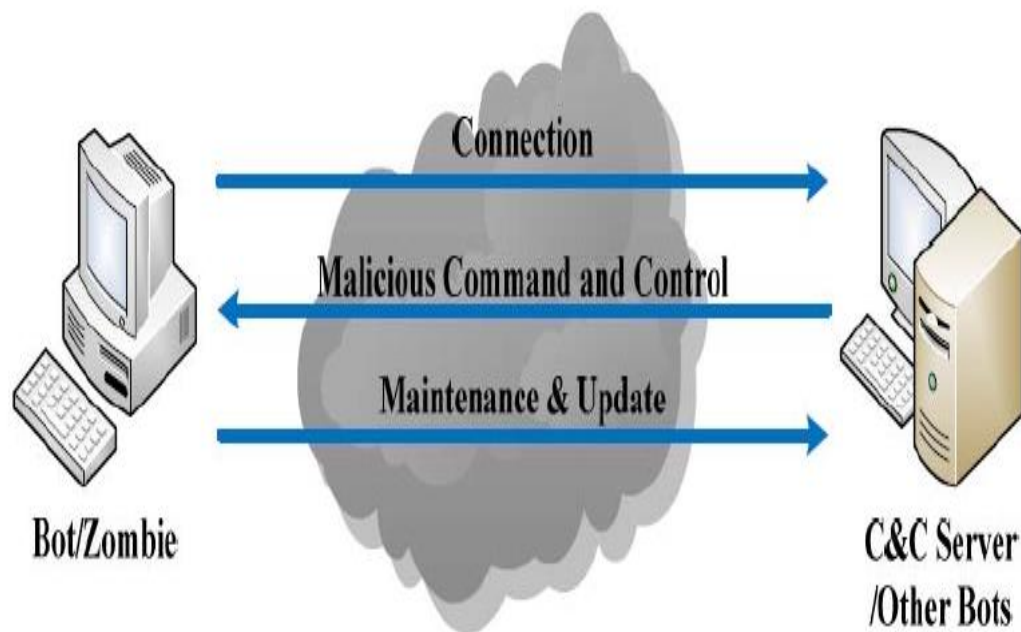


Figure 3.1 Three phases of botnets life-cycle considered for Invariant bot behaviour identification (Shahrestani, et all, 2012, p. 326)

The researchers focused on the botnet behaviour that involved the Fast Response Time, which is the time the bot takes to response to the botnet masters command. It uses a Small Size Command that is the size of the command and is usually a small packets size typically 1KB or less. The Instant Execution of Commands is the application that the bot launches in the infected host and is typically after the infection of the host. Their goal is to detect the existence of the botnet with evidence. They proposed “Visual Threat Monitor” VTM to identify the botnet behaviours in a monitored network. Their data source focused on the session data level. The characteristics that reflect the bot behaviours involve the following:

- The Response Time should be fast to consider as a command and that is a speed of 100ms for incoming packets and 3sec for outgoing packets.
- The size of the session should be small to reflect the small size of the command, which is less than 1KB.
- The Time Interval, which is the time between receiving the command and the time for the application to be launched in the infected host.
- The Session Count and the Destination Count should be low; however, the Average Count should be low which indicated that the machine is in the propagation process of the bot software.

They have used different techniques in their study to find out more about the bot behaviors. They have analyzed the overview of the traffic for both ongoing and outgoing traffic. The traffic was recorded either hourly or daily depending on the volume of the network traffic. The techniques they have used are a graph visualization of traffic overview, scatter plot of time intervals and parallel histogram visualization for time series. The graph visualization of traffic overview is monitoring the traffic that is going on and out of the network. The scatter plot of time interval assists to identify the destination of the traffic and the source, which provides the IP address for both the destination and the source. The parallel histogram visualization for time series focuses on the ongoing traffic to provide an evidence of an existing bot in the network.

The techniques that have been used in their study were focusing on detecting the behaviour of the bot in the infected host. Therefore, this study does not expect to detect any threat of attack. An expert and non-expert of the network examined the effectiveness of the proposed visualization. They both agree that this visualization increases the visibility of the network traffic. The percentage of this visualization was 78.57% in the assessment from the expert and non-expert.

3.1.2 Real-Time Botnets Command And Control Characterization At The Host Level

This study has been done by Etemad and Vahdani, (2006). The focus of their study was to detect the botnets in the host level as well as filtering the outgoing traffic. Their approach is to detect the existence of the botnets C&C communications in the host by analysing the inbound and outbound traffic. Their proposed detection of the botnets can be classified into two categories, which are protocol classifier and

communication patterns interpreter. Their proposed detection has two main components, which are the IRC part and the HTTP part that redirected to the protocol classifier. The IRC part is responsible for detecting the IRC communication with the C&C server.

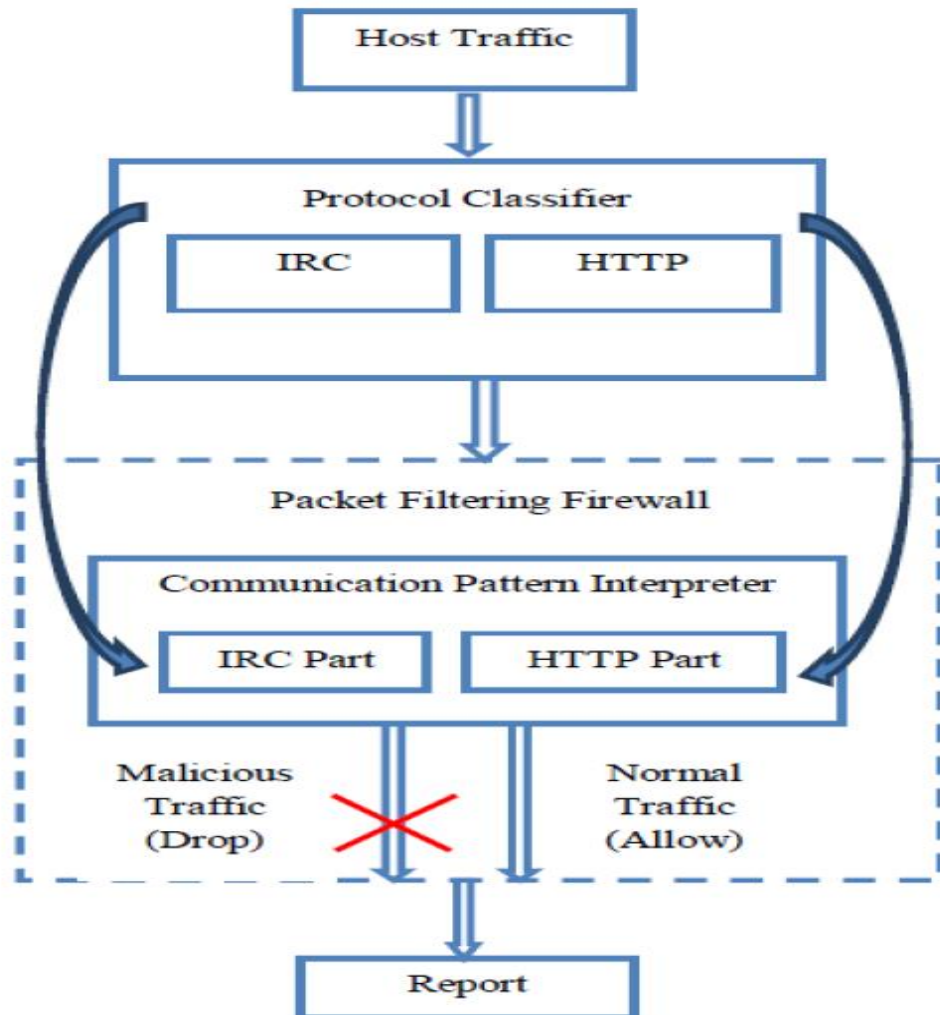


Figure 3.2 Architecture overview of our proposed approach (Etemad & Vahdani, 2012, p. 1006)

In addition, the HTTP part is responsible for detecting the HTTP communication with the C&C server and that based on Periodic Repeatability of messages. The malicious communication pattern then filtered from the normal traffic that is filtering in the firewall of the host.

In order for Etemad and Vahdani (2006) to detect the botnets based on the characterization of bot's C&C traffic. They have to separate the IRC and HTTP from the other protocols, as they are the most common protocols that used for the

communication between the C&C and the bot. The rest of the packets passed to the communication pattern interpreter components. The detection of the IRC traffic performs by inspecting the content of the packets for strings that indicate the communication between the C&C server and the bot. In order to recognize it, they looked for a NICK that is indicating the nickname of the clients, PASS for password, USER for username. In addition, they looked for JOIN for joining the channel as well as PRIVMSG for private message between the C&C server and the bot. Their aim was not to decrypt the IRC communication traffic as the botnet master uses this method to avoid detection. In addition, the detection of the HTTP traffic performed by inspecting the bytes of the early packets. They look for a specific pattern or keyword in the request message of the http. The client starts the connection to the server by sending a HTTP request to establish a connection, then, the server response to the request of the client by HTTP response message (i.e “here is the file”, then the file attached in the end of the contents). Then the HTTP becomes stateless which means is difficult to get the information of the transaction. Therefore, Etemad and Vahdani (2006) look for keywords from the outgoing traffic such as “GET”, “POST” and “HEAD”.

The communication between the C&C server and the bot usually done by using a “PULL” style and “PUSH” style based on the way a bot receives the command from the botnet master. What is more, they have categorized the IRC bot into two phases, which are the Phase 1 that indicates the period before the bot joins the IRC channel and Phase 2 that indicates the time after the bot has joined the channel. The HTTP bot usually is harder to distinguish, as they have to separate between the normal HTTP traffic and malicious HTTP traffic.

3.1.3 Collaborative Architecture For Malware Detection And Analysis

This study has done by Colajanni, Gozzi and Marchetti (2008). They have shown a collaborative architecture that aims to analyse an early detection as well as deployment of countermeasures. Their honeypot has a multiple sensors that records the malicious attempts from its location and collects the payloads of the offending worms. Some of the infection could be slow due to the firewall of some organizations that block some of inbound protocol connections. The sensor of their honeypot project monitors the malware spread, however, the local stored malware needs to be analysed for some behaviour and safe supervision. The low

interaction honeypots such as Nepenthes are able to collect the malware payload while the operation is not affected.

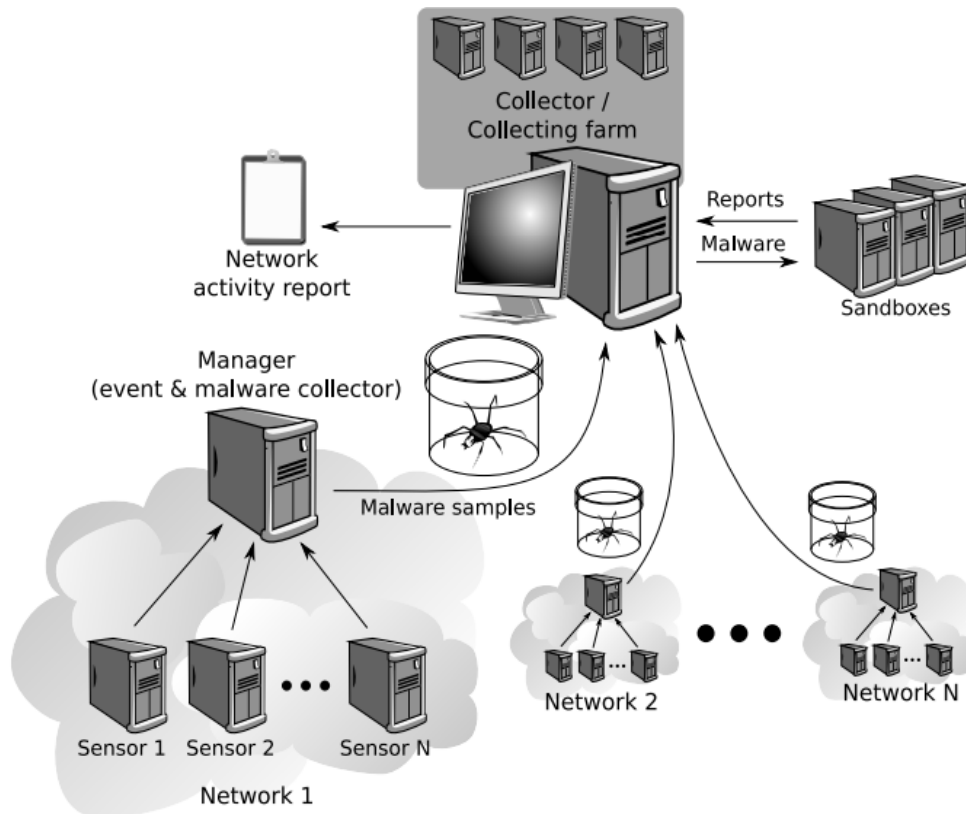


Figure 3.3 Cooperative architecture for malware detection and analysis
(Colajanni et al, 2008, p. 83)

Then, there is a Manager, which collects alerts and payloads from the sensors. One of the challenges that they faced is that the sensor does not display the information about the transfer facility of the Intrusion Detection Message Exchange Format (IDMEF), therefore, they have installed a script that assists to capture them and retrieves them with an unknown MD5 hash. The number of the payloads transferees kept to the minimum by transferring the new malware only to the manager. Each of the sensors has its local manager which manages the collection of the malware if a new malware is found then the local manager will transfer it to the higher level manager until it gets to the collector which is the highest level in this project.

The collector is the top of the hierarchical architecture that receives the new malware from the managers of each sensor. The process of the malware has two steps; the first step is identifying the signature of the malware by different antivirus engines. The second step is executed the malware in a protected environment such as sandboxing. The communication security is an important fact

the researchers have addressed to prevent a single node from polluting the set of collected data. They have used a public key cryptographic to solve this issue, as they had to trace back every alert to its origin.

3.1.4 Insight From The Analysis Of The Mariposa Botnets

This study has done by Sinha, Boukhtouta, Belarde & Debbabi (2010). Their study performed to understand the new technology of the P2P botnets. They have run their experiments on Windows XP using VMware 2.0.3 that allows the running of multiple virtual machines in an isolated environment. They also have used a Live CD for network security. The main purpose of their study is to analyse the bot behaviour as well as the analysis of the code of the bot. First, the host infected by the bot, then, the initialization phase takes place when the bot sends a request to join the server and register the IP address of the infected host. The latest message of the bot will include the some important information about the infected host such as the operating system and the country code of the infected host.

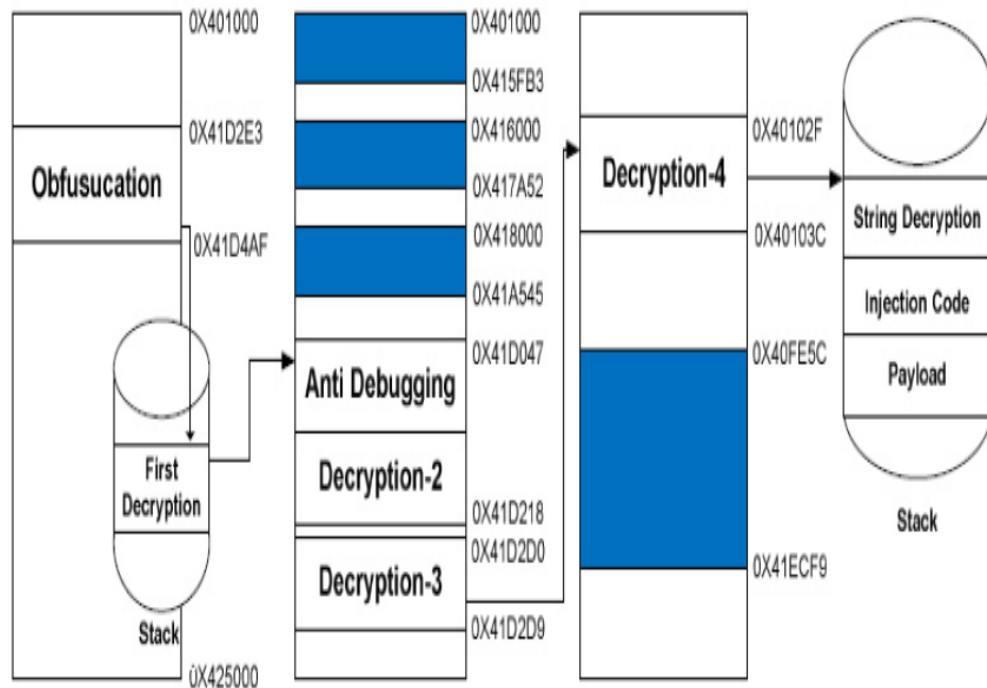


Figure 3.4 Overview of Mariposa Bot (Sinha et al, 2010, p. 4)

Then, the server checks the bot is still alive by sending a packet and waiting for the acknowledgment packet to receive, to make sure that the bot is alive. The action phase aims to send a request to the bot in the infected host to perform an action. They have analysed the Mariposa statically and dynamically as an important step because most of the botnets use a reverse engineering Sysanalyzer. Their purpose of the analyser is to get a bit deeper into the obfuscation and anti-debugging techniques as well as part of the code that executes the bot features. They found the code is confusing, as well as they found a loop that goes for 889,976,605 times and in the end of the loop, it goes over to an address that is located in the EAX register. The anti-debugging techniques are able to detect whether it runs in a controlled environment or a debugger.

In addition, they have also looked into the encryption techniques of the Mariposa botnets as it has three layers of encryption. They have reach part of the code that contains encryption routines, which is the second, third and fourth layers. They have found that the code of the Mariposa botnets code injected into the explorer.exe, which slows down the machine. The botnet master usually tries not to run more than one bot because that could crash the system.

Their study shows that the Mariposa botnets used to send emails spams and perform a DDos Attacks. An encryption key generation algorithm usually does the communication between peers.

3.2 RESEARCH DESIGN

Section 3.1 has reviewed some of the past research that has done in investigating and studying the botnets in depth. Their studies will be an advantage to this study to develop an effective research methodology and to adapt them to suit this study. This research aims to investigate the botnets at the host level. The main challenge of the botnets studies is that there are much network traffic and logs that need to be investigated and analysed, which require a huge amount of time and resources including equipment to set up a network as well as different computer systems to achieve it. The infected host typically has valuable information that should gathered by the forensic investigator. The information that gathered in the laboratory environment should not have much difference from the information that gathered in a real incident.

3.2.1 Summary Of Similar Studies

The four similar studies have reviewed in section 3.1 and the information that gathered from the previous studies used to identify guidance for doing research in this area. Shahrestani, Feily, Masood & Muniandy (2012) have monitored the traffic of a network to detect the behaviour of the bot while it is in the network. They have analysed the incoming and outgoing traffic in regards to the speed of the botnet communication packets. In addition, they believe that the command and control server should send a command of a size of 1KB or less to the bots in the infected host. Their focus was to prove the existence of the bot in the infected host, which means that the detection of the bot is not their aim and they did not expect to achieve that.

The second study has done by Etemad & Vahdani (2012), and their study focuses on the protocols of the communications between the bot in the infected host and the command and control server. They have analysed the two most common protocols (IRC and HTTP) that have used in the communication between the bot and the C&C server. Their proposal is aiming to filter the normal traffic from the malicious traffic in a real network environment for the inbound and outbound traffic. The proposal aims to allow the normal traffic to go through while the malicious traffic dropped.

The third study has done by Colajanni, Gozzi & Marchetti (2008), and they have used the honeypot in their project in order to collect different type of malware. As there are many honeypot projects that have involved different types of honeypot, this study used a version of honeypot called Nepenthes. The honeypot that Colajanni, Gozzi & Marchetti (2008) used in their research, which had a number of sensors that records each malware and send it to the malware collector if it catches a new malware. Then the malware executes in a protected environment such as sandboxing.

The fourth study done by Sinha, Boukhtouta, Belarde & Debbabi (2010), they have used one of the most popular botnets, which is Mariposa. The malware have infected more than 8 million hosts. Analysing such a popular botnet is a challenge that Sinha, Boukhtouta, Belarde & Debbabi took a step into finding out the secret of infecting a large number of machines. The aim for them was to study the behaviour of the Mariposa Botnets as well as the analysing the code for it.

They have analysed the Mariposa botnets statically and dynamically due to the reverse engineering.

3.2.2 Review Of The Problem Areas

Chapter 2 has discussed some of the issues and the problems that face most of the research in this area in section 2.6, as well as the section 2.5 discusses investigating botnets such as the detection of the existence of the botnets. The honeypot is one of the safe environments to collect malware so the malware does not spread throughout the network. As a forensic investigator, the first step to done when arriving to an incident is collecting information from the memory of the machine. The reason for that is that there is valuable information that stored temporary in the memory of the machine and once the machine turns off all the information in the memory will be gone and impossible to retrieve. Therefore, the forensic investigator needs to know what information to look for when examining the memory of the machine. The physical memory of the infected host will support the evidence that gathered is from the entire host; therefore, the entire host of the machine will be examined.

In addition, the propagation of the botnets is another challenge as they have different techniques to spread out the malicious software. One of the techniques that C&C server uses is a method called fast-flux, which changes the IP address of the server when the IP of the server identified. In addition, the signature based method of detecting the botnets is not an efficient method of detecting the botnets as they have a new signature, which may identified once the botnets has made damage to the system. What is more, the botnets behaviour inside the network is another issue that face the researchers, as they need to be analysed to discover them from the normal network traffic. Monitoring the traffic of the network is a huge amount of work that in order to be able to detect the existence of the botnets.

3.2.3 The Research Question & Hypothesis

The aim of this research is to identify the digital evidence in the infected host that is stored in the machine. In addition, the researcher will provide a proposal of the procedure for digital forensic investigation that is involving botnets.

The main research question based on the problem as well as the aim for this research presented as follows:

Q: What is the digital evidence that can be gathered from the infected-host in a botnet event?

After the main question there are sub questions that will assist the research to answer the main question.

Sub Question 1 (SQ1)

How many bots binaries were downloaded during the malware collection?

Sub Question 2 (SQ2)

Does the physical memory of the infected host contain any information in regards to the botnets event?

Sub Question 3 (SQ3)

Can the information of the physical memory be gathered and preserved?

Sub Question 4 (SQ4)

How the behaviour of the bot can be detect in the infected-host

Sub Question 5 (SQ5)

What is the behaviour of the bot inside the network of an infected-host?

Sub Question 6 (SQ6)

What is the suspicious activity of the command and control that can be found in the network traffic?

Sub Question 7 (SQ7)

Is the command and control instructions set encrypted?

Sub Question 8 (SQ8)

Is the command and control attack instructions set encrypted?

Sub Question 9 (SQ9)

Has the research been able to capture any sensitive information sent to the C&C server?

From the research sub-questions. Hypotheses are established accordingly as follows:

Hypothesis 1 (H1):

The infected host contains the information that was changed after the malicious activity of the infected machine.

Hypothesis 2 (H2):

The researcher's network has vulnerabilities that allowed the botnets to be downloaded.

Hypothesis 3 (H3):

The host is infected and it contains the information about the C&C server.

Hypothesis 4 (H4):

The bot in the infected host communicates with the command and control channel

3.2.4 Research Phases

The research proposal has divided into 5 phases. The Figure 3.5 shows the 5 phases that will use in this research in order to achieve the goal of the research.

The first phase that this research will take is to build up a database of the signature of the malware that have collected. The information about the malware that has collected and the result of the investigation of the dynamic analysis will perform by using an external service provider.

The second phase that this research will take is to select appropriate forensic tools from the previous studies, as well as to preserve the possible evidence in an infected host as forensic evidence of an incident. This step is mainly for the acquisition and preservation of the possible evidence in an infected host.

The third phase that this research will take is that the analysis of the malicious binaries of the botnets. In this steps the extraction of the data after the malicious binaries have identified for its activity in order to perform a static and dynamic method of analysis in infected host. The images that have taken from the infected host will be analysed forensically. The forensic investigator needs to classify whether if the malicious malware is involved or not.

The fourth phase that this research will take is that the dynamic analysis could assist the forensic investigator identify how the malicious malware is involve in the incident. In addition, the forensic investigator may analyse the memory of the infected host that may provide information about the incident. In addition, a live monitoring of the infected host will perform using malware tools, sniffer tools to be able to identify the C&C server communication existence in the infected host.

The fifth phase that this research will take is that presenting all the information about the investigation process and procedure that has taken during this research. The location of the evidence should identify as well as the type of the information to achieve the goal of this research.

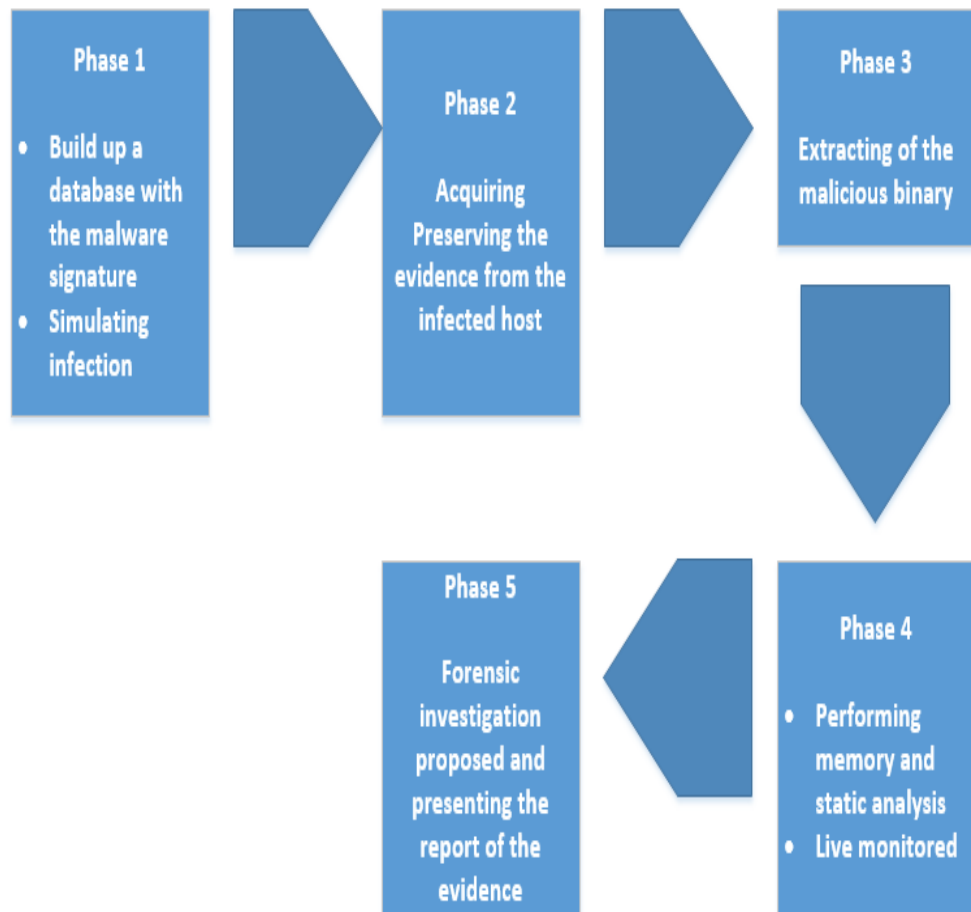


Figure 3.5 Research Phases

3.2.5 Data Map

The Data map has been presented on page 58.

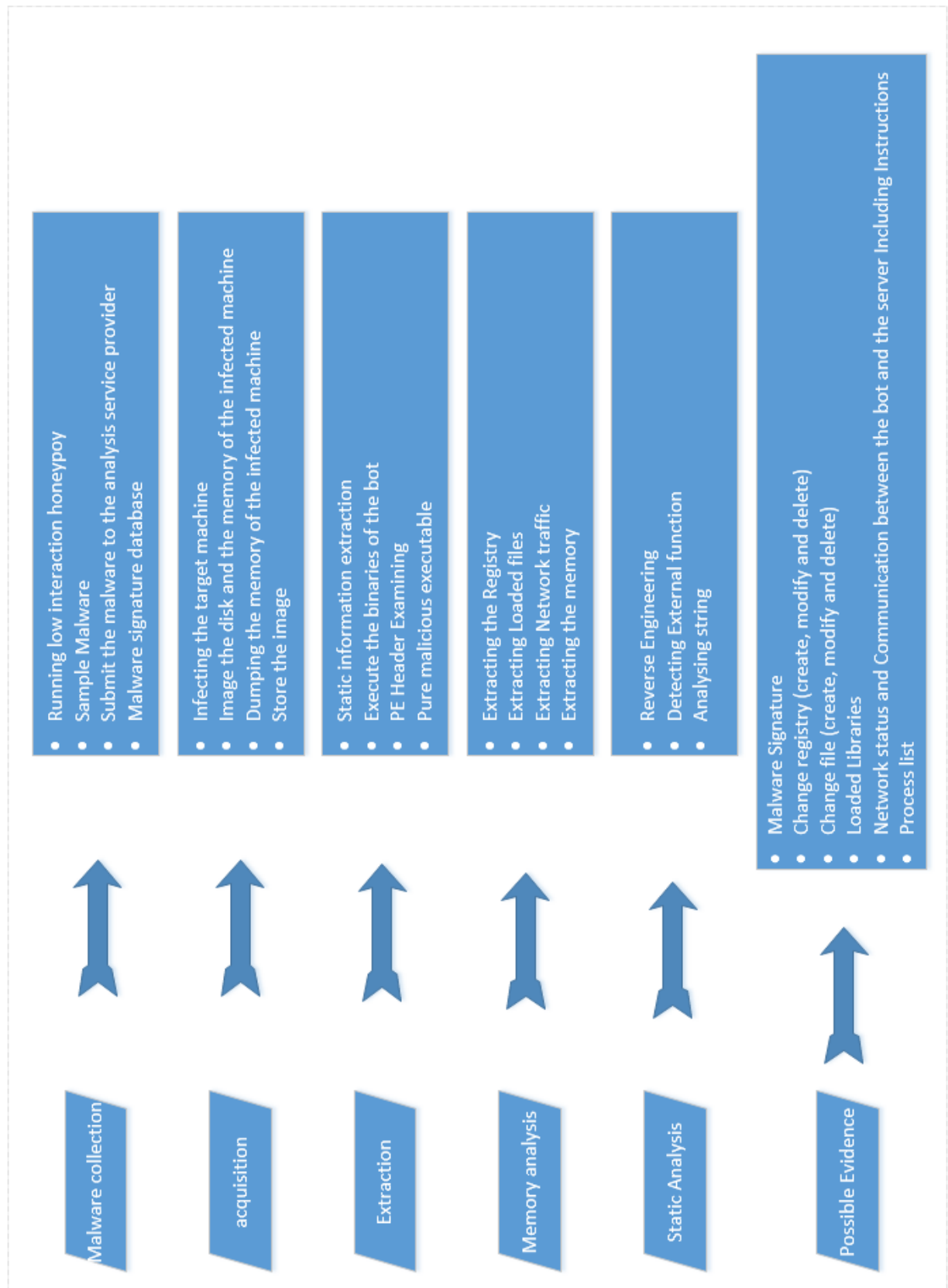


Figure 3.6 Proposed Research Data Map

3.3 DATA REQUIREMENTS

There are several sources of data, which are required for the proposed research including setting up the network and collecting malware. Section 3.5 will discuss the techniques that use to collect the data.

3.3.1 Data Type

This research will be looking mainly for three types of data. The first data type that this research will be looking for is the malware signature and building up a database that contains all the malware that has collected during the experiment of this research. The malware collection will contain the botnets signature as well as the other malware signature and the primary focus of this research will be the botnets malicious signature. The botnets signature will be determined using an external service that will provide the researcher whether the signature is relating to a botnets or bot.

In addition, this research will be looking for all the possible digital evidence in the infected host in regards to the botnets event. This research is focusing on investigating the infected host from a forensic point view, which means that all the possible evidence must be preserve and this type of data is one of the goals that this research needs to achieve.

The third type of data that the C&C server instructions between the C&C server and the infected host. The botnets typically uses the C&C server to perform malicious activities using the infected hosts. The isolated internet server connection will required to perform this step. However, the internet connection will be available for a short period to be able to capture the joining stage of the botnets army and connecting the C&C server. The internet connection then disconnected to disabled the abilities of harming other people.

3.3.1.1 Malware Signature

This research will be collecting malware by the signature of the malware. The reason for that is that the aim for this research is to examine the activity of the malware in an infected host. Most of the botnets attacks are targeting organizations and businesses to gain a financial benefit, therefore, the botnets activities localized and have a specific target (NCSC, 2012). The forensic investigator can use several information sources in order to detect an existence of

the botnets. In addition, the malware collection system may find one of the malicious malware have a similar signature of different malware.

The honeypot generates information in regards to the malware attacks and stores it in the system. In addition, the system stores all the logged remote access into the database system. The MD5 hash values of the malware identified by the honeypot and downloads, which the binaries of the malware in order to submit into the external analysis provider.

3.3.1.2 Digital Evidence

The digital evidence is important information that the forensic investigator needs to store for the procedure of the investigation. The information that the forensic investigator collects from the infected host is able to provide the cause of the incident after analysing the malware in an infected host. There are some challenges that the forensic investigator may experience during the collecting of the possible evidence for instance the information in the physical memory needs to be examined with a special care to preserve the evidence of the incident especially the physical memory of the infected host.

In addition, the research will be looking for possible communications between the C&C server and the infected host. The instructions that expected to capture the joining stage of the infected host. The instructions of performing attacks will not use to harm other people.

3.3.2 Data Collection

This section will discuss the collection of the data that this research will use to analyse the malware. There are different tools that need to use in this research in order to carry out the research and they describe in this section.

3.3.2.1 Laboratory Environment

The implementation of the laboratory environment in this research based on a physical machines and Virtual machines (VMs). The software that will use in this research will provide an efficient way to analyse the botnets and provide a flexible method to deploy a botnets laboratory analysis. The laboratory environment will be composed of several computers. The main reason for this research to use a VMs environment is that the physical computers to analysis the botnets will

increase the cost of the research. The purpose of having a VM is to be able to restore the computer into its original state in the case of the VM infected. This will reduce the time of the researcher repeating the experiment multiple times. In addition, some of the malware does not executed in a VM environment, which means the physical computers that, is host with Linux operating system can use to execute this type of malware. This will provide the research with a high accuracy of results and safety.

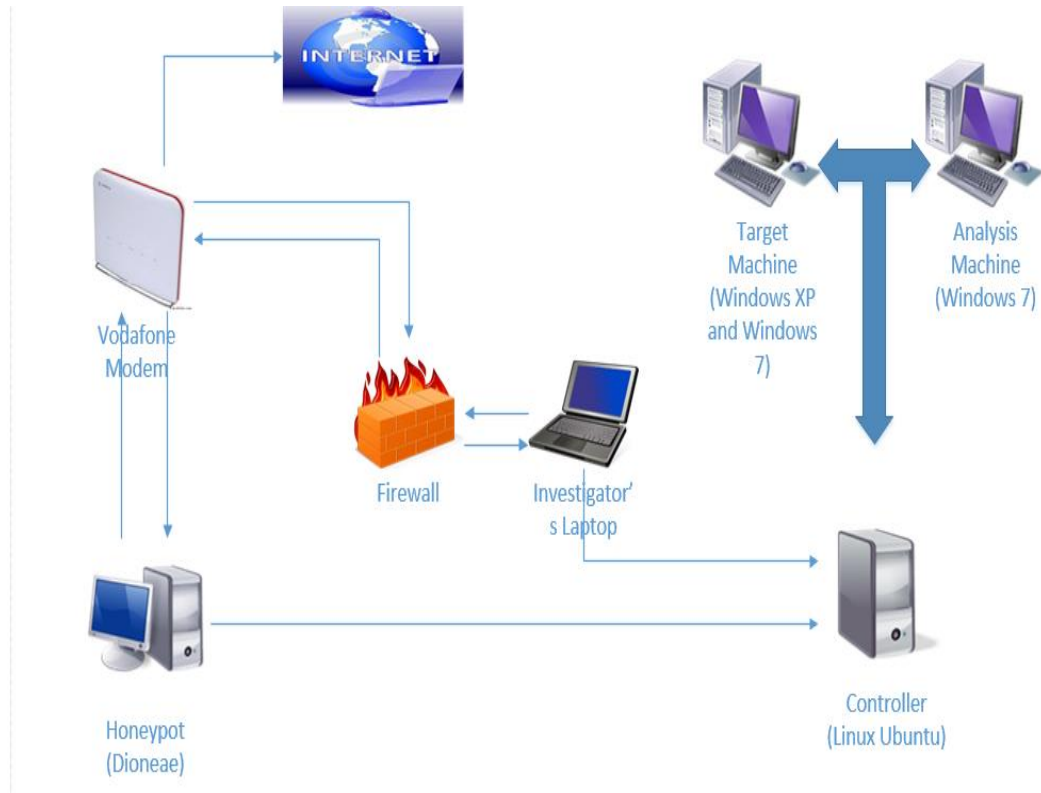


Figure 3.6 Laboratory Component

As this research will be analysing a malicious activity this means that the research's network will be considering the security of the other machines connecting to the research's network. The researcher will use DMZ (Demilitarized Zone) to forward the malicious traffic to the experimental machine and everything will be in isolation from the university network.

3.3.2.2 Laboratory Component

This section will describe the components in the laboratory that for the experiment of this research. There are physical computers that contain the host operating system of Linux. Also the Linux operating system contains a Virtual operating

system of Windows XP. The target in this experiment will be the physical and the virtual machine, the controller installs in the Linux operating system where the honeypot installed as well. There will be tools that will use to carry out the static analysis.

- Honeypot: The tool that the research has used in order to collect the malware samples from the internet. This tool is the safest way to collect the malware. The physical computer directly connected to the internet to expose the vulnerabilities. The honeypot that has chosen for this research is Dionaea, which is a low interaction honeypot.
- Controller: This Linux based operating system runs multiple tools and software to monitor the activities on the virtual windows machines. The Controller will be monitoring the network activity as well as simulate the network access. The controller will have a database that contains all the signatures of the malware.
- Virtual Target: This is Windows 7 based virtual machine malware analysis tools, which used to examine the bot.
- Physical Target: This is where the bot executed in the Windows XP and Windows 7 where the hard disk of the machine formatted in order to use it for the forensic imaging. The hard disk is working fine and the operating system installed properly in the machine. In addition, the machine monitored to capture the traffic of the suspicious C&C server instructions.
- Static analysis: This is a Virtual machine that Windows XP and Windows 7 will be installed in it, and the static analysis will be carried out and the memory will be examine by the forensic investigator. This research will be carrying on the static analysis separated from the dynamic analysis and the reason for that is that most of the reverse engineering tools support windows based operating system s only.

3.3.3 Data Processing

One of the aims of this research is have a collection of unique signatures of the malware from the internet and create a database that contains all the malware signatures found during the experiment. Another part of this research is to investigate the botnet event. The investigation carried out in four steps in order to

get high accuracy. There will be an acquisition, extraction and memory, and static analysis.

The first phase that this research will take is to build up a database of the signature of the malware that has collected through Dionaea Honeypot. The information about the malware that has collected and the result of the investigation of the dynamic analysis, that performed by using an external service provider.

The second phase is to select appropriate forensic tools from the previous studies, as well as to preserve the possible evidence in an infected host as forensic evidence of an incident. This step is mainly for the acquisition and preservation of the possible evidence in an infected host.

The third phase is that the analysis of the malicious binaries of the botnets that downloaded through Dionaea honeypot. In this step the extractions of the data after the malicious binaries have identified and its activity in order to perform static and dynamic methods of analysis in an infected host. The images that have taken from the infected host will be analysed forensically. The forensic investigator needs to classify whether if the malicious malware is involved or not.

The fourth phase is that the dynamic analysis could assist the forensic investigator to identify how the malicious malware is involved in the incident. In addition, the forensic investigator may analyse the memory of the infected host that may provide information about the incident. In addition, a live monitoring of the infected hos performed using malware tools, sniffer tools to be able to identify the C&C server communication existence in the infected host.

The fifth phase is that presenting all the information about the investigation process and procedures that have taken during this research. The location of the evidence identified as well as the type of the information to achieve the goal of this research.

3.3.4 Data Analysis

The analysis for this research will be performing a memory analysis and a static analysis. The reason for this research to take them both is that the host including memory contains valuable information about the incident that involves botnets, however, the memory will not provide all the information needed for the incident to investigate. Therefore, the host analysis, the memory analysis and the static

analysis is needed both to be able to have a full picture about the incident and the cause of the incident. In addition, the suspicious C&C server instructions from and to the infected host collected.

3.3.4.1 Memory Analysis

In this step, the analysis of the host of the infected host examined and all the processes that are running in the physical memory displayed with open registry, open files and loaded libraries. The memory image of the infected host taken. In this research, the memory will be analysed using the Volatility Framework.

3.3.4.2 Static Analysis

The static analysis is an important step for this research as this research will be analysing and executing the binary code of the botnets. The reason for that is that executing the binary code of the botnets will provide a better understanding of the techniques and functionality of this malicious malware. Most of the botnets involve reverse engineering techniques, therefore, this research needs to use a reverse engineering tools to analysis the malicious code to analysis the data structure and extract readable string.

3.3.4.3 Analysis tools

(The Purpose of the tools have been quoted from the Vendors' Websites)

Type	Name	Purpose
Malware collection	Dionaea	A low interaction honeypot that collects a sample of the malware around the network. The honeypot exploits the vulnerabilities in order to collect the malware.
Virtualization	VMware workstation	Tools for Visualizing the computer system
Memory Analysis	Volatility Framework	A forensic tool that extracts the information in the memory image.

Initial Virus Scan	VirusTotal	A public service for analysing the suspicious files and URLs.
Initial Sandbox scan	Anubis, ThreatExpert	Public service that analyse the behaviour of the malware.
Packer Detectors	PEiD v 0.94	A tool that detects packers and cryptors.
String Extractor	BinText v3.03	Free tool from McAfee to find ASCII, Unicode and resource strings in a file
Disassemblers and Debuggers	IDA Pro	Tool for reverse engineering
Control DNS Responses	ApateDNS	for controlling DNS responses
The Sysinternals Troubleshooting Utilities	SysinternalsSuite	Troubleshooting Utilities by Microsoft
Registry compare utility	Regshot	an open-source (LGPL) registry compare utility
Network utility	Netcat	Is a featured networking utility which reads and writes data across network connections, using the TCP/IP protocols

Table 3.1 Analysis Tools

3.4 LIMITATIONS

The aim of this research is to achieve the goal and answer the research questions. However, the botnet field is a large area of study and this research will not be able to cover everything about the analysis of the botnets. The limitation of the proposed research stated in this section.

The first limitation in this research is that the honeypot that is been chosen for this research is a low interaction honeypot. There are differences between low interaction honeypots and high interaction honeypots. The security risk of the low interaction is less than the high interaction. In addition, in regards to the deployment and the maintenance is less in low interaction honeypots than the high interaction honeypots.

This research aims to run the experiment in a safe environment, which means that the network will not be effected by this malicious malware; therefore, the experiment will be running in isolation and with the connection to the internet in an isolated server only. The connection outside of the secure network prevented by physical isolation. However, the activity between the bot and the C&C server needs to monitor to achieve one of the research goals; therefore, the controller will monitor this activity by a software that installed in it. The software will be able to locate the origin of the server.

Another limitation of this research is that most of the analysis tools provided for windows based systems. Therefore, this research will be using some selected tools in order to analyse the memory of the infected host. This research chooses to have the experiment in Windows XP even though it is an older version of Windows because the computers that provided do not support the new version and most of the analysis tools support windows mainly, therefore, Windows XP and Windows 7 were the best option for this research.

This research will not be analysing the botnet code. The reason for that is that analysing the code of the botnets is a time consuming to be able to understand the concept of the code as well as the code of the botnets is usually thousands of lines. The time that is provided to submit this research is limited, therefore, analysing the code of the botnets will need to be longer to be able analysis the host as well as the code of the botnets. Therefore, the depth analysis of the code of the botnets will not be included in this research.

3.5 CONCLUSION

This Chapter 3 has presented the methodology that the research will take in order to achieve the goal. This research states earlier that any infected host by the malicious botnets activity provides valuable information that the forensic investigator needs to examine in order to preserve the evidence out of the

machines of an infected host. Therefore, the aim for this research is to investigate the botnet event in the infected host side, as well as preserve the evidence and the valuable information in the memory of the infected host.

The research question for this research has identified and the sub questions of the research have identified to be able to investigate the infected host of the botnets event forensically. The answer for the research question and the sub questions achieved after the experiment of this research has carried out and the malware collection and malware analysis has performed.

The infected host of the botnets event connected to the internet for security reasons; however, the laboratory environment has been set up in using standard forensic procedures in order to perform the forensic investigation. The results of the testing and investigation reported in the next chapter five.

Chapter 4

Research Findings and Analysis

4.0 INTRODUCTION

Chapter 4 reports the findings of the experiment defined in chapter 3. The previous chapter derived the research plan to investigate and analyse the malware in the infected host using the low interaction honeypot called “Dionaea”. The machine that collected the malware was running for 22 days, and the binaries of the malware downloaded into a separate file with the unique MD5. The forensic investigator performed a host investigation to locate any possible evidence related to the botnets. The findings reported in this chapter. There are two bots that have been selected because of their behaviour is similar to the other. The two IRC bots that selected are 0a278f8d72e4d3d2d44485764398c84d and a650c67e14cfb27879999036741478d5.

4.1 VARIATIONS ENCOUNTERED

This section 4.1 will provide information about how the experiment defined in chapter 3 had to alter in practice so that the data collected.

4.1.1 Data Collection

The collection of malware used in this research, have downloaded through the Low interaction honeypot, which called Dionaea Honeypot. The Dionaea installed in the physical machine that has Linux Ubuntu operation installed in it as a host operating system. The reason for choosing Linux Ubuntu is that the Dionaea honeypot has tested and implemented for Linux Ubuntu, which improves the functionality of the honeypot. In addition, not having to deal with any technical issue that would face the research if using another operating system. The Dionaea honeypot was installed in the physical machine and was running for 22 days, and had more than 1000 attacks. The Dionaea used using the DMZ (Demilitarized Zone) that protects the other machines that connected to the researcher’s network from attacked and protected from spreading the malware throughout the researcher’s network. The malware kept in a file that called Binaries with the

name of each binary with its MD5 values to make it easier for the user to identifying each binary.

The Windows XP and Windows 7 that have used in this research to be infected used as Virtual Machines using VMware Workstation. Windows, which are in this case (Windows XP and Windows 7) were both infected with each of the malware that have been downloaded using the Dionaea honeypot and the behaviour of the Windows Operating system was monitored. All the changes that have occurred to the Windows Operating system have collected. Then compare with its original statues to be able to determine the changes that have performed in the Windows Operating system.

In addition, another Windows host operating system has been installed in a different physical machine to be able to analyse the malware using the malware analysis tools that is been implemented for Windows users. The purposes of having another Windows Physical host it to be able to analyse the malware with a host that is not infected and to be able to have deeper details about each malware. The purpose of this research is to study the bot; therefore, the analysis of the other malware might be present but the result of its analysis not to presented in this chapter 4.

4.1.2 Data Processing

The Hardware write-blocker used to copy all the files of the infected host to the researcher Windows analysing VM machine. The malware binaries that have used in this research have downloaded through Dionaea honeypot. Then, the malware binaries will be analysis using the external service to identify the bot from these malware binaries. The binaries of the bot infected to the Windows XP and Windows 7 VM to be able to study the behaviour of the bot in the infected host. The analysis of the infected host as well as the malware binaries using the malware analysis tools to be able to have a better understanding of the malware especially the botnets. Furthermore, all the evidence in the infected host preserved as a forensic requirement.

4.1.3 Data Analysis And Presentation

The screen shots of some of the results presented in this chapter 4 to provide information about the botnets. There are large amounts of repeated works that

have done in the infected host by the botnets. The most important results found in the experiment presented in chapter four.

4.2 MALWARE COLLECTION AND THE ANALYSIS OF THE MALWARE

This section will present the information that has gathered by the Dionaea honeypot that collected malware and download the binary of the malwares into a secure database with other information including the attackers' IP addresses.

4.2.1 Low Interaction Honeypot (Dionaea)

This research used the low interaction honeypot (Dionaea) to collect and capture malware. In addition, download the binaries of the malware in a separate file; each malware that Dionaea downloaded named with its MD5 and stored safely in a file. The malware that Dionaea download including all types of malware such as Virus, Worm and Trojan and so on.

Nepenthes developed by Markus Kotter and other developers and Dionaea is considered as the successor of nepenthes. Markus Kotter took a part of developing Dionaea as part of the Honey Project's Summer of Code 2009. Dionaea collect the samples of the malware and reply to the attacks over HTTP. Dionaea has been written in C and Python, however, it has a Python interface which means that any new modules can be developed without having to recompiling the base. In addition, Dionaea supports IPv6 and TLS and ultimately it logs all information about attacks on an SQLite3 database which makes it easier for the researchers to develop graph statistics. Dionaea uses port 445 and mail protocol is SMB and other protocols such as HTTPs, MSSQL, FTP, MYSQL and SIP. What makes Dionaea useful is that it takes the advantage of libemu to be able to analyse the shellcode of the malware. Libemu is a library that is small, written in C language that offers basic x86 emulation and shellcode detection using GetPC heuristics. The purpose of designing the Libemu is to use it in the honeypot and other network purposes (libemux86.emu, n.d.). It works by running the shellcode inside the libemu VM and API and then the call of it recorded. The information that provided by the Dionaea database will be useful for this research for further analysis of the bot. As Dionaea collects different types of malware,

therefore, by determining the MD5 of the malware the researcher will be able to distinguish the botnet signature to analyse further as this research focuses on the botnets.

Dionaea was connecting to the internet with a static IP to be able to receive high infection rate. The machine that has used for the experiment is located outside the network with the DMZ (Demilitarized Zone). The malware samples that will download through Dionaea will be analysed by an external service that will be able to analyse the malware samples and gives more information about the downloaded malware samples.

There are variety of sandboxes service that allows the user to execute the malware to have more detail about the malware. These sandboxes will provide information about the behaviour of the malware including the files that the malware will access once it gets the host of the victim infected. In addition, the sandboxes will provide information about the malware access to the network, crypto operations dynamic code loading and information leaks. Sandboxes will support this research by providing static analysis and dynamic analysis of the malware that will give the result of the researchers experiment.

Dionaea was running for 22 days and have downloaded 59 unique malware binaries downloads. The Dionaea honeypot dealt with many connections that has been either accepted or rejected. The malware binaries will keep in a secure file in the Dionaea honeypot that named binaries. Each of the binary will be kept in the binary file with the size of named as the unique MD5 hash value of the binary which is actually help to identify what type is it and makes it easier to analysis. The Dionaea also downloaded a binary of the malware that classified as unknown and the size of them is 0 bytes. This researcher was not able to analysis these malware binaries by either of the sandboxes that have used in this research and hence are not included in the report.

SHA256: 4e303131bf90b123f9c84340f9a14b8dd59df727a7d47a6b30779d8063407553
 File name: 0a278f8d72e4d3d2d44485764398c84d
 Detection ratio: 48 / 52
 Analysis date: 2014-05-13 04:32:54 UTC (1 hour ago)



Analysis File detail Additional information Comments 1 Votes

Antivirus	Result	Update
AVG	Win32/Virut	20140512
Ad-Aware	Trojan.Generic.5333379	20140513
Agnitum	Win32.Virut.Gen.4	20140511
AhnLab-V3	Win32/Virut.B	20140512
AntiVir	W32/Virut.AX	20140513
Avast	Win32:Virtob	20140513
Baidu-International	Virus.Win32.Virut.\$a	20140512
BitDefender	Trojan.Generic.5333379	20140513
Bkav	W32.VtLikeB.PE	20140512
CAT-QuickHeal	W32.Virut.E	20140513
CMC	Virus.Win32.Virut!O	20140512
ClamAV	Worm.Allapple-2	20140513
Commtouch	W32/Virut.7116	20140513
Comodo	TrojWare.Win32.Trojan.XPack.-gen1	20140513
DrWeb	Trojan.Starman.3913	20140513
ESET-NOD32	Win32/Virut.AV	20140513
Emsisoft	Trojan.Generic.5333379 (B)	20140513
F-Prot	W32/Sdbot.AEFV	20140513
F-Secure	Net-Worm:W32/Allapple.gen!B	20140513
Fortinet	W32/Virut.AV.gen	20140513
GData	Trojan.Generic.5333379	20140513
Ikarus	Net-Worm.Win32.Allapple	20140513
Jiangmin	Win32/Virut.af	20140512
K7AntiVirus	Virus (00001b781)	20140509
K7GW	Virus (00001b781)	20140509
Kaspersky	Backdoor.Win32.Rbot.adqd	20140513
Kingsoft	Win32.Virut.xf.57344	20140513
McAfee	W32/Virut.gen.a	20140513
McAfee-GW-Edition	Heuristic.BehavesLike.Win32.Suspicious-BAY.G	20140513
MicroWorld-eScan	Trojan.Generic.5333379	20140513
Microsoft	Virus:Win32/Virut.AC	20140513
NANO-Antivirus	Trojan.Win32.Rbot.vkyds	20140513
Norman	Allapple.gen10	20140512
Panda	W32/Virutas.FG	20140512
Qihoo-360	Backdoor.Win32.Bot.B	20140513

Rising	PE:Worm.Win32.Allaple.gp!1075352370	20140507
SUPERAntiSpyware	Trojan.Agent/Gen-Backdoor	20140513
Sophos	W32/Virut-W	20140513
Symantec	Backdoor.Trojan	20140513
TheHacker	Backdoor/Rbot.adqd	20140512
TotalDefense	Win32/Virut.7115	20140512
TrendMicro	PE_VIRUT.AV	20140513
TrendMicro-HouseCall	PE_VIRUT.AV	20140513
VBA32	Virus.Virut.07	20140512
VIPRE	Virus.Win32.Virut.a (v)	20140513
ViRobot	Win32.Virut.Gen	20140513
Zillya	Virus.Virut.Win32.24	20140512
nProtect	Virus/W32.Virut.Gen	20140512
AegisLab	✓	20140513
Antiy-AVL	✓	20140513
ByteHero	✓	20140513
Malwarebytes	✓	20140513

[Blog](#) |
 [Twitter](#) |
 contact@virustotal.com |
 [Google groups](#) |
 [ToS](#) |
 [Privacy policy](#)

Figure 4.1 The scan result of the Rbot from Virustotal

The binaries have been analysed by multiple external services such as virusTotal that provides a scan for the malware from different anti-virus engines Kaspersky and McAfee, as well as providing information about the malware from the Microsoft. The virusTotal search for information about the malware from up to 52 engines and provide the detection rate out of the 52 engines. VirusTotal is free online service that is able to scan a file or a URL to identify any possibility of malicious content (virusTotal, n.d.).

After the submission of the malicious file, virusTotal provides an overview of the file submitted that include the SHA256 and the MD5. The purpose of using the SHA256 and the MD5 is to use them to be able to find the information about the malware in the database. Furthermore, another purpose of using the SHA256 and the MD5 is that the virusTotal verify the submitted file to prevent any changes to the malware binary (virusTotal, n.d.). In addition, the detection rate and the analysis date is provided in the report that are shown in figure 4.1, The file that has been submitted to virusTotal shows that the detection rate of the Rbot are really high with the detection rate of 48 out of the 52 engines that the virusTotal searched. The Report in figure 4.1 shows that the name of the Rbot is different from one engine to another. Some of the engine shows that the

Rbot is a virus or Trojan and other shows that the file submitted is belong to Rbot. Four of the engines out of the 52, shows that the file is not malicious such as ByteHero and Malwarebytes.

The virusTotal has determined that the malicious malware classified as a win32 threat. VirusTotal used different types of tools that are able to gather information about the malware for different purposes. For example, the tools are able to determine some information structure about the Microsoft Windows portable executables (PEs) to be able to signed software that identified. .

4.2.2 Threat Expert

ThreatExpert is a public service that provides information about the behaviour of the malware. The ThreatExpert is advanced automated threat analysis system (ATAS), and the ThreatExpert reports the behaviour of the malware including worm, virus and Trojan in a fully automated mode. The ThreatExpert is a free service that allows uploading any samples to its database to be able to analysis it then reports the behaviour of the malware to the customers in just 2 to 3 minutes (ThreatExpert, 2009).

The ThreatExpert provides an important analysis step of any type of new malware that could threaten any computer system. The reason for that is that the anti-virus vendors could take up to 48 hours depending on the complexity of the malware to be able to analysis the malware and update their database in the customer's end. In this time it could infect many computer systems, however, not all the systems will be updated straight away after updating the database of the anti-virus vendors which could result of the infection being spread out to more victims that may result on loss of personal and businesses information. Therefore, the ThreatExpert provides information about the analysed malware in a few minutes, which saves more time analysing the malware and decreases the number of compromised systems (ThreatExpert, 2009).

Submission Summary:

Submission details:

- ▶ Submission received: 15 May 2014, 04:04:29
- ▶ Processing time: 8 min 13 sec
- ▶ Submitted sample:
 - └ File MD5: 0xA650C67E14CFB27879999036741478D5
 - └ File SHA-1: 0xA68EEA7FA0544B656E02E468C3A98437BAD7C52C
 - └ Filesize: 39,424 bytes
 - └ Alias:
 - └ Backdoor.Sdbot ▶ [Symantec]
 - └ Backdoor.Win32.IRCBot.jwy ▶ [Kaspersky Lab]
 - └ W32/Sdbot-DKI ▶ [Sophos]
 - └ Worm.Win32.Neeris ▶ [Ikarus]
 - └ Win32/Autorun.worm.32256.D ▶ [AhnLab]

Summary of the findings:

What's been found	Severity Level
A network-aware worm that uses known exploit(s) in order to replicate across vulnerable networks.	■■■■■■■■■■
MS04-011: LSASS Overflow exploit - replication across TCP 445 (common for Sasser, Bobax, Kibuv, Korgo, Gaobot, Spybot, Randex, other IRC Bots).	■■■■■■■■■■
Communication with a remote IRC server.	■
Contains characteristics of an identified security risk.	■■■■■■■■■■

Figure 4.2 The scan result of the IRC bot from ThreatExpert

The Figure 4.2 shows that the analysed report of the IRC bot from the ThreatExpert, The result shows that the MD5 of the malware (IRC bot) that has been analysed with the file size. Then, the ThreatExpert shows the alias of the malware from the biggest anti-virus vendors. The figure 4.2 shows also that the information that the ThreatExpert has found after analysing the malware such as the malware is able to duplicate itself after exploit itself in a network system. This means that the victims in the network will face a threat of having their machine

compromised. The malware will scan for vulnerabilities throughout the network to find more and more victims. One of the examples of the duplication of the network system and scanning for vulnerabilities is the attack that occurred to the Saudi Aramco that occurred from an external source and affected about 30,000 workstations. The malware was able to duplicate itself throughout the network and infect more workstations (Reuters, 2012). In addition, Figure 4.2 shows that the malware that has been analysed is using the TCP protocol to duplicate and is common for some of the IRC bots such as Blaster and Spybot. Then, ThreatExpert shows that this malware can communicate with the IRC server as well as the ThreatExpert report shows that it is been identified that some of the information in the malware contains a security threat. Overall, figure 4.2 shows that the security threat of the malware is a high level of security threat.

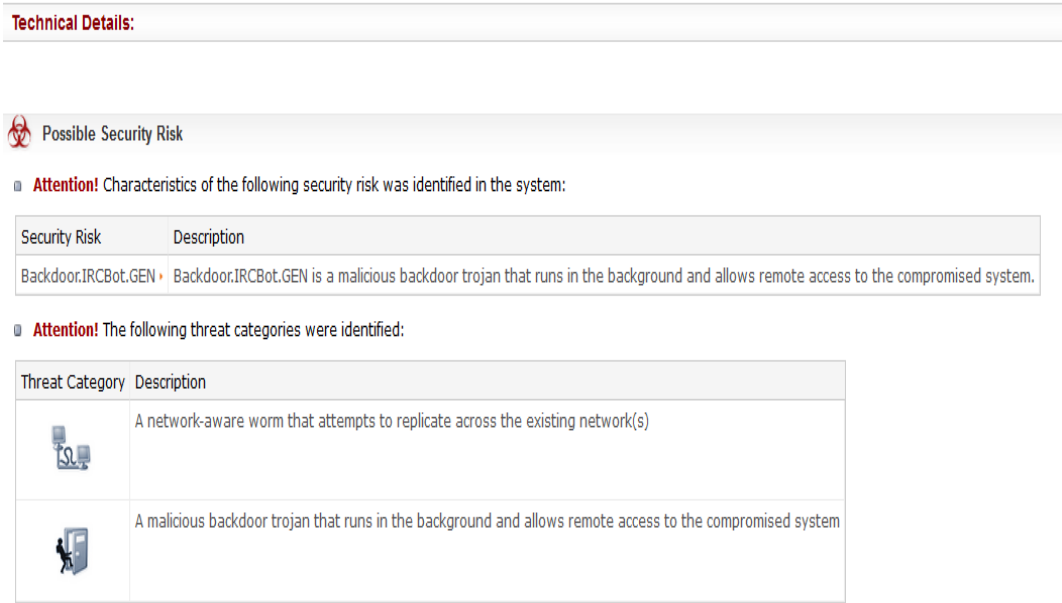


Figure 4.3 The scan result of the IRC bot from ThreatExpert

Figure 4.3 is the IRC bot that has been analysed in figure 4.3, which shows the IRC bot allowed remote access to the infected host to take control of the host. The Figure 4.3 shows also the malware that has been analysed, which able to duplicate itself as mentioned earlier as well as allowing the malware to be remotely controlled. This means that the botnet master is able to control the machine and perform any type of malicious activities without the knowledge of the owner of the machine.

File System Modifications

The following file was created in the system:

#	Filename(s)	File Size	File Hash	Alias
1	%System%\csrss.exe	39,424 bytes	MD5: 0xA650C67E14CFB27879999036741478D5 SHA-1: 0xA68EEA7FA0544B656E02E468C3A98437BAD7C52C	Backdoor.Sdbot • [Symantec] Backdoor.Win32.IRCBot.jwy • [Kaspersky Lab] W32/Sdbot-DKI • [Sophos] Worm.Win32.Neeris • [Ikarus] Win32/Autorun.worm.32256.D • [AhnLab]

Note:

%System% is a variable that refers to the System folder. By default, this is C:\Windows\System (Windows 95/98/Me), C:\Winnt\System32 (Windows NT/2000), or C:\Windows\System32 (Windows XP).

Figure 4.4 The scan result of the IRC bot from ThreatExpert

In addition, figure 4.4 shows that the file system changes on the infected host and the information about the file name, file size, file hash and the alias of the botnets from different security engines.

Memory Modifications

There were new processes created in the system:

Process Name	Process Filename	Main Module Size
csrss.exe	%System%\csrss.exe	589,824 bytes
[filename of the sample #1]	[file and pathname of the sample #1]	589,824 bytes
[generic host process]	[generic host process filename]	45,056 bytes

Notes:

[generic host process filename] is a full path filename of [generic host process].

There was a new service created in the system:

Service Name	Display Name	Status	Service Filename
WinSpoolSvc	Windows Spool Services	"Stopped"	"%System%\csrss.exe"

Figure 4.5 The scan result of the IRC bot from ThreatExpert

Figure 4.5 shows that the information of the new process that the ThreatExpert has found by analysing the malware. The figure shows that a new process has created with the process name, process filename and size of it in the victim's machine to be able to investigate it in the infected host. In addition, the report also shows that the original value of the registry key and the new value of registry key that has created by the malware.



Registry Modifications

- The following Registry Keys were created:
 - HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Enum\Root\LEGACY_WINSPOOLSVC
 - HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Enum\Root\LEGACY_WINSPOOLSVC\0000
 - HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Enum\Root\LEGACY_WINSPOOLSVC\0000\Control
 - HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\WinSpoolSvc
 - HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\WinSpoolSvc\Security
 - HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\WinSpoolSvc\Enum
 - HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Root\LEGACY_WINSPOOLSVC
 - HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Root\LEGACY_WINSPOOLSVC\0000
 - HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Root\LEGACY_WINSPOOLSVC\0000\Control
 - HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSpoolSvc
 - HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSpoolSvc\Security
 - HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSpoolSvc\Enum
- The newly created Registry Values are:
 - [HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Enum\Root\LEGACY_WINSPOOLSVC\0000\Control]
 - *NewlyCreated* = 0x00000000
 - ActiveService = "WinSpoolSvc"
 - [HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Enum\Root\LEGACY_WINSPOOLSVC\0000]
 - Service = "WinSpoolSvc"
 - Legacy = 0x00000001
 - ConfigFlags = 0x00000000
 - Class = "LegacyDriver"
 - ClassGUID = "{8ECC055D-047F-11D1-A537-0000F8753ED1}"
 - DeviceDesc = "Windows Spool Services"
 - [HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Enum\Root\LEGACY_WINSPOOLSVC]
 - NextInstance = 0x00000001
 - [HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\WinSpoolSvc\Enum]
 - 0 = "Root\LEGACY_WINSPOOLSVC\0000"
 - Count = 0x00000001


- NextInstance = 0x00000001
- [HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\WinSpoolSvc\Security]
 - Security = 01 00 14 80 90 00 00 00 9C 00 00 00 14 00 00 00 30 00 00 00 02 00 1C 00 01 00 00 00 02 80 14 00 FF 01 0F 00 01 01 00 00 00 00 00 01 00 00 00 00 02 00 60 00 04 00 00 00 00 00 14 00 FD 01 02 00 01 01 00 00 00 00 00 05 12 00 00 00 00 00 18 00 FF 01 0F 0
- [HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\WinSpoolSvc]
 - Type = 0x00000110
 - Start = 0x00000002
 - ErrorControl = 0x00000000
 - ImagePath = ""%System%\csrss.exe""
 - DisplayName = "Windows Spool Services"
 - ObjectName = "LocalSystem"
 - FailureActions = 0A 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 01 00 00 00 B8 0B 00 00
 - Description = "Windows Spool Services"
- [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Root\LEGACY_WINSPOOLSVCS\0000\Control]
 - *NewlyCreated* = 0x00000000
 - ActiveService = "WinSpoolSvc"
- [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Root\LEGACY_WINSPOOLSVCS\0000]
 - Service = "WinSpoolSvc"
 - Legacy = 0x00000001
 - ConfigFlags = 0x00000000
 - Class = "LegacyDriver"
 - ClassGUID = "{8ECC055D-047F-11D1-A537-0000F8753ED1}"
 - DeviceDesc = "Windows Spool Services"
- [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Root\LEGACY_WINSPOOLSVCS]
 - NextInstance = 0x00000001
- [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSpoolSvc\Enum]
 - 0 = "Root\LEGACY_WINSPOOLSVCS\0000"
 - Count = 0x00000001
 - NextInstance = 0x00000001
- [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSpoolSvc\Security]
 - Security = 01 00 14 80 90 00 00 00 9C 00 00 00 14 00 00 00 00 30 00 00 00 02 00 1C 00 01 00 00 00 02 80 14 00 FF 01 0F 00 01 01 00 00 00 00 00 01 00 00 00 00 02 00 60 00 04 00 00 00 00 00 14 00 FD 01 02 00 01 01 00 00 00 00 00 05 12 00 00 00 00 00 18 00 FF 01 0F 0
- [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSpoolSvc]
 - Type = 0x00000110

<ul style="list-style-type: none"> ▪ Start = 0x00000002 ▪ ErrorControl = 0x00000000 ▪ ImagePath = ""%System%\csrss.exe"" ▪ DisplayName = "Windows Spool Services" ▪ ObjectName = "LocalSystem" ▪ FailureActions = 0A 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 01 00 00 00 B8 0B 00 00 ▪ Description = "Windows Spool Services"
<ul style="list-style-type: none"> • The following Registry Values were modified: <ul style="list-style-type: none"> ○ [HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control] <ul style="list-style-type: none"> ▪ WaitToKillServiceTimeout = ○ [HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\SeviceCurrent] <ul style="list-style-type: none"> ▪ (Default) = ○ [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control] <ul style="list-style-type: none"> ▪ WaitToKillServiceTimeout = ○ [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\ServiceCurrent] <ul style="list-style-type: none"> ▪ (Default) = ○ [HKEY_USERS\DEFAULT\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders] <ul style="list-style-type: none"> ▪ Cookies = ▪ History =

Table 4.1 The scan result of the IRC bot from ThreatExpert

Table 4.1 shows that the new created values of the registry as well as the modified values in the infected host. The analysis of the IRC bot shows that this bot is capable of creating these new values in the infected host without the knowledge of the owner of the machine. The ThreatExpert presented in the table 4.1 all the new registry values created by the submitted IRC bot. It shows the path of the registry and the information about the key that created by the IRC bot. for example one of the registry keys is created in the path [HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\WinSpoolSvc] followed by the type and the start of the value key. The table 4.1 shows also the error control of 0x00000000 and the ImagePath of it which is ""%System%\csrss.exe"". In addition, the DisplayName that shows it is a Windows Spool Services, an ObjectName of LocalSystem, a FailureActions of 0A 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 01 00 00 00 B8 0B 00 00 00 and Description shows in the infected host as Windows Spool Services.

Table 4.1 shows that the list of modified values of the registry. The report of the ThreatExpert shows that the IRC bot is capable of modifying the value of the registry value in the infected host.


Other details

- To mark the presence in the system, the following Mutex objects were created:
 - Xx8K78xP
 - DesktopCleanupMutex
- The following Host Names were requested from a host database:
 - 127.0.0.0
 - 127.0.0.2
- There were registered attempts to establish connection with the remote hosts. The connection details are:

Remote Host	Port Number
127.0.0.2	445
127.0.0.3	445
127.0.0.4	445
127.0.0.5	445
127.0.0.6	445
127.0.0.7	445
127.0.0.8	445
127.0.0.9	445
127.0.0.10	445
127.0.0.11	445

Figure 4.6 The scan result of the IRC bot from ThreatExpert

Figure 4.6 show that the other details that have been analysed from the binaries of the IRC bot. The ThreatExpert report shows that the mtuex object Xx8K78xP and DesktopCleanupMutex created in the infected host. In addition, the IP addresses 127.0.0.0 and 127.0.0.2 requested from the host names. What is more, there were attempts to establish a connection with the remote host, the report of the ThreatExpert shows that the IP addresses from 127.0.0.2 to 127.0.0.11 were the IP addresses that the IRC bot server attempted to establish a connection using the port 445. As Dionaea Honeypot uses the port 445, this means that the IRC bot attempted to uses the vulnerabilities of 445 that Dionaea honeypot set it up as a vulnerability to be able to communicate and reply to the attack using this port.

4.2.3 Anubis



Analysis Report for 0a278f8d72e4d3d2d44485764398c84d

MD5: 0a278f8d72e4d3d2d44485764398c84d

Summary:












Description	Risk
Write to foreign memory areas: This executable tampers with the execution of another process.	 high
Changee Windows Firewall settings: This executable changes some settings of windows firewall.	 high
Performs Address Scan: The executable scans a range of IP Addresses. In most cases these scans identify more potential vulnerable targets.	 high
Performs File Modification and Destruction: The executable modifies and destructs files which are not temporary.	 low
AV Hit: This executable is detected by an antivirus software.	 high
Packed Binary: This executable is protected with a packer in order to prevent it from being reverse engineered.	 medium
Autostart capabilities: This executable registers processes to be executed at system start. This could result in unwanted actions to be performed automatically.	 medium
Changes security settings of Internet Explorer: This system alteration could seriously affect safety surfing the World Wide Web.	 low
Creates files in the Windows system directory: Malware often keeps copies of itself in the Windows directory to stay undetected by users.	 medium
Modify system files: This executable modifies files in the windows system directories.	 medium
Performs Registry Activities: The executable creates and/or modifies registry entries.	 low

Figure 4.7 The scan result of the IRC bot from Anubis

The Anubis developed by the international Secure Systems Lab with the professional security of small number of them whom are interested in security and analysing malware. Their aim to provide a free service that involves tools that are useful for advanced computer users to be able to gather information and analysing malware to learn and have more knowledge about malware. The aim of the Anubis is to analyse the malware and the behaviour of Windows PE-executables. The tool in Anubis provides information about submitted binaries in a report that is useful for humans to learn about the malware. The information that generated the report includes details about the modified data in the registry or the file system, as well as the information about the process (Anubis, 2014). The report from one of the malware that has been analysed by Anubis shows that the IRC bot is capable of performing different types of activities in the host side. Figure 4.7 shows that this IRC bot is able to write in the infected host's memory, change the firewall setting as well as performing a scan of the IP address with a risk being set to high. In addition, the report shows that the AV anti-virus is set to be high in regards to the risk level. This means that the anti-virus will not be able to detect this IRC bot. What is more, the report shows that the binaries are packet binary, which means the binary is ant-reverse engineering with the risk being medium. In addition to auto-start capabilities, create files in the windows system directory and modify system files with the risk being medium. Furthermore, There are three low risks that the report shows which are performs file modification and destruction, changes security setting of Internet Explorer and performs registry activities.

The report by the Anubis form the binary of the MD5: 0a278f8d72e4d3d2d44485764398c84d that contains 42 pages of report includes the dependency overview with the 19 exe files being analysed which shows the registry activities, file activities and network activities for each of the dependency. The report shows more of what this bot binary is capable of, each binary that has downloaded by the Dionaea honeypot have been analysed by the Anubis.

Dependency overview:

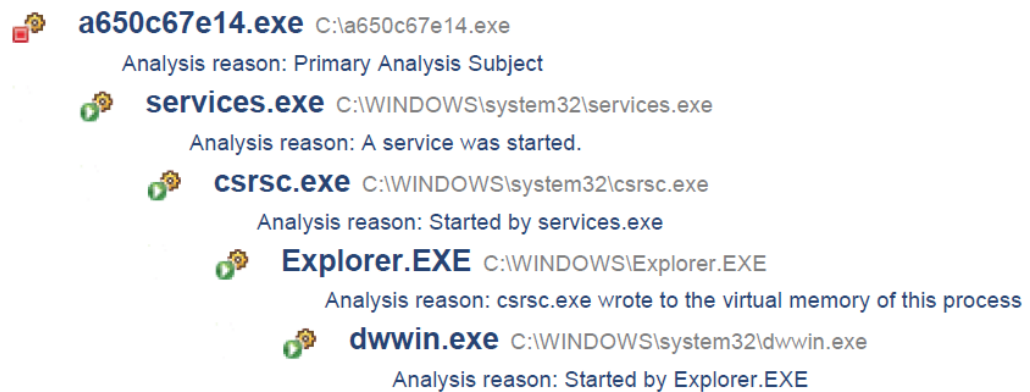


Figure 4.8 The scan result of the IRC bot Dependency from Anubis

Figure 4.8 shows the dependency with one primary dependency and six dependencies that have found inside the primary binary that has been analysed. The report shows that this particular IRC bot is able to perform many activities in the infected host to be able to gain control of the host as well as staying undetected and anonymous in the infected host. Figure 4.8 shows that there are five of dependency that are targeting svchost.exe; svchost.exe is a process that runs internal windows service as there are many services that runs in windows operating system . Therefore, having five dependencies make sense in botnets developers' prospective as it guarantees to get most of the infected host to be able to perform the activities by the botnet master.

Load-time DLLs		
Module Name	Base Address	Size
C:\WINDOWS\system32\ntdll.dll	0x7C900000	0x000AF000
C:\WINDOWS\system32\kernel32.dll	0x7C800000	0x000F6000
C:\WINDOWS\system32\advapi32.dll	0x77DD0000	0x0009B000
C:\WINDOWS\system32\RPCRT4.dll	0x77E70000	0x00092000
C:\WINDOWS\system32\Secur32.dll	0x77FE0000	0x00011000
C:\WINDOWS\system32\msvcrt.dll	0x77C10000	0x00058000
C:\WINDOWS\system32\shell32.dll	0x7C9C0000	0x00817000
C:\WINDOWS\system32\GDI32.dll	0x77F10000	0x00049000
C:\WINDOWS\system32\USER32.dll	0x7E410000	0x00091000
C:\WINDOWS\system32\SHLWAPI.dll	0x77F60000	0x00076000
C:\WINDOWS\system32\wsock32.dll	0x71AD0000	0x00009000
C:\WINDOWS\system32\WS2_32.dll	0x71AB0000	0x00017000
C:\WINDOWS\system32\WS2HELP.dll	0x71AA0000	0x00008000
C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83\comctl32.dll	0x773D0000	0x00103000
C:\WINDOWS\system32\comctl32.dll	0x5D090000	0x0009A000

Figure 4.9 The list of loaded libraries from Anubis

Figure 4.9 and 4.10 shows the information about the libraries affected by the bot. Figure 4.9 shows the list of the loading libraries in the infected host. Figure 4.10 shows that list of the running libraries in the infected host. The developer of the malicious malware uses these external libraries to improve the functionality of the infected host.

Base	Size	Entry	Name	File version	Path
00400000	00090000	004010F7	a650c67e		C:\Users\SuSu\Desktop\bin\a650c67e14cfb27879999006741478d5.exe
71A00000	00007000	71A01120	wsock32	6.1.7600.16385	C:\Windows\system32\wsock32.dll
75E70000	00040000	75E77DE0	KERNELBA	6.1.7600.16385	C:\Windows\system32\KERNELBASE.dll
75EC0000	0000A000	75EC136C	LPK	6.1.7600.16385	C:\Windows\system32\LPK.dll
75ED0000	000A1000	75F02433	RPCRT4	6.1.7600.16385	C:\Windows\system32\RPCRT4.dll
760C0000	00C4A000	76141681	shell32	6.1.7601.17514	C:\Windows\system32\shell32.dll
76F70000	00057000	76F89BA6	SHLAPI	6.1.7600.16385	C:\Windows\system32\SHLAPI.dll
76FD0000	000A0000	76FE49E5	advapi32	6.1.7600.16385	C:\Windows\system32\advapi32.dll
77070000	0001F000	77071355	IMM32	6.1.7601.17514	C:\Windows\system32\IMM32.DLL
77090000	00019000	77094975	sechost	6.1.7600.16385	C:\Windows\SYSTEM32\sechost.dll
77140000	000AC000	77140472	msuvert	7.0.7600.16385	C:\Windows\system32\msuvert.dll
771F0000	000C9000	7720D711	USER32	6.1.7601.17514	C:\Windows\system32\USER32.dll
772C0000	000CC000	772C168B	MSCTF	6.1.7600.16385	C:\Windows\system32\MSCTF.dll
773E0000	0004E000	773E9C09	GDI32	6.1.7601.17514	C:\Windows\system32\GDI32.dll
77430000	000D4000	7747BDE4	kernel32	6.1.7600.16385	C:\Windows\system32\kernel32.dll
77630000	0009D000	77663FD7	USP10	1.0626.7601.175	C:\Windows\system32\USP10.dll
77A70000	0013C000		ntdll	6.1.7600.16385	C:\Windows\SYSTEM32\ntdll.dll
77BC0000	00006000	77BC1782	NSI	6.1.7600.16385	C:\Windows\system32\NSI.dll
77BE0000	00035000	77BE145D	WS2_32	6.1.7600.16385	C:\Windows\system32\WS2_32.dll

Figure 4.10 The list of loaded libraries

Running the libraries would enable the botnet master to run the infected host in a standalone mode, which called static linking. In addition, the decrease of the binaries size is another approach that the developer of the botnets is aiming, therefore, the static linking will achieve this goal by using the libraries.

Figure 4.11 shows the Registry that has created in the infected host as well as the modified value of the Registry. In addition, the report shows that this IRC bot read 51 of the registry have affected by the IRC bot and either have created or modified. The binary of the IRC bot programmed to perform these actions to be able to get a full control of the host and to be able to be remote control by the botnet master.

```

HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\CurrentVersion\Internet Settings
HKEY_CURRENT_USER\SOFTWARE\Policies\Microsoft\Windows\CurrentVersion\Internet Settings
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings
HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Internet Explorer\Main\FeatureControl
HKEY_CURRENT_USER\Software\Policies\Microsoft\Internet Explorer\Main\FeatureControl
HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main\FeatureControl
HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\Main\FeatureControl
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Cache
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache
HKEY_LOCAL_MACHINE\System\Setup
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Content
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Content
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Cache\Paths
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Cache\Paths\Path1
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Cache\Paths\Path2
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Cache\Paths\Path3
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Cache\Paths\Path4
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Cache\Special Paths
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Cookies
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Cookies
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\History
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\History
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012013041020130411
HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\CurrentVersion\Internet Settings
HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_AUTOPROXY_CACHE_ANAME_KB921400
HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_TEMPORARYFILES_FOR_NOCACHE_840387
HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_TEMPORARYFILES_FOR_NOCACHE_840386
HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\Main\FeatureControl\RETRY_HEADERONLYPOST_ONCONNECTIONRESET
HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_CHUNK_TIMEOUT_KB914453
HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_CERT_TRUST_VERIFIED_KB936882
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache
HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache
HKEY_CURRENT_USER\SOFTWARE\Policies\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache
HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_BUFFERBREAKING_818408
HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_SKIP_POST_RETRY_ON_INTERNETWRITEFILE_KB896964
HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_ENSURE_FQDN_FOR_NEGOTIATE_KB899417
HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_HTTP_DISABLE_NTLM_PREAUTH_IF_ABORTED_KB902409
HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_PERMIT_CACHE_FOR_AUTHENTICATED_FTP_KB910274
HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_WPAD_STORE_URL_AS_FQDN_KB903926
HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_USE_CNAME_FOR_SPN_KB911149
HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_KEEP_CACHE_INDEX_OPEN_KB899342
HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_WAIT_TIME_THREAD_TERMINATE_KB886801
HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_FIX_CHUNKED_PROXY_SCRIPT_DOWNLOAD_KB843289
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Shell Extensions

```

Figure 4.11 The list of the Registry affected by the IRC bot

```
C:\Documents and Settings\User\Local Settings\Temporary Internet Files
C:\Documents and Settings\User\Local Settings\History
C:\Documents and Settings\User\Local Settings\Temporary Internet Files\Content.IE5\
C:\
C:\Documents and Settings\User\Local Settings\Temporary Internet Files\Content.IE5\index.dat
C:\Documents and Settings\User\Cookies\
C:\Documents and Settings\User\Cookies\index.dat
C:\Documents and Settings\User\Local Settings\History\History.IE5\
C:\Documents and Settings\User\Local Settings\History\History.IE5\index.dat
C:\WINDOWSExplorer.exe
C:\WINDOWS\system32\csrss.exe
```

Figure 4.12 The list of files affected by the IRC bot

Figure 4.12 shows the list of the files that effected by the IRC bot that perform unauthorised change to the infected host. The total files in the infected host that have been change is 11 files from the binary of the malware MD5 650c67e14cfb27879999036741478d5. Figure 4.8 shows those dependencies for this binary, which are a650c67e14.exe, services.exe, csrss.exe, Explorer.EXE and dwwin.exe. The Explorer.EXE and csrss.exe has been performed an action in the list of the files. The file csrss.exe has been placed into the system file which able the bot to be able to achieve the change to the infected host.

4.2.4 Wireshark

Wireshark has installed in the machine to be able to capture any data travelled inbound or outbound of the system. The First aim for the botnets is to scan the system to find any vulnerabilities in the system that would able the bot to inject the system. The Figure 4.6 shows that the IP address 127.0.0.0 and 127.0.0.2 requested from the host database as well as the botnets server tried to establish a connection by scanning the IP address 127.0.0.2 to 127.0.0.11. The botnets scans all the possible IP address to establish a connection with the server by injecting the host. The port that attempted to use was 445, which is the port of Dionaea honeypot that attempts the malware to use this port, as it is a vulnerability for the system to be able to download the binaries of the malware. After the scanning process, the botnet attempts to authenticate with the host to complete the injection

process. During the experiment, the Wireshark has been able to collect some of the database that has been connecting to the machine.

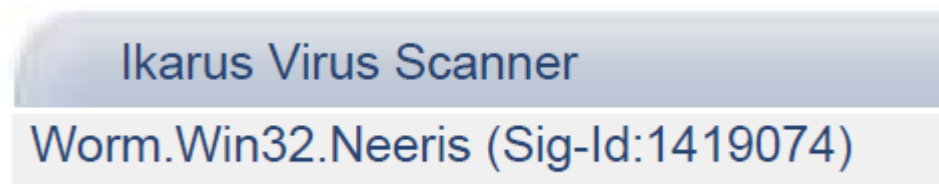


Figure 4.13 The scan result of the IRC bot from Anubis

75	Standard query 0xed0a	A	gg.arrancar.org
135	Standard query response 0xed0a	A	128.111.73.201

Figure 4.14 The DNS indicates of a suspicious IRC bot server from Wireshark

The wire shark has been able to capture some of the data that are going inbound and outbound of the infected host. There was the domain name server (DNS), which has requested from the infected machine as shown in figure 4.14. Therefore, after searching in google for the domain name, as it shown dangerous to access the website for security reasons; hence the result came out with results of this website being malicious. One of the results was from Windows Corporation that shows that the website is a worm.

Worm:Win32/Neeris.AN



Worm:Win32/Neeris.AN is a worm that spreads by removable drives and by attempting to exploit a number of particular vulnerabilities. The worm also contains backdoor functionality that allows unauthorized access and control of the affected computer.

Figure 4.15 The search result of the IRC bot from Windows

The Windows search in figure 4.15 shows that the search for the DNS gg.arrancar.org is a malicious website that backdoor which allows the IRC bot to be able to perform unauthorised access and control of the infected host. Figure 4.15 shows that the attempt to find a vulnerabilities in the host to be able to exploit the malicious binaries in the infected host. This supports the ThreatExpert in figure 4.6, which shows the IRC bot attempted to establish a connection with

the Dionaea honeypot port 445. Also Supports the Wireshark monitoring of the traffic shows that the IRC bot attempted to scan the host to be able to exploit the binaries in the host.

43	111.070539	128.111.73.201	192.168.0.2	TCP	60 dsf > syscomlan [RST, ACK] Seq=1 Ack=1 win=0 Len=0
44	112.712434	192.168.0.2	128.111.73.201	TCP	62 [TCP Retransmission] syscomlan > dsf [SYN] Seq=0 win=16384 Len=0 MSS=1460 SACK_PERM=1
45	112.713137	128.111.73.201	192.168.0.2	TCP	60 dsf > syscomlan [RST, ACK] Seq=1 Ack=1 win=0 Len=0
46	113.887431	192.168.0.2	128.111.73.201	TCP	62 [TCP Retransmission] syscomlan > dsf [SYN] Seq=0 win=16384 Len=0 MSS=1460 SACK_PERM=1
47	113.891775	128.111.73.201	192.168.0.2	TCP	60 dsf > syscomlan [RST, ACK] Seq=1 Ack=1 win=0 Len=0
48	115.061765	192.168.0.1	192.168.0.2	TCP	60 microsoft-ds > startron [FIN, ACK] Seq=1101 Ack=1613 win=9843 Len=0

Figure 4.16 The TCP Traffic of the suspicious IRC bot from Wireshark

The IRC bot attempted to make a connection to the IP address of the DNS 128.111.73.201. The result of the TCP traffic shows that the connection going between the malicious DNS and the infected machine. The researcher believes that the DNS for this malware is not the primary DNS for the bot. After further searching to get access to the DNS to analysis it further, the domain did not exist anymore and it appears that the DNS www.arrancar.xxx is not owned by anyone and it is for sale.

31	18003.6325	172.16.87.128	172.16.87.2	DNS	78 Standard query 0x0001 A xiaoruiip.3322.org
32	18004.0507	172.16.87.2	172.16.87.128	DNS	138 Standard query response 0x0001 A 113.5.121.60

Figure 4.17 The DNS indicates of a suspicious IRC bot server from Wireshark

Protocol	Length	Info
ICMP	62	Echo (ping) request id=0x166e, seq=0/0, ttl=16
ICMP	62	Echo (ping) request id=0x166e, seq=0/0, ttl=128 (reply in 27)
ICMP	62	Echo (ping) reply id=0x166e, seq=0/0, ttl=128 (request in 24)
ICMP	47	Echo (ping) request id=0x0200, seq=256/1, ttl=1 (reply in 40)
ICMP	47	Echo (ping) reply id=0x0200, seq=256/1, ttl=128 (request in 39)
ICMP	62	Echo (ping) request id=0xc66d, seq=0/0, ttl=16
ICMP	62	Echo (ping) request id=0xc66d, seq=0/0, ttl=128
ICMP	62	Echo (ping) request id=0x166e, seq=0/0, ttl=16
ICMP	62	Echo (ping) request id=0x166e, seq=0/0, ttl=128 (reply in 18643)
ICMP	62	Echo (ping) reply id=0x166e, seq=0/0, ttl=128 (request in 18640)

Figure 4.18 The checking statues of the bot from Wireshark

Figure 4.18 show that the suspicious C&C is checking the statues of the bot in the infected host. The checking statues performed by using Internet control message protocol (ICMP). The traffic that captured in figure 4.18 is encrypted the contained information could not be read.

The Wireshark has also identified another possible malicious DNS that has connected to the machine. The malicious website has been scanned by the

virusTotal that points out that the website is malicious which is identified by 3 of the security engines out of the 37.



Passive DNS replication

VirusTotal's passive DNS only stores address records. The following domains resolved to the given IP address.

2013-04-26 xiaoruiip.3322.org

Latest detected URLs

Latest URLs hosted in this IP address detected by at least one URL scanner or malicious URL dataset.

3/37 2013-04-26 15:17:42 http://xiaoruiip.3322.org/

Figure 4.19 The DNS result of the IRC bot from virusTotal

The Figure 4.19 shows that the site has identified as a malicious site. However, this result of the scan does not guarantee that the site is actually a malicious, the reason for that is that there are 34 engines do not indicate that the site is malicious. There are various reasons that explain why the site is not identified by the other 34 engines including that the site has not been scanned yet by the engine. In addition, there has been no report of the site submitted to the engine as well as the site might be taken down and no longer available for the bot is programmed to check this site first. Therefore, the three of the engines that indicate the site is malicious have a reason of the detection of the malicious activity of this site. In addition, Figure 4.19 shows the IP address of the suspicious site by searching for the IP address the domain name server <http://xiaoruiip.3322.xxx> came with the same result that shown in figure 4.19. This research has been able to identify another site that has the same result as the result of the figure 4.17.



URL:	http://xylox.su/php/login.php
Detection ratio:	3 / 51
Analysis date:	2013-11-27 17:00:33 UTC (6 months ago)

Analysis	Additional information	Comments 0	Votes
----------	------------------------	------------	-------

URL Scanner	Result
BitDefender	Malware site
Emsisoft	Malware site
Sophos	Malicious site
ADMINUSLabs	Clean site
AegisLab WebGuard	Clean site
AlienVault	Clean site
Antiy-AVL	Clean site
AutoShun	Unrated site

Figure 4.20 The DNS result of the IRC bot from virusTotal

Figure 4.20 shows that the scan of the site has resulted in that the site is a malware site, malicious site, clean site and unrated site. In Regards to the communication between the servers of the botnets, which called Command, and Control server (C&C) it is believed that the command that send to the infected host by the botnet master is sent usually by a plain-text message. The only issue that faces this step of analysing is that the aim of this research is to analysis the traffic and the bot without any harm to others. Connecting to the C&C server may result of an attack to another organisation or host, which this research goal is to avoid. Therefore, the internet server connection to the infected host will be connecting to the isolated Internet for a short period. The downside of this is that the research will not be able to get further information to analysis to be able to understand the botnets deeper. As mentioned earlier the research has been able to collect some suspicious domains that are supported by Windows or virusTotal.

Offset(V)	Name	PID	PPID	Thds	Hnds	Sess	Wow64	Start
0x841337f8	System	4	0	86	377	-----	0	2014-05-11 05:27:57 UTC+0000
0x84ab2628	smss.exe	244	4	2	29	-----	0	2014-05-11 05:27:57 UTC+0000
0x85215530	csrss.exe	328	320	9	475	0	0	2014-05-11 05:28:01 UTC+0000
0x8523a530	csrss.exe	380	372	10	1038	1	0	2014-05-11 05:28:01 UTC+0000
0x85244530	wininit.exe	388	320	3	75	0	0	2014-05-11 05:28:01 UTC+0000
0x85529d40	winlogon.exe	424	372	6	113	1	0	2014-05-11 05:28:01 UTC+0000
0x855412d8	services.exe	484	388	11	210	0	0	2014-05-11 05:28:02 UTC+0000
0x85549770	lsass.exe	492	388	7	509	0	0	2014-05-11 05:28:03 UTC+0000
0x846d1030	lsmd.exe	504	388	10	146	0	0	2014-05-11 05:28:03 UTC+0000
0x85565858	svchost.exe	592	484	13	363	0	0	2014-05-11 05:28:04 UTC+0000
0x85580030	svchost.exe	668	484	7	271	0	0	2014-05-11 05:28:05 UTC+0000
0x85598310	svchost.exe	752	484	19	477	0	0	2014-05-11 05:28:05 UTC+0000
0x855a7980	svchost.exe	796	484	15	262	0	0	2014-05-11 05:28:05 UTC+0000
0x855b3b78	svchost.exe	820	484	45	997	0	0	2014-05-11 05:28:05 UTC+0000
0x855c3190	audiodg.exe	884	752	7	131	0	0	2014-05-11 05:28:06 UTC+0000
0x855da530	svchost.exe	952	484	13	527	0	0	2014-05-11 05:28:06 UTC+0000
0x855ea808	svchost.exe	1016	484	18	492	0	0	2014-05-11 05:28:06 UTC+0000
0x85288030	spoolsv.exe	1136	484	15	315	0	0	2014-05-11 05:28:07 UTC+0000
0x8560ad40	svchost.exe	1164	484	20	316	0	0	2014-05-11 05:28:08 UTC+0000
0x8526ed40	vmtoolsd.exe	1320	484	10	274	0	0	2014-05-11 05:28:09 UTC+0000
0x84a81030	taskhost.exe	1496	484	9	150	1	0	2014-05-11 05:28:10 UTC+0000
0x84a82030	dwm.exe	1504	796	6	73	1	0	2014-05-11 05:28:10 UTC+0000
0x8567e718	explorer.exe	1544	1480	40	1003	1	0	2014-05-11 05:28:10 UTC+0000
0x8570dcb0	dllhost.exe	1888	484	18	187	0	0	2014-05-11 05:28:13 UTC+0000
0x8569d030	TPAutoConnSvc.	1964	484	10	139	0	0	2014-05-11 05:28:13 UTC+0000
0x8572da58	dllhost.exe	256	484	17	193	0	0	2014-05-11 05:28:14 UTC+0000
0x85758c00	msdtc.exe	1260	484	15	152	0	0	2014-05-11 05:28:15 UTC+0000
0x85760140	vmtoolsd.exe	1292	1544	8	226	1	0	2014-05-11 05:28:15 UTC+0000
0x8575d448	TPAutoConnect.	1280	1964	6	130	1	0	2014-05-11 05:28:15 UTC+0000
0x8575ed40	conhost.exe	1084	380	1	33	1	0	2014-05-11 05:28:16 UTC+0000
0x85787990	VSSVC.exe	1752	484	5	108	0	0	2014-05-11 05:28:18 UTC+0000
0x857ebb68	SearchIndexer.	1632	484	14	537	0	0	2014-05-11 05:28:21 UTC+0000
0x8439b898	WmiPrivSE.exe	2092	592	9	129	0	0	2014-05-11 05:28:22 UTC+0000
0x85828268	SearchProtocol	2232	1632	7	306	0	0	2014-05-11 05:28:26 UTC+0000
0x8582ecf0	SearchFilterHo	2252	1632	5	78	0	0	2014-05-11 05:28:26 UTC+0000
0x8435e7d0	4d4c2729b8aa56	2888	1544	265	1406	1	0	2014-05-11 05:29:54 UTC+0000
0x84526260	4d4c2729b8aa56	4016	1544	263	1346	1	0	2014-05-11 05:30:12 UTC+0000
0x84562450	svchost.exe	2056	484	6	68	0	0	2014-05-11 05:30:13 UTC+0000
0x8456e408	sppsvc.exe	2384	484	6	153	0	0	2014-05-11 05:30:13 UTC+0000
0x84580578	svchost.exe	664	484	17	340	0	0	2014-05-11 05:30:14 UTC+0000
0x8460f710	4d4c2729b8aa56	2792	1544	5	61	1	0	2014-05-11 05:30:17 UTC+0000
0x8463fd40	f024ff4176f003	4636	1544	262	1348	1	0	2014-05-11 05:30:31 UTC+0000
0x857d3d40	cmd.exe	5204	1544	5	104	1	0	2014-05-11 05:30:47 UTC+0000
0x857cb840	conhost.exe	5216	380	2	49	1	0	2014-05-11 05:30:47 UTC+0000
0x84992680	DumpIt.exe	6124	5204	2	37	1	0	2014-05-11 05:30:57 UTC+0000
0x84a73d40	conhost.exe	6132	380	2	49	1	0	2014-05-11 05:30:57 UTC+0000

Figure 4.21 the running process in the physical memory of the infected host

Figure 4.21 shows the running process in the physical memory of the infected host after the infection of the IRC bot. These processes have been targeting by the IRC bot to be able to make use of the infected host. The bot will notify the C&C server about the running process in the infected host. Out of all the malware that have

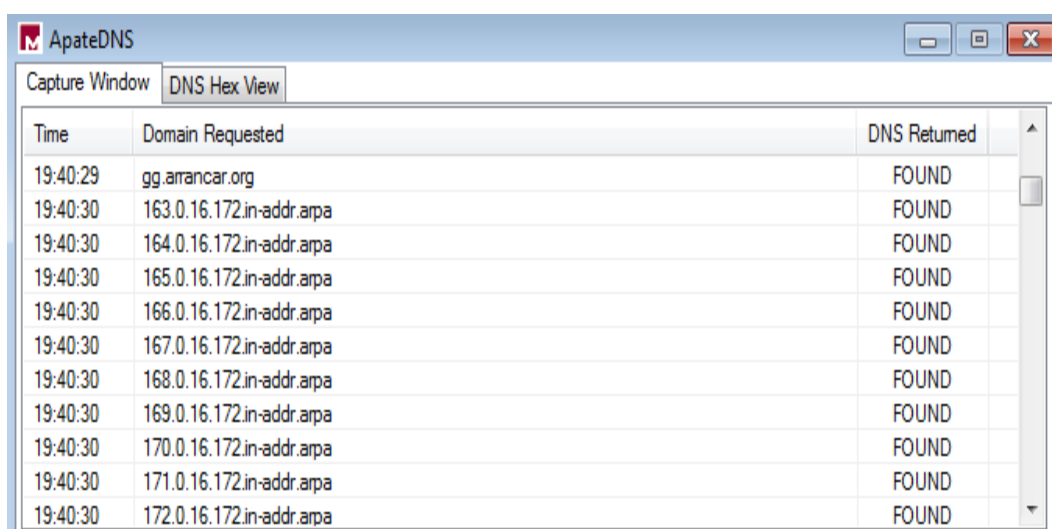
collected by Dionaea Honeypot, the main targeting processes in most of the malware are the system, svchost.exe and explorer.exe. The physical memory of the infected host has preserved as evidence by the Dumpit software that makes a copy of the physical memory of the infected host. Then the infected host's physical memory have been examined using the Volatility software that scans the running process in the infected host, perform a network scan as well as performing the open files scan in the infected host. After examining the physical memory of the infected host, the results have saved into a text file to preserve the results of the physical memory. The examination of the copy of the physical memory by Dumpit has examined again and the results were exactly similar to the first copy of the physical memory of the infected host.

4.2.3 Live Monitoring

This research has been able to download a number of malware binaries through Dionaea honeypot especially those malware that is using by botnets. The binaries of the botnets that have downloaded have used again for further analysis to be able to get an accurate result. The each binary has injected to the Virtual Machine multiple times to be able to study the changes that have made to the infected host. There are number of tools that have been useful for this live experiment. The binary of the bot will be injected to the host and then monitoring the changes that have been done to the infected host. This experiment was performing for a short period. The tool (Regshot) will first take a shot of the files of the system. The reason for that is that the Regshot is able to take two shots of the files of the system and compare the changes that have occurred to the system between the first shot and the second shot then compare them and display the changes that occurred in the system. The Windows installed as "out of the box" which means that the windows files system is in its original setting. Then the Regshot took the shot of the files of Windows system. After injected the host an IRC bot binary signature of a650c67e14cfb27879999036741478d5 there were number of changes that have occurred in the system. The keys added to the system of the Registry were 143 as well as there were 792 values added to the system. In addition to 72 values have modified which leads to 1007 total changes to the system. When the experiment has repeated, again with another IRC bot, which has the signature of

0a278f8d72e4d3d2d44485764398c84d the total changes have occurred to the system, were 112475 changes. It is clear that the second IRC bot was more active than the first IRC bot, which implies that the Second IRC bot is still active.

After that, the ApateDNS opened to be able to collect all the requested domain names from the machine. This tool will provide a live capture of all the domain names that has requested by the infected host. This will provide either the domain names that have requested or the IP address of the domain names. Google.com has accessed by the browser before the injection to insure an accurate result and only a google domain name was displaying in the program. Then after the injection of the system the domain name that has been requested in figure 4.14 has been requested again after the injection of the IRC bot a650c67e14cfb27879999036741478d5 signature then there were random selections of domain names with IP addresses which is the bot pre-programmed to connect to and exploit the binary which is shown in figure 4.22.



Time	Domain Requested	DNS Returned
19:40:29	gg.arrancar.org	FOUND
19:40:30	163.0.16.172.in-addr.arpa	FOUND
19:40:30	164.0.16.172.in-addr.arpa	FOUND
19:40:30	165.0.16.172.in-addr.arpa	FOUND
19:40:30	166.0.16.172.in-addr.arpa	FOUND
19:40:30	167.0.16.172.in-addr.arpa	FOUND
19:40:30	168.0.16.172.in-addr.arpa	FOUND
19:40:30	169.0.16.172.in-addr.arpa	FOUND
19:40:30	170.0.16.172.in-addr.arpa	FOUND
19:40:30	171.0.16.172.in-addr.arpa	FOUND
19:40:30	172.0.16.172.in-addr.arpa	FOUND

**Figure 4.22 The DNS requested by an IRC bot
a650c67e14cfb27879999036741478d5**

The figure 4.22 shows the domain name that has requested straight after the injection of the system. Then more than 200 IP addresses have requested within a minute of the injection of the system. The IRC bot with a signature of 0a278f8d72e4d3d2d44485764398c84d was acting a bit different as shown in figure 4.22

PID	Operation	Path	Result
3272	Process Start		SUCCESS
3272	Thread Create		SUCCESS
3272	Load Image	C:\Users\sultan\Desktop\exp\bin\650c67e14cfb2787999036741478d5.exe	SUCCESS
3272	Load Image	C:\Windows\System32\ntdll.dll	SUCCESS
3272	Load Image	C:\Windows\System32\kernel32.dll	SUCCESS
3272	Load Image	C:\Windows\System32\KernelBase.dll	SUCCESS
3272	Load Image	C:\Windows\System32\advapi32.dll	SUCCESS
3272	Load Image	C:\Windows\System32\msvcrt.dll	SUCCESS
3272	Load Image	C:\Windows\System32\sechost.dll	SUCCESS
3272	Load Image	C:\Windows\System32\rpcrt4.dll	SUCCESS
3272	Load Image	C:\Windows\System32\shell32.dll	SUCCESS
3272	Load Image	C:\Windows\System32\shlwapi.dll	SUCCESS
3272	Load Image	C:\Windows\System32\gdi32.dll	SUCCESS
3272	Load Image	C:\Windows\System32\user32.dll	SUCCESS
3272	Load Image	C:\Windows\System32\lpk.dll	SUCCESS
3272	Load Image	C:\Windows\System32\usp10.dll	SUCCESS
3272	Load Image	C:\Windows\System32\wssock32.dll	SUCCESS
3272	Load Image	C:\Windows\System32\ws2_32.dll	SUCCESS
3272	Load Image	C:\Windows\System32\ansi.dll	SUCCESS
3272	Load Image	C:\Windows\System32\imm32.dll	SUCCESS
3272	Load Image	C:\Windows\System32\msctf.dll	SUCCESS
3272	Load Image	C:\Windows\System32\sspicli.dll	SUCCESS
3272	Load Image	C:\Windows\System32\wininet.dll	SUCCESS
3272	Load Image	C:\Windows\System32\urlmon.dll	SUCCESS
3272	Load Image	C:\Windows\System32\ole32.dll	SUCCESS
3272	Load Image	C:\Windows\System32\oleaut32.dll	SUCCESS
3272	Load Image	C:\Windows\System32\crypt32.dll	SUCCESS
3272	Load Image	C:\Windows\System32\msasn1.dll	SUCCESS
3272	Load Image	C:\Windows\System32\iertutil.dll	SUCCESS
3272	Load Image	C:\Windows\winsxs\x86_microsoft.windows.common-controls_6595b64144ccf1df_6.0.7601.17514_none_41e6975e2bd6f2b2\comctl32.dll	SUCCESS
3272	Load Image	C:\Windows\System32\profapi.dll	SUCCESS
3272	Load Image	C:\Windows\System32\dnsapi.dll	SUCCESS
3272	Load Image	C:\Windows\System32\IPHLPAPI.DLL	SUCCESS
3272	Load Image	C:\Windows\System32\winnsi.dll	SUCCESS
3272	Load Image	C:\Windows\System32\icmp.dll	SUCCESS
3272	Load Image	C:\Windows\System32\netapi32.dll	SUCCESS
3272	Load Image	C:\Windows\System32\netutils.dll	SUCCESS
3272	Load Image	C:\Windows\System32\svcli.dll	SUCCESS
3272	Load Image	C:\Windows\System32\wksccli.dll	SUCCESS
3272	Load Image	C:\Windows\System32\schedcli.dll	SUCCESS
3272	Load Image	C:\Windows\System32\samcli.dll	SUCCESS
3272	Load Image	C:\Windows\System32\mpr.dll	SUCCESS
3272	Load Image	C:\Windows\System32\odbc32.dll	SUCCESS
3272	Load Image	C:\Windows\System32\odbcint.dll	SUCCESS
3272	Load Image	C:\Windows\System32\psapi.dll	SUCCESS
3272	Thread Create		SUCCESS
3272	Thread Exit		SUCCESS
3272	Thread Exit		SUCCESS
3272	Process Exit		SUCCESS

Figure 4.25 The list of loaded external libraries

4.3 ANALYSIS

In the previous sections, this research has presented evidence that that has been found in the infected machine. The binary of bots were analysed by using both an external service such as ThreatExpert and Anubis and other tools that are designed

to analysis the malware. This subsection will report 4.4.1 Propagation, 4.4.2 Infection and 4.4.3 connecting to the Botnets Server

4.3.1 Propagation

The botnets typically search for any vulnerability in a system to be able to propagate the botnet and have more victims involved in the army of the bot. The mechanism of the propagation is downloading in the database of the malware collection software that has used in this research, which called “Dionaea”. The infected machine (IP address 118.92.13.71) was injected by the remote host (IP address 220.135.173.144). The injection of the malware using the port 445 and the using the vulnerabilities of the Microsoft Server Message Block (SMB). The SMB vulnerability allowed the attacker to use it to be able to remote injection of the binary using the shellcode for the binary. Table 4.2 shows one of the shellcodes that has downloaded.

```
[
  {
    "call": "_lcreat",
    "args" : [
      ".exe",
      "6"
    ],
    "return": "4711"
  },
  {
    "call": "LoadLibraryA",
    "args" : [
      "ws2_32.dll"
    ],
    "return" : "0x71a10000"
  },
  {
    "call": "socket",
    "args" : [
```

<pre> "2", "1", "6"], "return": "65" }, { "call": "bind", "args": ["65", { "sin_family": "2", "sin_port": "9988", "sin_addr": { "s_addr": "0.0.0.0" }, "sin_zero": " " }, "16"], "return": "0" }, { "call": "listen", "args": ["65", "16"], "return": "0" }] </pre>
--

Table 4.2 Shellcode downloaded by Dionaea honeypot

4.3.2 Infection

This is the step when the binary of the botnet exploit the system of the victim's machine to be able to join the army of the botnets. The infection of the system of the victim's machine enables the bot to copy all the files that are in the malicious bot to the machine. Typically, the bot copies all the files into the C:\Windows\System*. These files enable the bot to be booted whenever the victim's machine is booted. The first aim for the bot after the exploitation of the binary is to disabled the firewall, the Security Centre Service as well as an Anti-virus software.

4.3.3 Connecting To The Botnets Server

A bot is believed to be pre-programmed to be able to perform some of the activity in the host side before joining the botnets server that known as command and control server (C&C). The activities that performed in the host side prior to joining the C&C server to guarantee that the communication between the bot and C&C stays undetected by disabling the firewall as well as the security service that has mentioned in the infection process. Then the bot will control the host and communicate with the server of the botnets to join the server as a new member and wait for instructions from the botnet master. The live monitoring of the infected-machine that has monitored in this research shows that the machine connected to more than 200 IP addresses within a minute of the infection by the IRC bot. In addition, another IRC bot connected to more than 400 IP addresses within a minute. This means that the bot pre-programmed to communicate to these IP addresses to join the server and wait for instruction.

4.3.4 Summary Of The Analysis

Table 4.3 is the Summary of the findings that have been presented in chapter 4. The results of the experiments have been presented in the figures and tables in this chapter 4. However, the table 4.3 shows the evidence found as well as how the researcher found them. The experiments of this research were performed by using external sandboxes services, malware analysis tools and live monitoring of the infected host. The malware analysis is when the malware was analyzed using the malware analysis tool and the live monitoring is when the host that infected by the bot is monitored using different tools such as sniffer tools, registry tools, SysinternalsSuite and other tools to monitored the infected host.

Table 4.2: Summarized of the analysis of the botnets

Botnets Analysis		Reconstructed Data
	Evidence	How
Bot in the malware collection	found	Using the ThreatExpert to verify the type of the malwares that have been collected using Dionaea honeypot
Registry changes	Found	Using sandboxes (ThreatExpert, Anubis), Regshot and SysinternalsSuite (Process Monitor)
Files Systems changes	Found	Using sandboxes (ThreatExpert, Anubis) and SysinternalsSuite (Process Monitor)
Network Activities	Found	Using sandboxes (ThreatExpert, Anubis), ApateDNS and Wireshark
Evidence in Physical Memory	Found	Using Dump it and Volatility Framework
C&C server	Found	Using Anubis sandbox, Wireshark and ApateDNS
C&C command	Found	Using Wireshark to sniff the traffic going inbound or outbound
Communication to the C&C server	Found	Using Wireshark, ApateDNS and SysinternalsSuite (Process Monitor)

Shellcode	Found	Dionaea database
-----------	-------	------------------

Table 4.3 Summary of the analysis

4.4 CONCLUSION

Chapter 4 has reported the findings of the malware collection and analysis by using an external analysis services as well as tools that developed for analysis of the malware. Binaries that has used in this research has been downloaded by Dionaea honeypot over a period of 22 days. The machine of the researcher has connected to the isolated internet server using Virtualization Technology and located DMZ inside the researcher network. The DMZ used to prevent the malware spreading out throughout the researcher network and affecting other machines that are connected to the network of the researcher. The researcher was able to download bots using the Dionaea honeypot and other malware that has classified as unknown. Using the external service sandboxes such as ThreatExpert and Anubis, which are a service that is, designed to analysis the malware samples that have been submitted online to their service and create a full report of the submitted malware sample.

In addition, other tools used in the machine of the researcher to be able to perform a further analysis of the bot sample. All the bot samples that have downloaded through Dionaea honeypot were used to inject the machine of the researcher to study the behaviour of the bot inside the machine and the network of the researcher machine. The bots monitored after the injection using several tools such as ApateDNS, Regshot, Wireshark, NC and SysinternalsSuite to be able to monitor the behaviour of the bot after the injection of the machine.

The result of chapter 4 as well as other results that have been gathered by the Dionaea honeypot, which has offered a large amount of information regardsing the downloaded malware binaries, shellcode, IP addresses of attackers as well as other useful information that has been stored in a database. In addition, the collected malwares have been analysis by an analysis malware software as well as an external service that offered useful information about the malware. Furthermore, the downloaded malware binaries have injected to a machine to be

able to monitor the behaviour of the malware inside the infected machine. All the information and the analysis of the malware are about the malware activities in the registry files, files system as well as the network behaviour of the malware. The analysis information use in chapter 5 to answer the main research question as well as the sub-questions.

Chapter 5

Research Discussion

5.0 INTRODUCTION

Chapter 4 reported on the findings of the research according to the research design that presented in chapter 3. The findings of the experiment show further knowledge about the malicious malware especially the botnets, which is the main aim for this research. The findings of the experiment that have been presented in chapter 4 which has been performed in a forensic investigation manner and it will able the researcher to answer the research questions and the hypothesis that was presented in section 3.2.3. In addition, the chapter 4 findings will assist to answer the main question as well as the sub-question that relate to the hypothesis.

Section 5.1 is to gather the findings from Chapter 4 and answer the research question in which the hypotheses will be tested. The main question of this research and the sub-questions specified in section 3.2.3, the sub-questions will be answered and will be discussed first in order to be able to determine the arguments made for or against for each derived hypothesis in section 3.2.3. Associated hypotheses in section 5.1.2 and the main research hypothesis in section 5.1.3 each of them will presented in a table form. The justification of the hypothesis will made as accepted, rejected or indeterminate, which based on the arguments made in regards to the research findings. The discussion of the research findings in chapter 4 from the experiment will be presented in section 5.2 and the expectations set from the literature review in chapter 2. Finally, the conclusion of chapter 5 will be in section 5.3.

5.1 ANSWERING THE RESEARCH QUESTION

This section tests the research hypotheses established in section 3.2.3 with the findings that were collected from the section 4.3 and 4.4 in order to have an appraisal of the arguments in relation to the research hypotheses. The presentation of this section is as follows; section 5.1.1 is to answer the sub-question of the research from collected evidence that has collected from the experimental testing

in this research. Section 5.1.2 tests the main hypotheses and the associated hypotheses of this research with arguments for and arguments against in a table form. The arguments for is the argument that supports the hypotheses, nevertheless, the argument against is the argument is that which refutes the asserted hypothesis. Ultimately, the main question of this research will be answered in section 5.1.3.

5.1.1 Sub-Question Answers

The ability of answering the hypothesis will rely on the answer of the sub-questions that presented in section 3.2.3. Therefore, the sub-questions of this research will need to be answered first which will be presented in table form.

Table 5.1: Sub-Question 1 and Answer

Sub-Question 1 (SQ1):
How many bots binaries were downloaded during the malware collection?
Answer:
13 IRC botnet binaries signatures. 59 malware binaries.
Summary:
The software that has used in this research for the collection of the malware was Dionaea honeypot. Dionaea honeypot is a tool that has developed in order to collect malware from the internet; Dionaea installed. Dionaea was running for 22 days and was able to download 59 binaries of malware signatures. The malware binaries downloaded into a separate file. The binaries of the downloaded malware were then analysed by an external service sandboxes to be able to distinguish the type of the malwares that have downloaded. The scan of the external sandboxes that have used in this research provides the information from different anti-virus vendors to have an accurate result of the analysis. However, the binaries that had been downloading using Dionaea were different types of malwares such as backdoor malware, Trojan, bots and others. The Dionaea's development is to download different types of malware including the botnets binaries. Therefore, the malware that has been downloaded using Dionaea are not all botnet binaries but all types of malwares have been downloaded including an unknown one. The total of the botnet binaries that have downloaded using

Dionaea are 13 with all of botnet binaries being IRC botnet binaries.

Table 5.2: Sub-Question 2 and Answer

Sub-Question 2 (SQ2): Does the physical memory of the infected host contain any information in regards to the botnets event?
Answer: Yes
Summary: The information that is extracted from the physical memory of the infected host is valuable information that could be gathered from the infected host. The information of the physical memory can be extracted only when the machine has not turned off. The reason for that is that once the machine is turned off the information that is in the physical memory will be deleted as the information is temporary saved until the machine is turned off. The information that has extracted from the physical memory is the running processes in the infected machine as the time of the extraction. In addition to the researcher performed a network scan that provides the information about the network activity at the time of the extraction of the physical memory. Furthermore, the opening files can be extracted from the physical memory of the infected machine. The information that has extracted from the physical memory in the infected host does provide information about the active process, network and files. However, the researcher is not able to investigate the information in further detail as the physical memory of the infected machine does not provide more detail. In additional to the information apart from the extracted information that has been mentioned earlier.

Table 5.3: Sub-Question 3 and Answer

Sub-Question 3 (SQ3): Can the information of the physical memory be gathered and preserved?

Answer:

Yes

Summary:

The information that has extracted from the physical memory of the infected host is by using the Dumpit software. Dumpit is a tool that dumps the information of the physical memory into a file with that file extension as .raw. Then the information of the physical memory of the infected host can be investigated using the Volatility tool that provides a scan to the dumpit ,raw file. The information that will be provided to the investigation will be the running process, network and open files. The Dumpit tool provides the ability to preserve the information of the physical memory, which will be available to the investigator to be able to repeat the investigation of the physical memory again. Ultimately, the information of the physical memory will be preserved as evidence for additional action.

Table 5.4: Sub-Question 4 and Answer

Sub-Question 4 (SQ4):

How can the behaviour of the bot can be detect in the infected-host?

Answer:

The Registry Activity, Network Activity and File Activity can detect the behaviour of the bot.

Summary:

The detection of the bot inside the infected host can be challenging, as there is no guaranteed detection techniques that have found yet. Even the tools that are provided by the biggest operating system corporation such as Microsoft that are developed to remove the malicious malware are not efficient enough. Each bot has a different activity inside the infected host as shown in the chapter four. Even the IRC bots can behave differently from each other. Therefore, monitoring the traffic going inbound and outbound of the host is still one of the efficient ways to detect the existence of the bot inside the host. In addition, the bot usually scans the other network machines to be able to find vulnerabilities in another machine to get infected by the same bot. The Figure 4.7 shows how the IRC bot scans the IP addresses from 127.0.0.2 to 127.0.0.11 to find

vulnerabilities and establish a connection with one of these addresses. Prior to the infections, which will be the traffic that the security should look at. The bot usually tries to scan the whole network IP addresses to be able to infect them. Therefore, monitoring the live traffic is the sufficient way to detect the behaviour of the bot inside the network. However, the Registry activity can be monitored as well because the bot creates, modifies and deletes registry files at the time of the infection. In addition to the file activities it is another way of monitored the activity of the bot. The bot usually creates, modifies and deletes files in the infected host.

Table 5.5: Sub-Question 5 and Answer

Sub-Question 5 (SQ5):
What is the behaviour of the bot inside the network of an infected-host?
Answer:
Disable the Firewall, Security Centre and anti-virus software
Summary:
The behaviour of all bot in the infected host is typically the same by all bots. The first aim by the bot to perform in the infected host is to disabled the Windows Firewall, Security Windows Centre in the infected host as well as the anti-virus software. Changing the value of the Registry key of the related keys as well as file systems to be able to avoid detection. Changing the value of these keys would enable the bot to have a full control of the host as well as being undetected because the Windows Firewall, Security Windows Centre and Anti-virus software have had changed their values. Then, the bot will run itself as a Windows process in the host and connect to the server of the bot to update its statues and join the botnets communication channel.

Table 5.6: Sub-Question 6 and Answer

Sub-Question 6 (SQ6):
What is the suspicious activity of the command and control that can be found in the network traffic?
Answer:
Kernal32, GetUserName, UserName, Password and other commands

Summary:

There were suspicious command and control channel instructions that have found during the experiment of this research. There was little traffic that has been connected to the command and control channel. The Domain Name System (DNS) sends a query to the machine then the connection established between the infected host and the server. The command instructions seem to be sent in a plain-text format and can be seen clearly by monitoring the traffic. By following the TCP connection traffic we can see that the C&C server has been sent instructions to the infected host. NewUserName, NewPassword, InternalPort, RemoteHost, SetConnectionType, NewConnectionType, GetNewConnectionTypeInfo, RequestConnection, GetStatusInfo, NewConnectionStatuses, NewInternetClient, NewUpstreamMaxBitRate, NewDownstreamMaxBitRate and GetExternalIPAddress. These suspicious instructions sent to and from the infected host seem to set the infected host to be part of the C&C server. Therefore, the C&C server have the information about the infected host then establish a connection that is undetected with the C&C server then the infected machine will be linked as the instructions `http://192.168.1.1:80`, `http://192.168.1.1:2555` was set to link the C&C with the infected host. The suspicious C&C server seems to set new settings to the infected host to adopt it with its activities. It can be seen that the port 80 and 2555 was used in the linking process.

Table 5.7: Sub-Question 7 and Answer**Sub-Question 7 (SQ7):**

Is the command and control instructions set encrypted?

Answer:

Not for the IRC bot. However, the HTTP bot uses encryption methods.

Summary:

SQ6 have found instructions that found in the traffic of the infected host. The instructions that sent by the C&C server to the bot in the infected host seems to be sent in a plain-text format. The information followed in the infected host by the traffic sniffer tools and was able to detect suspicious instructions from different malicious domains. However, checking the statues of the bot in the

infected host were unable to achieve. The checking statues of the bot by the C&C seems to be encrypted and the sniffers tools were unable to read these instructions. The infected host were disconnected and then connected after a few days later and the checking statues also unable to achieve recognition. There was a ping request and a ping reply sent in Internet Control Message Protocol (ICMP). The IP address is different than the IP address of the malicious DNS that sent the query. The reason for that is that the botnets seems to have different servers in different locations which the report by Anubis shows that one of the IRC bot have more than 200 DNSs requested. Clearly the command is sent by one of these servers but checking the statues of the command is the challenging part as this research connects to the server for a really short time with each connection being less than a minute to prevent the bot to performing an attack to others. Therefore, The researcher believes that checking the statues of the bot, is being encrypted by the botnet master.

Table 5.8: Sub-Question 8 and Answer

Sub-Question 8 (SQ8):
Is the command and control attack instructions set encrypted?
Answer:
Not for the IRC bot. However, the HTTP bot uses encryption methods.
Summary:
The attack instructions send to the infected host in plain-text for the IRC bot. The IRC disadvantage is that the instructions usually send in a plain text to the infected host. However, the instructions for the HTTP botnet is encrypted.

Table 5.9: Sub-Question 9 and Answer

Sub-Question 9 (SQ9):
Has the research been able to capture any sensitive information sent to the C&C server?
Answer:
Yes.
Summary:
The research was able to capture information about the infected host sent to the

suspicious C&C server. The information that has captured using the sniffer tool (wireshark) such as the username and password. This information captured using the Wireshark that has sent from the infected host to the C&C server in plain text.

5.1.2 Hypothesis Tests

There are three associated hypotheses to be tested in order is verifying the validity of the research findings and to answer the research main question. These hypotheses tests with arguments made for and against to prove or refute the tested hypothesis with the supporting evidence that obtained from the experimental results. The tested hypothesis presented from Table 5.10 to Table 5.12.

Table 5.10: Tested Hypothesis 1

Hypothesis 1 (H1): The researcher's network has vulnerabilities that allowed the botnets to be downloaded.	
TESTED RESULT: Accepted	
ARGUMENT FOR: The Dionaea honeypot was installed which developed for the purpose of malware collection. The traffic coming from the internet directed to the honeypot without any filtering rules. The technique that used for directing traffic to the Dionaea honeypot is Demilitarized Zone (DMZ). The Dionaea emulate the known Microsoft vulnerabilities to be able to exploit it and download each the binary captured into a file named with its MD5 value. Therefore, the malware exists in the researcher's network and the malwares have been downloading through the	ARGUMENT AGAINST: The DMZ technique used to forward all the unhandled traffic to the honeypot. The malwares captured inside the honeypot. The collected malwares were not captured inside the computers that have been connected to the researcher's network as well as all the computers that are connected to the researcher's network is protected by firewall and an anti-virus software. However, malwares captured by the Dionaea honeypot then the malware transferred to an experiment computer, which means that the malware does not exist in the researcher's network but it has

researcher's network using Dionaea.	downloaded by Dionaea honeypot in isolation.
<p>SUMMARY:</p> <p>The collection of this research performed inside a secure server local network. There are few other networks around the local researcher's network, which they might have a malware exist inside them. In addition, the researcher's local isolate network is not as large as originations' network. Therefore, the result of this research might achieve the purpose of this research; however, the result would be a lot better if it had been experimented in a large origination's network. There was a separate access to the internet was set to this research in the origination , nevertheless, the result was not impressed as only 14 unknown binaries were downloaded in a duration of 22 days. Therefore, the researcher performed the collection of the malware from the researcher's private network. The researcher believes that the issue with the honeypot running the origination's network and not downloading the malware binaries is that the firewall of the origination does not allow all the traffic to get through. In addition, the researcher believes that setting more computers in different location would achieve a higher number and be able to achieve more results and further analysis of these malwares.</p>	

Table 5.11 Tested Hypothesis 2

<p>Hypothesis 2 (H2):</p> <p>The host is infected and it contains the information about the C&C server.</p>
<p>TESTED RESULT:</p> <p>Accepted</p>

<p>ARGUMENT FOR:</p> <p>The Chapter 4 has presented the malware evidence that have found in the infected host. The host infected by the malware that have downloaded by Dionaea honeypot and monitored before and after the infection to report the changes in the infected host. The infected host connected to many domain names and IP addresses, which is believed to be the suspicious C&C. The bot malware binary itself does not provide information about the C&C server. However, infecting the host would help to be able to gather information about the C&C. The figure 4.23 shows that the IRC bot requested more than 300 domain names and IP addresses which requires a large amount of work to be able to gather all the hosts requested. In addition, the information that the infected host will provide is that the domain name or the IP address of the C&C server.</p>	<p>ARGUMENT AGAINST:</p> <p>It is easier for the researcher to navigate the activity of the bot as the bot named as the signature of the binary of the bot. Furthermore, the bot is tested and examined in an original windows operating system, which called “out of the box”. There was not any browser opened in the infected host, which means that the research was able to sniff the information transferred inbound and outbound the infected host easily. The information about the C&C was not accessible. In addition, the only information that gathered from the infected host was the domain names and IP addresses.</p>
<p>SUMMARY:</p> <p>The research was able to infect the host in order for the host to communicate with the C&C server. The infected host communicated with the suspicious C&C servers within seconds of the infection of the bot. The domain names and the IP addresses of the suspicious C&C servers requested from the infected host. The ApatDNS was able to show the requested domain names and IP addresses as shown in figure 4.22 and 4.23. The Wireshark was able to capture the information transferred to the suspicious C&C server. The existence of the bot identified and the communication with the suspicious C&C server was able to be seen. The</p>	

infected host shows the domain names and IP addresses requested. However, further detail about the C&C servers was not accessible.

Table 5.12: Tested Hypothesis 3

Hypothesis 3 (H3): The bot in the infected host communicates with the command and control channel	
TESTED RESULT: Yes	
ARGUMENT FOR: During the experiments, the infected host has been communicating to different domains with one of the IRC bot connected to more than 200 domains and IP addresses whereas another IRC connect to more than 400 domains and IP address with both IRC bot being connection to secure internet server. The communication to the domains and IP addresses believes to update the C&C server that a new host has infected and joining the botnets. Then the bot will be ready to receive any instruction from the botnet master. During this short period, the IRC bot communicates with a suspicious C&C server continuously until the internet is disconnected. The traffic of the infected host has been analysed and found some of the instructions found received in the infected host by the C&C server. The	ARGUMENT AGAINST: The domains that have found in the infected host are not malicious as some of the security engines found. In addition, some of the domains that have been found in the infected host were not accessible and in another words do not exist. The communication to a particular domains and IP addresses do not mean that the domains and the IP addresses are malicious. What is more, these domains and IP addresses could be the domain name of the targeted websites that the IRC bot attempts to attack and the main development of this IRC bot is to perform an attack from each individual host that is infected by the IRC bot.

<p>IRC bot believes to pre-program to communicate a list of IP addresses that is including domains once the host has infected. Some of the domains have scanned by a virusTotal and found malicious by some of the security engines. However, without the connection of the internet the infected host tried to connect to the IP addresses continuously even though the internet was not connected.</p>	
<p>SUMMARY:</p> <p>The infected host monitored using a network sniffer tools that would enable the researcher to monitor the traffic that is going inbound or outbound of the infected host. The sniffer tools have captured domains that have been involved in the communication with the suspicious C&C server. In addition, the sniffer tools have captured IP addresses that have been involved in the communication with the suspicious C&C server. The infected host connects to the suspicious C&C server straightway after the infection of the host. The communication lasts for up to a minute with the communications was going inbound and outbound until the infected host disconnected from the secure internet server. The assumption was that the suspicious C&C server as the infected host not connected to any domains and the browser of the internet not launched. The only tools that were running in the infected host were the sniffer tools and other malware analysis tools, the domains and IP addresses requested in the background without the authorization of the owner of the infected host. The IP addresses that have requested in the infected host were from different countries, which means that the C&C servers were actually in different locations. For example, an IP address looked up, which shows the IP address is from China while the other shows the IP address is from USA. The ApateDNS shows the domains and IP addresses that were requested, whereas Wireshark shows same domains and IP addresses being requested which both approves that the suspicious C&C server is exists in the infected host after the infection of the host. The requested domain names and IP addresses were</p>	

running in the background even after the internet server disconnected. This means that the infected host controls by the bot tries to connect to the domain names and IP addresses even though the internet was not connected.

5.1.3 The Research Question Answer

The following Table 5.11 is the research main question and the main hypothesis that is to be tested based on the answer gathered from research sub-questions and the tested associate hypothesis in section 5.1.1 and section 5.1.2 respectively.

Table 5.13: Research Main Question and Tested Hypothesis

Main Question: What is the digital evidence that gathered from the infected-host in a botnets event?	
Main Hypothesis: The infected host contains the information that was changed after the malicious activity of the infected machine	
TESTED RESULT: Accepted	
ARGUMENT FOR: The aim of this research is to find out the evidence that found in the infected host by the Botnet event. The experiment shows the large amount of activities that the bot performed in the infected host. The evidence in the infected host divided into five categories, which are file system activity, registry activity, network activity, loaded libraries and running process in the physical memory. The files system activity shows that there were a large amount of files being changed (created, modified, read and deleted) by the IRC bot. The total	ARGUMENT AGAINST: The evidence that found in the infected host does not provide much information about the activity of the botnets in the infected host. The experiment shows the changes in the infected host that include file activity, registry activity, network activity, process activity and physical memory. The experiment results show the changes of the infected host that have performed by the IRC bot but does not show the illegal activities that the infected host performed in the internet. The experiment shows the performed activities, however, the information that

<p>amounts of files that have changed in the infected host were 3,790. Most of the files that were changes were in the system files with the majority of these changes created then closed. In addition, surprisingly the registry activity shows that the huge amount of activity performed with 112,475 changes (created, modified, read and deleted). The infected host were connected for up to a minute long and the registry were kept changing per second which makes is it hard to monitored. The network activity described into 2 parts. The first part is the domain names that have discussed in Hypothesis 3 and other part is the connection that being captured by the process monitored by Windows corporation tools, which shows that the infected host were connected and disconnected to different IP addresses that believe to be a suspicious C&C server. The IRC bot clearly run in a background without the notice of the owner of the infected host. What is more, the process activity monitored shows that the IRC bot was loading libraries in order to improve the functionality to improve the use of the infected host. In addition, the IRC bot was creating more than 100 threads then either loading or exiting them. The</p>	<p>has changed not performed clearly in detail. In addition to the physical memory of the infected host does not provide information that called as forensic evidence against the botnet master. The attacks that the botnets performs using the infected host has not identified and experimented in this research.</p>
--	--

<p>physical memory of the infected host shows that running process that the physical memory of the machine is running at the time of the infection. However, the physical memory does not provide any evidence about the infection of the bot but it does support the other evidence such as the file, registry and network activities of the malware in general and bot in particular.</p>	
<p>SUMMARY:</p> <p>The researcher has examined the infected host and the outcome of the investigations of this research shows that the evidence found in the infected host found regards to the activities of the bot. In this research, the result shows that the evidence divided into five sections Files activities, Registry activities, Network activities, loading libraries and physical memory activities. The result shows that with no doubt that the bot performs unauthorized changes to the infected host to be able to control the host without the consent of the owner. The majority of the file activities have been perform in the file system of the windows operating system. In addition, the Registry activities created, modified, read and deleted. The tools that have been used in this research shows the amount of changes that have done in the Registry were a huge amount. Surprisingly, when the infected host by the IRC bot were connected to the secure internet server the Registry were changing continuously and did not stop until the secure internet server was disconnected from the infected host, which the activities were mostly creating and deleting Registry values. The network activities show that the infected host were communicating with the C&C server using the TCP protocol for sending command from the infected host and receiving command from the C&C server. The infected host monitored with sniffer tools that captured commands that have received from the C&C server. The commands that have captured were mainly identifying the operating system of the host as well as other information related to the host such as running process and open ports. The commands of performing</p>	

unauthorized activities did not happen for the research as the communications enabled for a short time. The reason for that is that the research is not willing to be part of performing an attack to another host. Therefore, the commands that meant to perform unauthorized activities has not been part of this research. In addition, the research shows the bot loaded a number of libraries that improve the functionality of the infected host, which the bot loaded at the time of the infection. What is more, The physical memory of the infected host has been preserved and examined the running process in it, however, the physical memory does not support enough evidence about the bot because the activities of its process cannot be viewed but it does show the running process which is to support the other infected host evidence.

5.2 DISCUSSION

This section will be focusing on the important findings that have found in the experimental test results of the infected host. Section 5.2.1 will discuss the infected host environment, section 5.2.2 will discuss the data acquisition and extraction from the infected host, and section 5.2.3 will discuss the reconstruction and analysis; and section 5.2.4 will discuss the recommendations for tracking botnets.

5.2.1 Discussion Of The Infected Host Environment

The experiment of this research was very similar to the real botnets event. The design systems of the infected host used in this research were Windows XP and Windows 7. The reason for choosing the Windows operating system in particular is that the majority of the malware are targeting Windows operating systems. Both version of Windows operating system used as Virtual machines through VMware Workstation (VM). The host infected with many IRC bot and the behaviour of them were mostly similar. However, there were some difference between them such as the number of domains and IP addresses that have connected to, two of the IRC forced the Windows 7 to restart as well as two required the program, which is the bot to run as administrator. The Widows operating system (Windows XP and Windows 7) have not been turned off after the infection of the host and

the operating system have been examined including the physical memory of the infected host. The reason for not turning off the machine is that the physical memory of the infected host deleted at the time of the infection if the host has turned off, when the Windows installed in the VM. The First image of the registry files have taken by Regshot to be able to determine the changes that have done to the infected host. Then the infection of the host takes place and has monitored by the malware tools and other tools that provided by Windows Corporation.

The behaviour of the bot in the infected machine has monitored. The Sniffer tools have monitored the communication traffic of the infected host such as Wireshark and ApateDNS. In addition, the system have been monitored by other tools such as SysinternalsSuite by Windows and the second shot have been taken by the Regshot to compare it with the first shot and identify the changes of the infected host. It is obvious that the investigation of the infected host is time consuming and it takes a large amount of time to be able to determine the signatures and the damage that has done by the botnets. The time of the investigation in a single VM host has taken a large amount of time to collect the data at first and then examine all the data that has gathered from the infected host. Therefore, this could reflect the time that the forensic investigator has to spend in the infected host to collect the data and in a real scenario of a botnet event. The forensic investigator will be require to examine typically more than one single machine such as the Aramco attacks when the total machines that have been infected were more than 30,000 machines (Reuters, 2012). Therefore, the other challenge will be the time of collecting data from those machines, then, examining the data, which requires a large group of Forensic Investigators as well as the time to reach the outcome of the attack and the damage. Overall, the result of this experiment does not reflect resource requirements for the real botnet even; however, it does reflect the time that the forensic investigator procedures take in the single host to be able to collect the data and to be able to examine them later on.

5.2.2 Discussion On Data Acquisition And Extraction From The Infected Host

The malware that have used in this research have collected by the Dionaea honeypot that have installed in the physical machine that have used in this research. The collected malware binaries have been analysed by two methods, which are an external service malware analysis that called Sandbox. The sandboxes that have used in this research were Anubis and ThreatExpert. All the binaries have submitted to those sandboxes except the unknown malware as most of them have 0 KB size. Both sand boxes provide information about the malware, however, they report could be slightly different information about the malware. The Anubis sandbox provided information about the malware in regards to the dependencies that have been found in the malware binary as well as the file, registry and network activities of the malware. The reports that have been created by the Anubis were downloaded in a PDF format that contains at least 40 pages of information about the submitted malware binary. The ThreatExpert was also been participated in this research by submitting all the binaries files to it. Some of the information that have been found in the ThreatExpert were slightly different information in regards to the type of the malware and the threat that the malware cause to the host such as the host can be remotely controlled.

5.2.3 Discussion On Reconstruction & Analysis

The analysis of the malware binaries that have collected by Dionaea honeypot was analysed by the malware analysis tools to be able to understand the nature and the behaviour of the botnets inside the infected host. The malware analysis tools used to be able to analysis the effect and damage to the infected host to be able to have another result that would be useful to compare it to the results that have provided by the sandboxes. The malware analysis tools have provided useful information about the malware, the information was provided that the language that the malware was written on which shows that most of the binaries have been written in C++ language. This means that the developers of the botnets are an advance programmer as the C++ one of the most challenging program languages. In addition, as mentioned earlier that the sniffer tools were installed in the infected host which shows there is a similarity of the information that have been captured

by the sniffer tools and the malware analysis. For example, the malware analysis shows what the malware, which is the bot, in this case pre-programmed to perform in the infected host once the host is infected. The bot transferred the information about the infected host operating system and process A which is a programming variable that is meant to perform a list of actions that the bot is familiar with. The sniffer tools have captured this information that has been transferred by the bot to the suspicious C&C server that is shown by the analysis of the malware analysis.

Furthermore, the live monitored of the infected host was monitored after the infection with being connected to the secure internet server for less than a minute to be able to view the behaviour of the bot with a connection to the isolated internet server and without a connection to the real internet. It was vital to notice that the bot was not performing any suspicious behaviour in the infected host. The infected host was responding to the user as well as there was not any program running after the infection of the bot. Furthermore, the infected host was communicating to different domains and IP addresses in the background without the knowledge of the owner of the machine, which have identified by the sniffer tools that have installed in the infected host prior to the infection of the host. On the other hand, the host that has infected without the connection of the internet, which was trying to connect to the domains and IP addresses but obviously was not able to as the host disconnected from the internet. The infected host then followed by a forensic investigation procedure as the physical memory of the infected host examined without turning the machine off, and dumped it into a file by a tool called Dumpit. Then the physical memory of the infected host examined using Volatility tool that gathered all the information such as running process in the physical memory at the time of the extraction. Other tools have been involved in the investigation of the host such as ApateDNS, Wireshark, Regshot and other tools that are meant to be used in this research to analyse the bot in dynamic as well as live monitored which provided more information about the bot behaviour in the infected host.

5.2.4 Command And Control Communication

The command and control channel C&C communication is the one of the features of the botnets that makes it even more harmful than the other type of the malwares. The C&C server is one of the powerful distinctions of the botnets as the activities of the botnets is unpredictable as the activities and the targets usually rely on the purposes of the botnet master. This research shows the instructions that are sent from the infected host to the suspicious C&C server even though the connection to the secure internet server was for up to a minute. The communications captured by the sniffer tools showed how the bot notified the suspicious C&C server then to the botnet master about the operating system of the infected host. In addition, the bot notified the suspicious C&C server about the other information that the bot is pre-programmed to perform at the time of the infection. The infected host shows that the suspicious C&C notified about the running processes, browsers and other information. The bot also disabled the firewall of the operating system as well as the anti-virus software to be able to stay undetected. This means that the communications will not be able to identified by the owner of the machine unless the owner of the machine has a high computer skills to be able to investigate whether the machine is part of the botnet army or not. A study shows that millions of people that are part of the botnets are not aware of themselves being part of the botnets army.

The C&C communications channel is one of the most complicated parts of the botnets threat because they can be used different methods to encrypt the communications that disabled others to identify the content of them. However, the sniffer tools shows that the botnets tried to scan the network IP addresses of the infected host after the infection that may lead to infect other machine. The researcher's network was 192.XXX.XX.1. The bot tried to scan the network of the researcher from 192.XXX.XX.2 to 192.XXX.XX.12. These scans of the researcher's network captured on domain name service (DNS) on the Wireshark tool.

The IRC bot that this research performed the experiment on, which shows that the IRC bot was using the TCP protocol to communicate with the suspicious C&C server. The instructions sent in plain text format. The researcher then disconnects the infected host from the secure internet server for a while then

connected again. The purpose of this step is to try to capture the checking the statues of the bot in the infected host. The previous research that was reviewed in chapter 2 shows that the C&C server checks the statues of the bot in the infected host. The sniffer tools could not capture this stage and there was no information leading to the suspicious C&C server-checking statues sent in a plain text like the instructions that captured. This means that the checking process of the statues of the bot in the infected host encrypted and it is difficult to be able to capture this type of information using the sniffer tools.

5.2.5 Recommendation On Tracking Botnets

This section will discuss the possible steps that should be taken into consideration to be able to track botnets and bot masters.

5.2.5.1 Cross Border Issues

The name botnet event usually involves international incidents where the victims of the botnet attack would be from many countries on different continents. The main issues and challenge that would face the effort of stopping this type of incident is that there are still countries that do not have a cybercrime law, which means that performing the attack or cybercrime activities is not a crime in these countries. There are many examples of botnets events where the botnet master prosecution would require an international effort such as Aramco Oil Company that is located in Saudi Arabia that have been attacked by a botnet masters group. The internal investigation of the incident shows that the damage that caused by the event were severe with more than 30,000 were damage and the attack were performed from four countries in four continents (Reuters, 2012). Another example of the Cross Border Issues is that when the Mariposa botnet masters managed to steal sensitive information from 800,000 users across 190 countries. These two examples show just how the international effort should be gathered to be able to stop this type of cybercrime from destroying the internet environment by implementing an international cybercrime law to be able to stop these computing criminal from keep performing their cyber activities. The joint international effort was able to arrest the three Mariposa botnet masters in Spain, however, the effort is still needs a long way to go as many countries do not have

cybercrime law. Overall, the cross-border-issue is one of the challenges to the international effort to stop this type of crime as many of these hackers would think again before performing a cyber-attack as the consequences would stop some of them from being part of these electronic crimes.

5.2.5.2 Tracking The Botnet Master

It is obvious now that the command and control channel (C&C) can be located by its domains and IP addresses and could be blocked. This would not stop the C&C server as the fast-flux is a new technology that enables the C&C server to change its IP frequently. Furthermore, the botnet master usually have more than 200 servers that being hosts from different providers. Therefore, blocking or tracking the C&C server is not a useful step to be taken anymore to stop this type of event from occurring. The only way to stop the botnets from having more victims involved in the botnet army is to take down the botnet master. The reason for that is that examining the botnet master machine would enable the forensic investigators to be to locate all the possible C&C servers and take them down. For example when the police in Spain arrested the three Mariposa botnet masters whose have compromised more than 12 million hosts.

Overall, in order to be able to take down the botnet master the international effort should be placed to be able to arrest the botnet master and take down the botnets. The cross-border issue has to be resolved to be able to take down the botnet master and take down the botnets.

5.2.5.3 Improving The Detection Of The Botnets

As there are many researchers that are studying parts of the botnet challenge. The detection of the botnets is not effective enough to be able to detect all the botnets. Most of the detection techniques that are used are monitoring the traffic of the system to be able to identify any possible existence of the botnets. However, many people who use the computers have a lack of knowledge of dealing with malware and the botnets. Therefore, as the botnet aims to be undetected in the host, this means that the owner of the machines will not be able to detect the existence of the botnets in the machine. This may lead to the machine used in performing attacks to the other host as well as assisting to spread out the botnets

and increase the botnet army. In addition, the anti-virus and firewall is able to detect the other types of malware, nevertheless, the existence of the bot in the host will block the anti-virus and firewalls and any other security program to be able to control the host. Therefore, the security of the host should be improved to be able to detect botnets in the host.

5.3 CONCLUSION

This chapter discussed the findings of this research that presented in chapter 4. The findings of the research, enabled the researcher to answer the main research question and the sub-questions. The main research question and the sub-questions of this research have presented in chapter 3 and relevant findings presented in sections 4.3 and 4.4. The asserted hypotheses were tested accordingly with the arguments made for and against in order to justify the asserted hypothesis as accepted, rejected or indeterminate. The limitations of this research and the challenges have discussed.

The research aims to find the evidence that could be found in the infected host from the bot. The research also was aiming to identify the communication between the infected host and the command and control channel which is the server that sends the instruction to the bot in the infected host. The research was able to find the evidence of the existence of the bot in the infected host as well as the changes that have been performing the infected host without the knowledge of the owner of the host. The research also was able to identify the suspicious command and control channel that captured by the sniffer tools installed in the infected host prior to the infection.

Overall, this chapter has answered the main research question and sub-questions. In addition, the evidence was collected must be kept for further analysis and for further forensic investigation. However, the next chapter, Chapter 6 will present a conclusion of this research that outlines the significant research findings. What is more, the importance of the future research work that will assist to improve the detection of this type of malware.

Chapter 6

Conclusion

6.0 INTRODUCTION

This chapter is the conclusion of the entire thesis project and presents the final conclusions and suggestions for further research. The conclusion is mainly based on the chapter 4 the findings and chapter 5 the discussion of the findings. The gap in botnet research has been identified and presented in the problem identification in chapter 2 and the chapter 3 methodology development. This chapter will present the possibilities for future work that could be under-taken.

Two external sandbox services were used for this research to be able to analyze the malware of the botnet. The Anubis and ThreatExpert exploits the malware in a safe environment and reports back the analysis report. The malware analysis tools were able to provide information about the activities that the bot needs to perform at the time of the infection of the host. The malware analysis shows that the information gathered by the external sandboxes services were almost 90% similar. In addition, the malware analysis shows the loading libraries that the bot loaded in the infected host to improve the functionality in the infected host.

The live monitoring of the infected host shows how the domain names and the IP address from the infected host may be observed. In addition, the instructions sent from and to the infected host have been captured with the sniffer tools. However, the connection to the secure internet server was for a really short time to prevent the possibilities of using the host to perform an attack other people. The Kernel 32 and other information were captured which provides information about the host to the C&C server then to the botnet master. The host was changing continuously while it was connected to the secure internet server. The changes to the host were surprisingly large with registry changes totalling 112,475, the files system changes total of 3,709, the network activities total of 8,303 and the process activities total of 466. The activities of the network in this situation means when the infected host connects and disconnects from domains names or IP addresses. The changes show what the botnet is capable of performing in the infected host.

The inbound and outbound traffic captured by Wireshark showed that there was information sent to and from the infected host. Furthermore, the research was able to capture sensitive information being transferred to the suspicious C&C server.

Overall, the evidence that was found in the infected host was found in the registry activities, file system, network activities and the physical memory of the infected host. Some of bots did not let the researcher to perform an examination of the physical memory of the infected host as the bot forced the host to reboot straight after the infection. The analysis malware tools and sniffer tools shows that the existence of the botnets is proved and the communication with the C&C server is involved in the infected host. The communications between the C&C server and the infected host were captured and the infected host were changing continuously during the connection as well as connecting to more than 200 hosts in a really short period of time.

Nevertheless, this experiment shows what the botnet is capable of performing in the infected host and this is only a single host. The issues of the botnets typically have an army of millions of infected hosts. In addition, it is often driven by cybercrime organizations that intend to harm the internet security and to gain a financial reward.

6.1 LIMITATIONS OF THE RESEARH

The research limitations have been discussed in chapter 3 section 3.4 addressing the areas that are out of the scope of this research. The limitations are presented in section 3.4 and in addition to the limitation that has been found during the experiment of this research will be presented in this section 6.1.

The thesis project managed to capture the traffic between the C&C server and the infected host. However, the thesis project did not manage to capture the instructions of the infected host performing an attack to other targets such as organizations. The reason for not reaching the attacks instructions stage is that it becomes a threat to other people and the research aims to prevent the attack and not to be part of the attacks. Therefore, the C&C server attack instructions are still a limitation of this research.

In addition, the research was able to analysis most of the bots captured through Dionaea honeypot, however, most of the bots captured were IRC bots. This means that the other types of bots needs to be studied in order to achieve a

better understanding of botnets. For example the P2P bot has not been examined in this research due to the limited budget for this thesis project.

The code of the botnets was one of the limitations of this thesis project. The reason for that is that the botnet code is one of the hardest codes to understand for many reasons. Firstly, it is typically written in more than 15,000 of lines which is a time consumption to understand, therefore, this thesis project has a limited time to be completed which means that the analysis of the botnets code needs to be studied in a separate research project.

6.2 FUTURE RESEARCH

This research has performed a malware collection to be able to collect malware for the experiment purpose of this research. In addition, this research analyses the malwares that have been collecting using Dionaea honeypot and has uploaded them into the external sandboxes services. The external sandboxes (ThreatExpert and Anubis) services provided analysis of the malware uploaded with different information that assist this research to have a better understanding of the botnets. Furthermore, this research performed an analysis of the malware collected using malware analysis. The information gathered from the external sandboxes services and malware analysis tools provided a higher level of understanding of what the botnet is capable of. In addition, this research was able to capture the C&C server communications between the C&C server and the infected host.

The detection techniques of the botnets is still needs to be improving to be able to detect the botnet existence in the host. The signature of the botnets detection techniques works for the detected malware but it does not work for the new botnet signatures. This means that the zero attack is the opportunity for the botnet master to be able to infect many victims. The reason for that is that by the time of detecting the botnet signature, then updating the database and then updating the database in the end user side, the end user might be already be being infected by the botnets. Therefore, the detection of the botnets prior to the infection of the botnets is essential to be able to prevent millions of machines infected by the botnets. This is one of the most difficult future studies of the botnets as there are many research projects focused in this area. However, the

detection of the botnets needs to more research effort put into it to be able to improve the detection of the botnet.

The command and control (C&C) channel is one of the most complicated parts of the botnet area. The C&C server needs to be studied in a completely separate research project. The reason for that is that the C&C server has many areas involved in it. The C&C server is the leader of the bot army, which means that controlling millions of infected hosts using this server. The research needs to study the number of hosts that are typically involved in the hosting of the C&C communication. In addition, the type of communications used by each type of botnet such as IRC bot, HTTP bot and P2P bot. What is more, the destruction of the C&C server is another work that required to be performed by other future researchers. This is a higher-level research work that needs to be able to destroy the C&C server. The destruction of the C&C server is the most recommended future research to be able to shut down the C&C server easily. However, this is a large future research work as the number of C&C servers could be hosting more than 200 hosts.

In addition, tracking the botnet master is another work for a higher budget research work. Tracking the botnet master and being able to locate the botnet master rather than the C&C server. The reason for that is that the C&C server is hosted by a large number of hosts and taking the effort to block the C&C is not an appropriate solution anymore. The botnet master is able to create a new C&C server in a different host and then register a new host as the previous C&C and re-join the bots to the new C&C server. Tracking the botnet master is the solution to be able to take down the botnets completely as this approach is able to shut down all the hosts of the C&C server.

REFERENCES

- Anubis. (2014). *Anubis - Learn about the Anubis Malware Analysis Tool*. Retrieved January 15, 2014, from <https://anubis.isecclab.org/?action=about>
- The National Cyber Security Centre (NCSC). (2013, February 8). *NCSC – 2012 Incident Report Summary / NCSC*. Retrieved August 12, 2013, from <http://www.ncsc.govt.nz/newsroom/ncsc-2012-incident-report-summary/>
- The National Cyber Security Centre (NCSC). (2012, June 28). *NCSC – 2011 Incident Summary / NCSC*. Retrieved August 12, 2013, from <http://www.ncsc.govt.nz/newsroom/ncsc-2011-incident-summary/>
- Bailey, M., Cooke, E., Jahanian, F., Xu, Y., & Karir, M. (2009). A Survey of Botnet Technology and Defenses. *Paper presented at 2009 Cybersecurity Applications & Technology Conference for Homeland Security*, 299-304. doi:10.1109/CATCH.2009.40
- Boshmaf, Y., Muslukhov, I., Beznosov, K., & Ripeanu, M. (2013). Design and analysis of a social botnet. *Computer Networks*, 57(2), 556-578. doi:10.1016/j.comnet.2012.06.006
- Britz, M. (2009). *Computer Forensics and Cyber Crime: An introduction*. Upper Saddle River, N.J: Pearson Prentice Hall
- Chiang, K., & Lloyd, L. (2007). A Case Study of the Rustock Rootkit and Spam Bot. Sandia National Laboratories. Livermore, CA. 1-8.
- Choi, Y.-H., Liu, P., & Seo, S.-W. (2010). Creation of the importance scanning worm using information collected by Botnets. *Computer Communications*, 33(6), 676-688. doi:10.1016/j.comcom.2009.11.012
- Choo, i.-K. R. (2007). Zombies and botnets. *Trends & Issues in crime and criminal justice*, 1-6.

- Clark, C., Chaffin, L., Chuvakin, A., Paladino, S., Dunkel, D., Fogie, S., Gregg, M., Grossman, J., Hansen, R., Petkov, P., Rager, A., & Schiller, C., (2008). InfoSecurity 2008 Threat Analysis. Burlington, MA: Syngress.
- Colajanni, M., Daniele Gozzi, & Marchetti, M. (2008). Collaborative architecture for malware detection and analysis. *Proceedings of The Ifip Tc 11 23rd International Information Security Conference IFIP – The International Federation for Information Processing*, 278, 79-93. doi: 10.1007/978-0-387-09699-5_6
- Correia, P., Rocha, E., Nogueira, A., & Salvador, P. (2012). Statistical Characterization of the Botnets C&C Traffic. *Procedia Technology*, 1, 158-166. doi:10.1016/j.protcy.2012.02.030
- Daswani, N., & Stoppelman, M. (2007). The Anatomy of Clickbot.A, 1-11.
- Dietrich, C. J., Rossow, C., & Pohlmann, N. (2013). CoCoSpot: Clustering and recognizing botnet command and control channels using traffic analysis. *Computer Networks*, 57(2), 475-486. doi:10.1016/j.comnet.2012.06.019
- Etemad, F. F., & Vahdani, P. (2012). Real-Time Botnet Command and Control Characterization at the Host Level. *6'th International Symposium on Telecommunications (IST'2012)*, 1005-1009. doi: 10.1109/ISTEL.2012.6483133
- Feily, M., & Shahrestani, A. (2009). A Survey of Botnet and Botnet Detection. *Presented at Emerging Security Information, Systems and Technologies*. doi:10.1109/SECURWARE.2009.48
- Gordon, S., & Ford, R. (2006). On the definition and classification of cybercrime. *Springer-Verlag France 2006*, 13-20. doi:DOI 10.1007/s11416-006-0015-z

- Grizzard, J. B., Sharma, V., & Nunnery, C. (2007). Peer-to-Peer Botnets: Overview and Case Study, 1-8.
- Hay, B., & Nance, K. (2008). Forensics Examination of Volatile System Data Using Virtual Introspection. *SIGOPS Operating Systems Review*, 42(3), 74-82.
- Heron, S. (2007). Working the botnet: how dynamic DNS is revitalising the zombie army. *Network Security*, 2007(1), 9-11. doi:10.1016/s1353-4858(07)70005-3
- Jang, D.-i., Kim, M., Jung, H.-c., & Noh, B.-N. (2009). Analysis of HTTP2P Botnet : Case Study Waledac. *Proceedings of the 2009 IEEE 9th Malaysia International Conference on Communications 15 -17 December 2009 Kuala Lumpur Malaysia*, 409-412.
- Kharouni, L. (2009). SDBOT IRC Botnet Continues to Make Waves. *A Trend Micro White Paper*, 1-20.
- Lee, C. P. (2009). Framework for Botnet Emulation and Analysis. *Georgia Institute of Technology* (Doctor of Philosophy in the School of Electrical and Computer Engineering, Atlanta, Georgia, in the United States). Retrieved from <http://www.chrislee.dhs.org/projects/rubot/rubot-thesis.pdf>
- Lee, S., & Kim, J. (2013). Fluxing botnet command and control channels with URL shortening services. *Computer Communications*, 36(3), 320-332. doi:10.1016/j.comcom.2012.10.003
- Li, C., Jiang, W., & Zou, X. (2009). Botnet: Survey and Case Study. *2009 Fourth International Conference on Innovative Computing, Information and Control*, 1184-1187. doi: 10.1109/ICICIC.2009.127

- Lu, W., Rammidi, G., & Ghorbani, A. A. (2011). Clustering botnet communication traffic based on n-gram feature selection. *Computer Communications*, 34(3), 502-514. doi:10.1016/j.comcom.2010.04.007
- Pham, V.-H., & Dacier, M. (2011). Honeypot trace forensics: The observation viewpoint matters. *Future Generation Computer Systems*, 27(5), 539-546. doi:10.1016/j.future.2010.06.004
- Rajab, M. A., Zarfoss, J., Monroe, F., & Terzis, A. (2006). A Multifaceted Approach to Understanding the Botnet. 41-52.
- Reuters. (2012, December 9). Aramco Says Cyberattack Was Aimed at Production. *The New York Times* [Jeddah]. Retrieved from http://www.nytimes.com/2012/12/10/business/global/saudi-aramco-says-hackers-took-aim-at-its-production.html?_r=0
- Rodríguez-Gomez, R. A., Andez, G. M. A.-F., & García-Teodoro, P. (2013). Survey and Taxonomy of Botnet Research through Life-Cycle. *ACM Computing Surveys*, 45(4), 1-45.
- Rouse, M. (2013). What is bot (robot)?. *Definition from WhatIs.com*. Retrieved January 10, 2014, from <http://searchsoa.techtarget.com/definition/bot>
- Rrushi, J., Mokhtari, E., & Ghorbani, A. A. (2011). Estimating botnet virulence within mathematical models of botnet propagation dynamics. *Computers & Security*, 30(8), 791-802. doi:10.1016/j.cose.2011.07.004
- Schiller, C. A., Binkley, J., Harley, D., Evron, G., Bradley, T., Willems, C., & Cross, M. (2007). *Botnet The Killer Web App*. Rockland, MA: Syngress.
- Shahrestani, A., Feily, M., Masood, M., & Muniandy, B. (2012). Visualization of Invariant Bot Behavior for Effective Botnet Traffic Detection. *International*

- Symposium on Telecommunication Technologies*, 325-330. doi: 10.1109/ISTT.2012.6481606
- Silva, S. S. C., Silva, R. M. P., Pinto, R. C. G., & Salles, R. M. (2013). Botnets: A survey. *Computer Networks*, 57(2), 378-403. doi:10.1016/j.comnet.2012.07.021
- Sinha, P., Boukhtouta, A., Belarde, V. H., & Debbabi, M. (2010). Insights from the Analysis of the Mariposa Botnet. *Risks and Security of Internet and Systems (CRiSIS), 2010 Fifth International Conference*, 1-9. doi: 10.1109/CRISIS.2010.5764915
- Song, L.-P., Jin, Z., & Sun, G.-Q. (2011). Modeling and analyzing of botnet interactions. *Physica A: Statistical Mechanics and its Applications*, 390(2), 347-358. doi:10.1016/j.physa.2010.10.001
- Techterms.com. (n.d.). *Malware Definition*. Retrieved May 18, 2014, from <http://www.techterms.com/definition/malware>
- ThreatExpert. (2009). *ThreatExpert - Introduction*. Retrieved January 18, 2014, from <http://www.threatexpert.com/introduction.aspx>
- Ullah, I., Khan, N., & Aboalsamh, H. A. (2013). Survey on Botnet: Its Architecture, Detection, Prevention and Mitigation. *Networking, Sensing and Control (ICNSC), 2013 10th IEEE International Conference*, 660-665. doi: 10.1109/ICNSC.2013.6548817
- Wang, K., Huang, C.-Y., Lin, S.-J., & Lin, Y.-D. (2011). A fuzzy pattern-based filtering algorithm for botnet detection. *Computer Networks*, 55(15), 3275-3286. doi:10.1016/j.comnet.2011.05.026

- Xie, Y., Yu, F., Achan, K., Panigrahy, R., Hulten, G., & Osipkov, I. (2008). Spamming Botnets: Signatures and Characteristics. *Computer Networks* 55 (2011), 3275- 3285.
- Yan, G., Ha, D. T., & Eidenbenz, S. (2011). AntBot: Anti-pollution peer-to-peer botnets. *Computer Networks*, 55(8), 1941-1956. doi:10.1016/j.comnet.2011.02.006
- Yen, P.-H., Yang, C.-H., & Ahn, T.-N. (2009). Design and Implementation of a Live-analysis Digital Forensic System. *International Conference on Convergence and Hybrid Information Technology*, 239-243.
- Zahid, M., Belmekki, A., & Mezrioui, A. (2012). A new architecture for detecting DDoS/Brute forcing attack and destroying the botnet behind. *Multimedia Computing and Systems (ICMCS), 2012 International Conference*, 1-5. doi: 10.1109/ICMCS.2012.6320256
- Zhao, D., Traore, I., Sayed, B., Lu, W., Saad, S., Ghorbani, A., & Garant, D. (2013). Botnet detection based on traffic behavior analysis and flow intervals. *Computers & Security*, 1-15. doi:10.1016/j.cose.2013.04.007
- Zhao, S., Lee, P. P. C., Lui, J. C. S., Guan, X., Ma, X., & Tao, J. (2012). Cloud-based push-styled mobile botnets: a case study of exploiting the cloud to device messaging service. *the 28th Annual Computer Security Applications Conference*, 119-128 doi:10.1145/2420950.2420968
- Zhuge, J., Holz, T., Han, X., Guo, J., & Zou, W. (2007). Characterizing the IRC-based Botnet Phenomenon. *Reihe Informatik*, 1-16.

APPENDICES

Appendix 1

Dionaea Installation Script.

(The installation script has copied from the Dionaea's Website. The reason for listing mentioning the script is to show how the Dionaea has been installed for this research)

Ubuntu

```
aptitude install libudns-dev libglib2.0-dev libssl-dev libcurl4-openssl-  
dev \  
libreadline-dev libsqlite3-dev python-dev \  
libtool automake autoconf build-essential \  
subversion git-core \  
flex bison \  
pkg-config
```

tar xfz ...

libglib (debian <= etch)

liblcfg (all)

```
git clone git://git.carnivore.it/liblcfg.git liblcfg  
cd liblcfg/code  
autoreconf -vi  
./configure --prefix=/opt/dionaea  
make install  
cd ..  
cd ..
```

libemu (all)

```
git clone git://git.carnivore.it/libemu.git libemu  
cd libemu  
autoreconf -vi  
./configure --prefix=/opt/dionaea  
make install  
cd ..
```

libnl (linux && optional)

```
apt-get install libnl-3-dev libnl-genl-3-dev libnl-nf-3-dev libnl-route-3-
```

```
dev
```

else install it from git.

```
git clone git://git.infradead.org/users/tgr/libnl.git
cd libnl
autoreconf -vi
export LDFLAGS=-Wl,-rpath,/opt/dionaea/lib
./configure --prefix=/opt/dionaea
make
make install
cd ..
```

libev (all)

```
wget http://dist.schmorp.de/libev/Attic/libev-4.04.tar.gz
tar xzf libev-4.04.tar.gz
cd libev-4.04
./configure --prefix=/opt/dionaea
make install
cd ..
```

Python

```
wget http://www.python.org/ftp/python/3.2.2/Python-3.2.2.tgz
tar xzf Python-3.2.2.tgz
cd Python-3.2.2/
./configure --enable-shared --prefix=/opt/dionaea --with-computed-gotos \
    --enable-ipv6 LDFLAGS="-Wl,-rpath=/opt/dionaea/lib/ -
L/usr/lib/x86_64-linux-gnu/"
make
make install
```

Cython (all)

We have to use cython >= 0.15 as previous releases do not support Python3.2
__hash__'s Py_Hash_type for x86.

```
wget http://cython.org/release/Cython-0.15.tar.gz
tar xzf Cython-0.15.tar.gz
cd Cython-0.15
/opt/dionaea/bin/python3 setup.py install
cd ..
```

udns (!ubuntu)

```
wget http://www.corpit.ru/mjt/udns/old/udns_0.0.9.tar.gz
tar xzf udns_0.0.9.tar.gz
cd udns-0.0.9/
```

```
./configure
make shared
```

There is no make install, so we copy the header to our include directory.

```
cp udns.h /opt/dionaea/include/
and the lib to our library directory.
cp *.so* /opt/dionaea/lib/
cd /opt/dionaea/lib
ln -s libudns.so.0 libudns.so
cd -
cd ..
```

libpcap (most)

```
wget http://www.tcpdump.org/release/libpcap-1.1.1.tar.gz
tar xzf libpcap-1.1.1.tar.gz
cd libpcap-1.1.1
./configure --prefix=/opt/dionaea
make
make install
cd ..
```

Compiling dionaea

```
git clone git://git.carnivore.it/dionaea.git dionaea
then ..
cd dionaea
autoreconf -vi
./configure --with-lcfg-include=/opt/dionaea/include/ \
--with-lcfg-lib=/opt/dionaea/lib/ \
--with-python=/opt/dionaea/bin/python3.2 \
--with-cython-dir=/opt/dionaea/bin \
--with-udns-include=/opt/dionaea/include/ \
--with-udns-lib=/opt/dionaea/lib/ \
--with-emu-include=/opt/dionaea/include/ \
--with-emu-lib=/opt/dionaea/lib/ \
--with-gc-include=/usr/include/gc \
--with-ev-include=/opt/dionaea/include \
--with-ev-lib=/opt/dionaea/lib \
--with-nl-include=/opt/dionaea/include \
--with-nl-lib=/opt/dionaea/lib/ \
--with-curl-config=/usr/bin/ \
--with-pcap-include=/opt/dionaea/include \
--with-pcap-lib=/opt/dionaea/lib/
make
make install
```

Appendix 2

Malware MD5	Type
a650c67e14cfb27879999036741478d5	IRC bot
0a278f8d72e4d3d2d44485764398c84d	IRC bot
706b0c15ac6206298fabb68c432e93f5	IRC bot
06965414B531915726B24523263B9C12	IRC bot
c2e9a9884a40f242bac1d7d9fe39056d	IRC bot
3a97d25ada27b727ae4ee6a1f7050546	IRC bot
2315ebb40c11bc349e2d660dd0105a06	IRC bot
1b72419f00e25d657c3ba74bb189de47	IRC bot
1db61ae18c85d6aca77a4a3800af07b4	IRC bot
1b68d6ebec876704a5414ad638c93bd3	IRC bot
a2e26ff29944a44d6f632e26931a4936	IRC bot
6e6985e4684c03282eebc6b55380c269	IRC bot
251616a9205e376778b261330b11da9b	IRC bot
3a4c590f30be34684125e1c974fe13c6	Backdoor
360b11c542d3926e254af6439bcd151	Backdoor
20ccb3d22de6857e350c95dc866b71cd	Backdoor
b591da6d2233fd3053aa55d2a0e473f1	Backdoor
c000f32147ba346e7543ca07a5e9dc16	Backdoor
2d68ee6d3666ac1ef27d85aab144b09e	Backdoor
a2eea7882ae094f1b5f181d482b6d281	Backdoor
19d3c2833878a56c694c544735f67674	Torjan
786ab616239814616642ba4438df78a9	Torjan
1f4c43adfd45381cfdad1fafeal6b808	Torjan
f2d9e278bfca9e93578a8ea9536da93a	Torjan
7867de13bf22a7f3e3559044053e33e7	Torjan
3ca30fdc5e4b2150f42aa09ba37f326e	Torjan
4d4c2729b8aa56e70eaf9ef84e9d5d3d	Torjan
b1cf9504f90372cc8697c1870cee7734	Torjan
065172e07a125623ea0a0fbcdaaa6dee	Torjan
496f0929c4f95f2053b1d4da6a05a3ef	Torjan
2d892b54776407b32ad19a691acaed05	Torjan
7657fcb7d772448a6d8504e4b20168b8	Torjan
26c7885b95501af4da1ffa621f793027	Torjan
0f0a3eeeccdadc6711b3745e9444aba9	Worm
31c33d00a9eee8ca01a0495da2654b06	Worm
d401881cf9aadd1b7705fb7cc1458536	Worm
903b591da5dfc0268b062ac16b4dee31	Worm
b9d04b4adfb16d9ba2cdfaf9820aaf2	Worm
b91241a4f52f90e1ecc6596a357f7602	Worm

235c33bd3673a690b7d92db9b4e84176	Virus
78580b5a9b12f8d75a0b23fbb1460ea5	Virus
d41d8cd98f00b204e9800998ecf8427e	unknown
f8ee428e4d73df4ab34debcd7947da98	unknown
a670deb3dd6febfcfda8392305041657	unknown
0a2d9f2db8e53ba5b8e8336b08b62f16	unknown
0f34b4f178f27aeb67507f49f3476e36	unknown
1b72d870bc551a6287237e487eba0d50	unknown
1b790fe248432412933958035faba106	unknown
1b722ed6019599026ea5cb023b05a0c7	unknown
1b84f900b568e9987c66bcdca398152f	unknown
1b8e88fa0ef2c8a43c77f24e77f4bb21	unknown
55f9bb14d4e205df91636a22e5477420	unknown
1b97039c91aab8d4556455d026685450	unknown
1b90e2f87e383dd5fd9ff70d757d0c38	unknown
22c16fd590d3c7efb60882acf0591270	unknown
3a5a0fcf137af0ea046d4a903617f7c1	unknown
3a6329f86bda47213101bdde1972f906	unknown
235d0b6681e4067bd3c0850025c70e06	unknown
3ad2a54e654d235048207897310369a0	unknown

Appendix 3

Regshot 1.8.3-beta2

Comments:

Datetime:2014/5/1 10:42:01 , 2014/5/1 10:51:09

Computer:WIN-NMRAPVI2Q68 , WIN-NMRAPVI2Q68

Username:sultan , sultan

Keys deleted:32194

HKLM\COMPONENTS

HKLM\COMPONENTS\CanonicalData

HKLM\COMPONENTS\CanonicalData\Catalogs

HKLM\COMPONENTS\CanonicalData\Catalogs\004032bbaef889cfb12e3e9ac2a
efabb6e9edb84f19400473eac4fc733e845b0

HKLM\COMPONENTS\CanonicalData\Catalogs\02508beadc145dc6a0851af799
1867abff1abdfa4899760a7c727b092f3f59d7

HKLM\COMPONENTS\CanonicalData\Catalogs\0329cabe5e666fea68d1c148
4e222f7b37de11ae5201c1d78e608c40067857

HKLM\COMPONENTS\CanonicalData\Catalogs\036d29818ce211f90bafd5db1e
6f1301aeb71e7a86c9622e403a73b3c54f6335

HKLM\COMPONENTS\CanonicalData\Catalogs\0377a7ab352e740ac1495cd361
1726ebc786bc0bd90e4a487901847d5fa5fc5e

HKLM\COMPONENTS\CanonicalData\Catalogs\040727d8ba2cb23eeeecdc84cf
209781f8c7731ab174547f1dd436422b565883

HKLM\COMPONENTS\CanonicalData\Catalogs\044e36836db3ef91292eb01e5b
c4901b2f7b0fb8f9995fdb3a9cf92bc1140efd

HKLM\COMPONENTS\CanonicalData\Catalogs\05ceb27a547db4fe3c8cbacae92
aa2e222ca05dd6265afb3df3346d8e0eccee2

HKLM\COMPONENTS\CanonicalData\Catalogs\0647a384f024f34e91dfb0944b
84f83e032fa332c0d636443d2fc6a5aa8496d1

HKLM\COMPONENTS\CanonicalData\Catalogs\064eee7b5056b900b4628d168a
d714db70901c8e0bf11f9c97a8ff4751c5748f

HKLM\COMPONENTS\CanonicalData\Catalogs\0669b5ecb77f2a32867dbd4711
383895133489254455e1f0ca394e732666c993

HKLM\COMPONENTS\CanonicalData\Catalogs\069c99928ad56c2b2e0f4e6213
a6c35e2385f021ce6c150d925777780b6134ce

HKLM\COMPONENTS\CanonicalData\Catalogs\074e11461559875e47a768b47
d55be7cad200d5b8a981044baf5a8dcbe8c7712

HKLM\COMPONENTS\CanonicalData\Catalogs\0853955a72862f2558cf2b54f8f
148dbc4900f944b33b15e3bb10149da93dd1c

HKLM\COMPONENTS\CanonicalData\Catalogs\09c1e4839d5aba58cff1b59070
5e27c20c195ca6a88d5c6de1e510e8577c82b3

HKLM\COMPONENTS\CanonicalData\Catalogs\0afbf70342a22fac70e9d17ccd4
bac8645bb9b8e88d10f10dc05af7d4948116b

HKLM\COMPONENTS\CanonicalData\Catalogs\0b19afcd0b6ce2ce6c10b8fa4d3
9c48d4d874d8a56ca4a749bc0c601730f6c13

HKLM\COMPONENTS\CanonicalData\Catalogs\0c04c6d6b12b35cdcecb880b98
1b10fd4418874ee91539227f21ad6732599a35

HKLM\COMPONENTS\CanonicalData\Catalogs\0d31fa0c10121232edbb092b4b
2a654aab13d8c454f821ac4bc5b820accd1cc1

HKLM\COMPONENTS\CanonicalData\Catalogs\0dba7ce00969ac2f154b4b9c5e
08022ef40125c16a3bf84509b6d17ec623225f

HKLM\COMPONENTS\CanonicalData\Catalogs\0f129505504ccd1bc5b833b8c3
bd67fb9e1c23f1c0edd2c912afe8f039ac634f

HKLM\COMPONENTS\CanonicalData\Catalogs\1030cd2b2deb7ff340c121f3a7e
713616f53d8eef23f3055a9158626ec5408aa

HKLM\COMPONENTS\CanonicalData\Catalogs\111f8c6f9261d9e5a568297313
83aecdfc4f0bee7896abee1a06148ef22ec079

HKLM\COMPONENTS\CanonicalData\Deployments\xnacc.inf_31bf3856ad364e
35_6.1.7600.16385_b381dfe1d4da7da9

HKLM\COMPONENTS\ccpinterface

HKLM\COMPONENTS\Configuration

HKLM\COMPONENTS\DerivedData

HKLM\COMPONENTS\DerivedData\Components

HKLM\COMPONENTS\DerivedData\Components\msil_accessibility_b03f5f7f1
1d50a3a_6.1.7600.16385_none_2232298e4f48d6ba

HKLM\COMPONENTS\DerivedData\Components\msil_addinprocess_b77a5c56
1934e089_6.1.7601.17514_none_f9a5b9a7f0e068e4

HKLM\COMPONENTS\DerivedData\Components\msil_addinutil_b77a5c56193
4e089_6.1.7601.17514_none_1a816bc7556b71eb

HKLM\COMPONENTS\DerivedData\Components\msil_aspnetmmcontext.resources
_b03f5f7f11d50a3a_6.1.7600.16385_en-us_1d29e1e36ee548cc

HKLM\COMPONENTS\DerivedData\Components\msil_aspnetmmcext_b03f5f7f11d50a3a_6.1.7600.16385_none_54ffde5552ddf5e9

HKLM\COMPONENTS\DerivedData\Components\msil_aspnet_compiler.resources_b03f5f7f11d50a3a_6.1.7600.16385_en-us_18626f3678f342b6

HKLM\COMPONENTS\DerivedData\Components\msil_aspnet_regbrowsers.resources_b03f5f7f11d50a3a_6.1.7600.16385_en-us_dcce6cedc0f76e7e

HKLM\COMPONENTS\DerivedData\Components\msil_aspnet_regsql.resources_b03f5f7f11d50a3a_6.1.7600.16385_en-us_696aa04f9de29ac9

HKLM\COMPONENTS\DerivedData\Components\msil_caspol.resources_b03f5f7f11d50a3a_6.1.7600.16385_en-us_82448578a2be9841

HKLM\COMPONENTS\DerivedData\Components\msil_comsvconfig.resources_b03f5f7f11d50a3a_6.1.7600.16385_en-us_473893ee91bba5b8

HKLM\COMPONENTS\DerivedData\Components\msil_comsvconfig_b03f5f7f11d50a3a_6.1.7601.17514_none_bfe4d387913dbb8f

HKLM\COMPONENTS\DerivedData\Components\msil_cscompmgd_b03f5f7f11d50a3a_6.1.7600.16385_none_ed1eb8fd6654bbd7

HKLM\COMPONENTS\DerivedData\Components\msil_datasvcutil.resources_b77a5c561934e089_6.1.7600.16385_en-us_d3d1b9e1b06af0b6

HKLM\COMPONENTS\DerivedData\Components\msil_datasvcutil_b77a5c561934e089_6.1.7601.17514_none_cfdc452bbab5ec47

HKLM\COMPONENTS\DerivedData\Components\msil_dfsvc_b03f5f7f11d50a3a_6.1.7600.16385_none_3a54952b454a8916

HKLM\COMPONENTS\Installers

HKLM\COMPONENTS\Installers\RegKeySDTable

HKLM\COMPONENTS\ServicingStackVersions

Keys added:207

HKLM\SOFTWARE\Classes\.5vw

HKLM\SOFTWARE\Classes\.acp

HKLM\SOFTWARE\Classes\.apc

HKLM\SOFTWARE\Classes\.atc

HKLM\SOFTWARE\Classes\.bfr

HKLM\SOFTWARE\Classes\.cap

HKLM\SOFTWARE\Classes\.enc

HKLM\SOFTWARE\Classes\.erf

HKLM\SOFTWARE\Classes\.fdc

HKLM\SOFTWARE\Classes\.ntar

HKLM\SOFTWARE\Classes\.out

HKLM\SOFTWARE\Classes\.pcap

Values deleted:78958

HKLM\COMPONENTS\CanonicalData\Catalogs\ff9354c6a7bae10e6edc0008a8a
cebdb965857cab73def78dcc961222e7d94a\c!microsoft-
w..anguagepack_31bf3856ad364e35_6.1.7600.16385_1bf194f31711fd1e:
(NULL!)

HKLM\COMPONENTS\CanonicalData\Catalogs\fe5742a66874a82cf706b5c014
c4d7a2d28974f344b45e69d7baadc033d6e4d0\c!microsoft-
w..anguagepack_31bf3856ad364e35_6.1.7600.16385_9c679c365b2a2bf1:
(NULL!)

HKLM\COMPONENTS\CanonicalData\Catalogs\fd79845e6f29b9702a49e65213
b44918af3a97c35947dc861281f05052770ab1\c!microsoft-w..-
deployment_31bf3856ad364e35_6.1.7600.16385_b165212581dbff4b: (NULL!)

HKLM\COMPONENTS\CanonicalData\Catalogs\fd556224aa1762cb5d9fc39e14
12b4d1e2ebfdce27d5a93997a7d869eb193930\c!microsoft-w..-
deployment_31bf3856ad364e35_8.0.7601.17514_ca43a950e5c549b1: (NULL!)

HKLM\COMPONENTS\CanonicalData\Catalogs\fcc2dbd22f9d9b461ab536b98a
67632ffd2c60c867eeb734e07d667b5a2d2076\c!microsoft-
w..anguagepack_31bf3856ad364e35_6.1.7600.16385_a4f5e9a711ce7e7c:
(NULL!)

HKLM\COMPONENTS\CanonicalData\Catalogs\fa4d29bf1d77cf86378f3bde9c1
42aac1dc5271c37a4b9583008c5f50b2840e8\c!microsoft-
w..anguagepack_31bf3856ad364e35_6.1.7601.17514_b417d7ae214f2923:
(NULL!)

HKLM\COMPONENTS\CanonicalData\Catalogs\fa3ea2ed65e7c40c2fd2088e739
6c336371affc4b52acb3e020e426ea8ab3cd8\c!microsoft-w..-
deployment_31bf3856ad364e35_6.1.7600.16385_cfe1377a190424f5: (NULL!)

HKLM\COMPONENTS\CanonicalData\Catalogs\f9b4977053d540617c18c2d221
5fd5c7d93a98be0e27ff00239cb0e3d3ea828d\c!microsoft-
w..anguagepack_31bf3856ad364e35_6.1.7600.16385_4a8e678394a0f8ca:
(NULL!)

HKLM\COMPONENTS\CanonicalData\Catalogs\f8b338712deac04c496dfe64b9
37ea1fb04e2d7b7697af312316f0e8e4e3d500\c!microsoft-w..-
deployment_31bf3856ad364e35_6.1.7601.17514_3bfc547efe9cb9c7: (NULL!)

HKLM\COMPONENTS\CanonicalData\Catalogs\f74ddb6d03076bde4dc1d8aa9a
9ed3f582f7648df5debe1788b4d6da3ec63542\c!microsoft-w..-
deployment_31bf3856ad364e35_6.1.7600.16385_3b548adcacabdfdf: (NULL!)

HKLM\COMPONENTS\CanonicalData\Catalogs\f6a4f39ce8b4730ef9f74afe486a
ed11e42285cb0588674932ab5ec7964895a6\c!microsoft-
w..anguagepack_31bf3856ad364e35_6.1.7600.16385_40f3f391bb0cc4b3:
(NULL!)

HKLM\COMPONENTS\CanonicalData\Catalogs\f323662ca70635244bacd73205
dfb69ce5a3d8f35d41609dd6cb964b1011557d\c!microsoft-w..-
deployment_31bf3856ad364e35_6.1.7601.17514_5a15a0d8332b4022: (NULL!)

HKLM\COMPONENTS\CanonicalData\Catalogs\f2700dc92d5471b12904d4d945
350a4aa4adf488c76e4a4aaa468df3ba70b74c\c!subsystem-
f..anguagepack_31bf3856ad364e35_6.1.7600.16385_ff084bdcff15a096: (NULL!)

Values added:1058

HKLM\SYSTEM\ControlSet001\services\NPF\Enum\Count: 0x00000001

HKLM\SYSTEM\ControlSet001\services\NPF\Enum\NextInstance: 0x00000001

HKLM\SYSTEM\ControlSet001\services\NPF\Type: 0x00000001

HKLM\SYSTEM\ControlSet001\services\NPF\Start: 0x00000002

HKLM\SYSTEM\ControlSet001\services\NPF\ErrorControl: 0x00000001

HKLM\SYSTEM\ControlSet001\services\NPF\ImagePath:
"system32\drivers\npf.sys"

HKLM\SYSTEM\ControlSet001\services\NPF\DisplayName: "NetGroup Packet
Filter Driver"

Values modified:58

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\GlobalAssoc
ChangedCounter: 0x00000003

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\GlobalAssoc
ChangedCounter: 0x00000006

Total changes:112475

Appendix 4

Some of the changes to the file system

Time	Process Name	PID	Operation	Path	Result	Detail
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFile	C:\Windows\Prefetch\4D4C2729BA456E70EAF9EF84E9D52940DE57.pf	NAME NOT FOUND	Desired Access: G...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFile	C:\Users\salut\Desktop\exp\binaries\binaries	SUCCESS	Desired Access: E...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFile	C:\Windows\System32\echohost.dll	SUCCESS	Desired Access: R...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	QueryBasicInformationFile	C:\Windows\System32\echohost.dll	SUCCESS	Creation Time: 14/0...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CloseFile	C:\Windows\System32\echohost.dll	SUCCESS	
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFile	C:\Windows\System32\echohost.dll	SUCCESS	Desired Access: R...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFileMapping	C:\Windows\System32\echohost.dll	FILE LOCKED WITH ONLY READERS	SyncType: SyncTy...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFileMapping	C:\Windows\System32\echohost.dll	SUCCESS	SyncType: SyncTy...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CloseFile	C:\Windows\System32\echohost.dll	SUCCESS	
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFile	C:\Windows\System32\imm32.dll	SUCCESS	Desired Access: R...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	QueryBasicInformationFile	C:\Windows\System32\imm32.dll	SUCCESS	Creation Time: 21/1...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CloseFile	C:\Windows\System32\imm32.dll	SUCCESS	
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFile	C:\Windows\System32\imm32.dll	SUCCESS	Desired Access: R...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFileMapping	C:\Windows\System32\imm32.dll	FILE LOCKED WITH ONLY READERS	SyncType: SyncTy...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	QueryStandardInformationFile	C:\Windows\System32\imm32.dll	SUCCESS	AllocationSize: 118...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFileMapping	C:\Windows\System32\imm32.dll	SUCCESS	SyncType: SyncTy...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CloseFile	C:\Windows\System32\imm32.dll	SUCCESS	
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFile	C:\Windows\System32\imm32.dll	SUCCESS	Desired Access: R...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	QueryBasicInformationFile	C:\Windows\System32\imm32.dll	SUCCESS	Creation Time: 21/1...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CloseFile	C:\Windows\System32\imm32.dll	SUCCESS	
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFile	C:\Windows\System32\imm32.dll	SUCCESS	Desired Access: R...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFileMapping	C:\Windows\System32\imm32.dll	FILE LOCKED WITH ONLY READERS	SyncType: SyncTy...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFileMapping	C:\Windows\System32\imm32.dll	SUCCESS	SyncType: SyncTy...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CloseFile	C:\Windows\System32\imm32.dll	SUCCESS	
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFile	C:\Windows\Globalization\Sorting\SortDefault.nls	SUCCESS	Desired Access: G...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFileMapping	C:\Windows\Globalization\Sorting\SortDefault.nls	FILE LOCKED WITH ONLY READERS	SyncType: SyncTy...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	QueryStandardInformationFile	C:\Windows\Globalization\Sorting\SortDefault.nls	SUCCESS	AllocationSize: 2.9...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFileMapping	C:\Windows\Globalization\Sorting\SortDefault.nls	SUCCESS	SyncType: SyncTy...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CloseFile	C:\Windows\Globalization\Sorting\SortDefault.nls	SUCCESS	
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFile	C:\Users\salut\Desktop\exp\binaries\binaries\NetApi32.dll	NAME NOT FOUND	Desired Access: R...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFile	C:\Windows\System32\netapi32.dll	SUCCESS	Desired Access: R...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	QueryBasicInformationFile	C:\Windows\System32\netapi32.dll	SUCCESS	Creation Time: 21/1...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CloseFile	C:\Windows\System32\netapi32.dll	SUCCESS	
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFile	C:\Windows\System32\netapi32.dll	SUCCESS	Desired Access: R...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFileMapping	C:\Windows\System32\netapi32.dll	FILE LOCKED WITH ONLY READERS	SyncType: SyncTy...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFileMapping	C:\Windows\System32\netapi32.dll	SUCCESS	SyncType: SyncTy...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CloseFile	C:\Windows\System32\netapi32.dll	SUCCESS	
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFile	C:\Users\salut\Desktop\exp\binaries\binaries\netutils.dll	NAME NOT FOUND	Desired Access: R...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFile	C:\Windows\System32\netutils.dll	SUCCESS	Desired Access: R...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	QueryBasicInformationFile	C:\Windows\System32\netutils.dll	SUCCESS	Creation Time: 21/1...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CloseFile	C:\Windows\System32\netutils.dll	SUCCESS	
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFile	C:\Windows\System32\netutils.dll	SUCCESS	Desired Access: R...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFileMapping	C:\Windows\System32\netutils.dll	FILE LOCKED WITH ONLY READERS	SyncType: SyncTy...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFileMapping	C:\Windows\System32\netutils.dll	SUCCESS	SyncType: SyncTy...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CloseFile	C:\Windows\System32\netutils.dll	SUCCESS	
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFile	C:\Users\salut\Desktop\exp\binaries\binaries\nvcl.dll	NAME NOT FOUND	Desired Access: R...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFile	C:\Windows\System32\nvcl.dll	SUCCESS	Desired Access: R...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	QueryBasicInformationFile	C:\Windows\System32\nvcl.dll	SUCCESS	Creation Time: 21/1...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CloseFile	C:\Windows\System32\nvcl.dll	SUCCESS	
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFile	C:\Windows\System32\nvcl.dll	SUCCESS	Desired Access: R...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFileMapping	C:\Windows\System32\nvcl.dll	FILE LOCKED WITH ONLY READERS	SyncType: SyncTy...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CreateFileMapping	C:\Windows\System32\nvcl.dll	SUCCESS	SyncType: SyncTy...
10:45...	444c2729baa5670ea9ef04e955d3d.exe	3960	CloseFile	C:\Windows\System32\nvcl.dll	SUCCESS	

Appendix 5

Some of the changes to the Registry

Time ...	Process Name	PID	Operation	Path	Result
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager	REPARSE
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegQueryValue	HKLM\System\CurrentControlSet\Control\Session Manager\CWD\Legal\DLLSearch	NAME NOT FOUND
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegCloseKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\Control\Terminal Server	REPARSE
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\Control\Terminal Server	SUCCESS
10:45...	C:\Users\sultan\Desktop\leap\binaries\binaries\444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\Control\Terminal Server\TSAppCompat	NAME NOT FOUND
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegQueryValue	HKLM\System\CurrentControlSet\Control\Terminal Server\TSUserEnabled	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegCloseKey	HKLM\System\CurrentControlSet\Control\Terminal Server	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\Control\SafeBoot\Option	REPARSE
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\Control\SafeBoot\Option	NAME NOT FOUND
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\Control\Sip\GP_DLL	REPARSE
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\Control\Sip\GP_DLL	NAME NOT FOUND
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegQueryValue	HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers\TransparentEnabled	NAME NOT FOUND
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegCloseKey	HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers	NAME NOT FOUND
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\Control\Nls\Sorting\Versions	REPARSE
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\Control\Nls\Sorting\Versions	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegQueryValue	HKLM\System\CurrentControlSet\Control\Nls\Sorting\Versions\Default	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager	REPARSE
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegQueryValue	HKLM\System\CurrentControlSet\Control\Session Manager\SafeDllSearchMode	NAME NOT FOUND
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\Control\Error Message Instrument\	REPARSE
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\Control\Error Message Instrument	NAME NOT FOUND
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\GRE_Initialize	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegQueryValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\GRE_Initialize\DisableMetaFiles	NAME NOT FOUND
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegCloseKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\GRE_Initialize	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Compatibility32	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegQueryValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Compatibility32\444c:272b:0aa56e70...	NAME NOT FOUND
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegCloseKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Compatibility32	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\IME Compatibility	NAME NOT FOUND
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegQueryValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows\LoadAppInit_DLLs	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegCloseKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\Control\Terminal Server	REPARSE
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\Control\Terminal Server	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegQueryValue	HKLM\System\CurrentControlSet\Control\Terminal Server\TSAppCompat	NAME NOT FOUND
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegQueryValue	HKLM\System\CurrentControlSet\Control\Terminal Server\TSUserEnabled	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegCloseKey	HKLM\System\CurrentControlSet\Control\Terminal Server	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Diagnostics	NAME NOT FOUND
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\Software\Policies\Microsoft\SQMClient\Windows	NAME NOT FOUND
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\Software\Microsoft\SQMClient\Windows	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegQueryValue	HKLM\SOFTWARE\Microsoft\SQMClient\Windows\CEIPEnable	NAME NOT FOUND
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegCloseKey	HKLM\SOFTWARE\Microsoft\SQMClient\Windows	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\Services\WinSock2\Parameters	REPARSE
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\Services\WinSock2\Parameters	ACCESS DENIED
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\Services\WinSock2\Parameters	REPARSE
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\Services\WinSock2\Parameters	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegQueryValue	HKLM\System\CurrentControlSet\services\WinSock2\Parameters\WinSock_Registry_Version	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegQueryValue	HKLM\System\CurrentControlSet\services\WinSock2\Parameters\WinSock_Registry_Version	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\Control\Nls\CustomLocale	REPARSE
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\Control\Nls\CustomLocale	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegQueryValue	HKLM\System\CurrentControlSet\Control\Nls\CustomLocale\en-US	NAME NOT FOUND
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegCloseKey	HKLM\System\CurrentControlSet\Control\Nls\CustomLocale	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\Control\Nls\ExtendedLocale	REPARSE
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\Control\Nls\ExtendedLocale	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegQueryValue	HKLM\System\CurrentControlSet\Control\Nls\ExtendedLocale\en-US	NAME NOT FOUND
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegCloseKey	HKLM\System\CurrentControlSet\Control\Nls\ExtendedLocale	SUCCESS
10:45...	444c:272b:0aa56e70eaf9ef04e9f5d3d.exe	3960	RegOpenKey	HKLM\System\CurrentControlSet\services\WinSock2\Parameters\Applid_Catalog	SUCCESS

Appendix 6

Some of threads – effected

Time	Process Name	PID	Operation	Path	Result	Detail
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Process Start		SUCCESS	Parent PID: 1496, ...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 2796
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Users\cultur\Desktop\exp\binaries\binaries\404c2729baa5fe70ea9ef04e9b553d.exe	SUCCESS	Image Base: 0x400...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\ntdll.dll	SUCCESS	Image Base: 0x77a...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\kernel32.dll	SUCCESS	Image Base: 0x77e...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\kernelbase.dll	SUCCESS	Image Base: 0x75e...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\user32.dll	SUCCESS	Image Base: 0x77c...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\gdi32.dll	SUCCESS	Image Base: 0x781...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\gdi32.dll	SUCCESS	Image Base: 0x79...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\ole32.dll	SUCCESS	Image Base: 0x782...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\advapi32.dll	SUCCESS	Image Base: 0x781...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\sechost.dll	SUCCESS	Image Base: 0x784...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\port4.dll	SUCCESS	Image Base: 0x775...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\ws2_32.dll	SUCCESS	Image Base: 0x784...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\rsa.dll	SUCCESS	Image Base: 0x79F...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\imm32.dll	SUCCESS	Image Base: 0x774...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\oleaut32.dll	SUCCESS	Image Base: 0x774...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\ole32.dll	SUCCESS	Image Base: 0x743...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\ntutils.dll	SUCCESS	Image Base: 0x743...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\ole32.dll	SUCCESS	Image Base: 0x758...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\ole32.dll	SUCCESS	Image Base: 0x743...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\ole32.dll	SUCCESS	Image Base: 0x726...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\ole32.dll	SUCCESS	Image Base: 0x743...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\ole32.dll	SUCCESS	Image Base: 0x75b...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\ole32.dll	SUCCESS	Image Base: 0x755...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\IPHLPAPI.DLL	SUCCESS	Image Base: 0x73f...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\winnsi.dll	SUCCESS	Image Base: 0x73e...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 3140
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\dhcpssvc6.dll	SUCCESS	Image Base: 0x73d...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\dhcpssvc6.dll	SUCCESS	Image Base: 0x73d...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\dhcpssvc6.dll	SUCCESS	Image Base: 0x757...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Load Image	C:\Windows\System32\WSHCTCP.DLL	SUCCESS	Image Base: 0x751...
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 3216
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 1052
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 1928
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 3740
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 3476
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 2880
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 1952
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 3952
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 3292
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 1404
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 1988
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 1628
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 3676
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 908
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 2160
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 3112
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 816
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 1348
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 2032
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 1668
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 1796
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 2044
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 3564
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 3828
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 3148
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 3356
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 3792
10:45...	444c2729baa5fe70ea9ef04e9b553d.exe	3980	Thread Create		SUCCESS	Thread ID: 2712

Appendix 7

Network Report by Anibus for the 251616a9205e376778b261330b11da9b IRC
bot

1.a) - Network Activity

TCP Scans:

48 IPs on Port 445

24.31.0.0/16

48 IPs on Port 139

24.31.0.0/16

Unknown TCP Traffic:

From ANUBIS:1065 to 193.166.255.170:80

State: Normal establishment and termination - Transferred outbound Bytes: 71 -
Transferred inbound Bytes: 0

From ANUBIS:1179 to 24.31.55.181:445

State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0

Data sent:

0000 00a8 ff53 4d42 7200 0000 0008 0140SMBr.....@

0000 0000 0000 0000 0000 0000 0000 a804

0000 2000 0085 0002 5043 204e 4554 574fPC NETWO

524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..

4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWORK

4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO

4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.

024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.

3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1

0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.

024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.

From ANUBIS:1189 to 24.31.20.110:445

State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0

Data sent:

0000 00a8 ff53 4d42 7200 0000 0008 0140SMBr.....@

0000 0000 0000 0000 0000 0000 0000 a804

0000 1000 0085 0002 5043 204e 4554 574fPC NETWO

524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..

4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWORK

4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO

4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.

024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.

3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1

0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.

024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.

From ANUBIS:1221 to 24.31.19.200:445

State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0

Data sent:

```

0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMBr.....@
0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 3000 0085 0002 5043 204e 4554 574f ..0.....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1151 to 24.31.29.52:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Analysis Report for 251616a9205e376778b261330b11da9b - submitted on
10/30/13, 15:38:16 UTC
Unknown TCP Traffic:
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMBr.....@
0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 2000 0085 0002 5043 204e 4554 574f .. .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1184 to 24.31.19.30:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMBr.....@
0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 1000 0085 0002 5043 204e 4554 574f .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1196 to 24.31.118.86:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMBr.....@
0000 0000 0000 0000 0000 0000 0000 a804 .....

```

```

0000 1000 0085 0002 5043 204e 4554 574f .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1162 to 24.31.190.164:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMB.....@
0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 2000 0085 0002 5043 204e 4554 574f .. .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1164 to 24.31.181.140:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMB.....@
0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 2000 0085 0002 5043 204e 4554 574f .. .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
Analysis Report for 251616a9205e376778b261330b11da9b - submitted on
10/30/13, 15:38:16 UTC
Unknown TCP Traffic:
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1190 to 24.31.229.203:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMB.....@
0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 1000 0085 0002 5043 204e 4554 574f .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..

```

```

4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWORK
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1180 to 24.31.246.162:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMB.....@
0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 2000 0085 0002 5043 204e 4554 574f .. .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWORK
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1193 to 24.31.155.221:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMB.....@
0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 1000 0085 0002 5043 204e 4554 574f .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWORK
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1202 to 24.31.164.186:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMB.....@
0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 1000 0085 0002 5043 204e 4554 574f .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWORK
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1

```



```

0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1185 to 24.31.61.83:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMBr.....@
0000 0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 2000 0085 0002 5043 204e 4554 574f .. .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
Analysis Report for 251616a9205e376778b261330b11da9b - submitted on
10/30/13, 15:38:16 UTC
Unknown TCP Traffic:
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWORK
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1203 to 24.31.249.50:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMBr.....@
0000 0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 1000 0085 0002 5043 204e 4554 574f .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWORK
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1209 to 24.31.211.10:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMBr.....@
0000 0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 1000 0085 0002 5043 204e 4554 574f .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWORK
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.

```

From ANUBIS:1195 to 24.31.179.36:445
State: Connection established, not terminated - Transferred outbound Bytes: 172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140SMB.....@
0000 0000 0000 0000 0000 0000 0000 a804
0000 1000 0085 0002 5043 204e 4554 574fPC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.

From ANUBIS:1187 to 24.31.146.102:445
State: Connection established, not terminated - Transferred outbound Bytes: 172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140SMB.....@
0000 0000 0000 0000 0000 0000 0000 a804
0000 1000 0085 0002 5043 204e 4554 574fPC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.

From ANUBIS:1200 to 24.31.250.155:445
State: Connection established, not terminated - Transferred outbound Bytes: 172 - Transferred inbound Bytes: 0
Data sent:
Analysis Report for 251616a9205e376778b261330b11da9b - submitted on 10/30/13, 15:38:16 UTC
Unknown TCP Traffic:
0000 00a8 ff53 4d42 7200 0000 0008 0140SMB.....@
0000 0000 0000 0000 0000 0000 0000 a804
0000 1000 0085 0002 5043 204e 4554 574fPC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.

From ANUBIS:1205 to 24.31.106.69:445
State: Connection established, not terminated - Transferred outbound Bytes:

```

172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMBr.....@
0000 0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 1000 0085 0002 5043 204e 4554 574f .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1211 to 24.31.141.125:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMBr.....@
0000 0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 1000 0085 0002 5043 204e 4554 574f .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1178 to 24.31.89.248:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMBr.....@
0000 0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 2000 0085 0002 5043 204e 4554 574f .. .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1149 to 24.31.63.194:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMBr.....@
0000 0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 2000 0085 0002 5043 204e 4554 574f .. .....PC NETWO

```

524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWORK
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
Analysis Report for 251616a9205e376778b261330b11da9b - submitted on
10/30/13, 15:38:16 UTC
Unknown TCP Traffic:
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1176 to 24.31.83.9:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140SMBr.....@
0000 0000 0000 0000 0000 0000 0000 a804
0000 2000 0085 0002 5043 204e 4554 574fPC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWORK
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1177 to 24.31.23.250:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140SMBr.....@
0000 0000 0000 0000 0000 0000 0000 a804
0000 2000 0085 0002 5043 204e 4554 574fPC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWORK
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1191 to 24.31.71.63:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140SMBr.....@
0000 0000 0000 0000 0000 0000 0000 a804
0000 1000 0085 0002 5043 204e 4554 574fPC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWORK

```

4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1210 to 24.31.15.7:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMB.....@
0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 1000 0085 0002 5043 204e 4554 574f .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWORK
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1182 to 24.31.29.164:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMB.....@
0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 2000 0085 0002 5043 204e 4554 574f .. .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWORK
Analysis Report for 251616a9205e376778b261330b11da9b - submitted on
10/30/13, 15:38:16 UTC
Unknown TCP Traffic:
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1208 to 24.31.146.255:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMB.....@
0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 1000 0085 0002 5043 204e 4554 574f .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWORK
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.

```

```

024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1168 to 24.31.21.53:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMB.....@
0000 0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 2000 0085 0002 5043 204e 4554 574f .. .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWORK
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1186 to 24.31.236.229:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMB.....@
0000 0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 2000 0085 0002 5043 204e 4554 574f .. .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWORK
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1201 to 24.31.59.224:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMB.....@
0000 0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 1000 0085 0002 5043 204e 4554 574f .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWORK
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1181 to 24.31.4.202:445

```

State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0

Data sent:

Analysis Report for 251616a9205e376778b261330b11da9b - submitted on
10/30/13, 15:38:16 UTC

Unknown TCP Traffic:

0000 00a8 ff53 4d42 7200 0000 0008 0140SMBr.....@
0000 0000 0000 0000 0000 0000 0000 a804
0000 1000 0085 0002 5043 204e 4554 574fPC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.

From ANUBIS:1206 to 24.31.97.12:445

State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0

Data sent:

0000 00a8 ff53 4d42 7200 0000 0008 0140SMBr.....@
0000 0000 0000 0000 0000 0000 0000 a804
0000 1000 0085 0002 5043 204e 4554 574fPC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.

From ANUBIS:1183 to 24.31.166.55:445

State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0

Data sent:

0000 00a8 ff53 4d42 7200 0000 0008 0140SMBr.....@
0000 0000 0000 0000 0000 0000 0000 a804
0000 2000 0085 0002 5043 204e 4554 574fPC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.

From ANUBIS:1199 to 24.31.96.91:445

State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0

Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140SMBr.....@
0000 0000 0000 0000 0000 0000 0000 a804
0000 1000 0085 0002 5043 204e 4554 574fPC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWORK
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0..
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1..
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0..
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1197 to 24.31.178.166:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0

Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140SMBr.....@
0000 0000 0000 0000 0000 0000 0000 a804
0000 3000 0085 0002 5043 204e 4554 574f ..0.....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWORK
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0..
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1..
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0..
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
Analysis Report for 251616a9205e376778b261330b11da9b - submitted on
10/30/13, 15:38:16 UTC

Unknown TCP Traffic:
From ANUBIS:1192 to 24.31.10.162:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0

Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140SMBr.....@
0000 0000 0000 0000 0000 0000 0000 a804
0000 1000 0085 0002 5043 204e 4554 574fPC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWORK
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0..
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1..
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0..
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1212 to 24.31.240.216:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0

Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140SMBr.....@


```

0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 1000 0085 0002 5043 204e 4554 574f .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1166 to 24.31.93.208:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMB.....@
0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 2000 0085 0002 5043 204e 4554 574f .. .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1188 to 24.31.75.237:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMB.....@
0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 1000 0085 0002 5043 204e 4554 574f .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1170 to 24.31.85.178:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMB.....@
0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 2000 0085 0002 5043 204e 4554 574f .. .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO

```

4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
Analysis Report for 251616a9205e376778b261330b11da9b - submitted on
10/30/13, 15:38:16 UTC
Unknown TCP Traffic:
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1194 to 24.31.20.130:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140SMBr.....@
0000 0000 0000 0000 0000 0000 0000 0000 a804
0000 1000 0085 0002 5043 204e 4554 574fPC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1216 to 24.31.180.201:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140SMBr.....@
0000 0000 0000 0000 0000 0000 0000 0000 a804
0000 1000 0085 0002 5043 204e 4554 574fPC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1219 to 24.31.159.162:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140SMBr.....@
0000 0000 0000 0000 0000 0000 0000 0000 a804
0000 3000 0085 0002 5043 204e 4554 574f ..0.....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.

```

3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1198 to 24.31.25.217:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMBr.....@
0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 1000 0085 0002 5043 204e 4554 574f .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1217 to 24.31.64.157:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMBr.....@
0000 0000 0000 0000 0000 0000 0000 a804 .....
Analysis Report for 251616a9205e376778b261330b11da9b - submitted on
10/30/13, 15:38:16 UTC
Unknown TCP Traffic:
0000 1000 0085 0002 5043 204e 4554 574f .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1204 to 24.31.86.178:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMBr.....@
0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 1000 0085 0002 5043 204e 4554 574f .....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.

```

```

024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
From ANUBIS:1215 to 24.31.35.95:445
State: Connection established, not terminated - Transferred outbound Bytes:
172 - Transferred inbound Bytes: 0
Data sent:
0000 00a8 ff53 4d42 7200 0000 0008 0140 .....SMBr.....@
0000 0000 0000 0000 0000 0000 0000 a804 .....
0000 3000 0085 0002 5043 204e 4554 574f ..0.....PC NETWO
524b 2050 524f 4752 414d 2031 2e30 0002 RK PROGRAM 1.0..
4d49 4352 4f53 4f46 5420 4e45 5457 4f52 MICROSOFT NETWOR
4b53 2031 2e30 3300 024d 4943 524f 534f KS 1.03..MICROSO
4654 204e 4554 574f 524b 5320 332e 3000 FT NETWORKS 3.0.
024c 414e 4d41 4e31 2e30 0002 4c4d 312e .LANMAN1.0..LM1.
3258 3030 3200 024c 414e 4d41 4e32 2e31 2X002..LANMAN2.1
0002 4e54 204c 414e 4d41 4e20 312e 3000 ..NT LANMAN 1.0.
024e 5420 4c4d 2030 2e31 3200 .NT LM 0.12.
TCP Connection Attempts:
From ANUBIS:1029 to 24.31.159.162:139
From ANUBIS:1030 to 24.31.93.208:139
From ANUBIS:1031 to 24.31.159.162:139
From ANUBIS:1033 to 24.31.190.164:139
From ANUBIS:1034 to 24.31.83.9:139
From ANUBIS:1035 to 24.31.246.162:139
From ANUBIS:1037 to 24.31.164.186:139
From ANUBIS:1038 to 24.31.4.202:139
From ANUBIS:1039 to 24.31.29.164:139
From ANUBIS:1040 to 24.31.155.221:139
From ANUBIS:1042 to 24.31.61.83:139
From ANUBIS:1032 to 24.31.19.30:139
From ANUBIS:1043 to 24.31.89.248:139
From ANUBIS:1041 to 24.31.23.250:139
From ANUBIS:1036 to 24.31.93.208:139
From ANUBIS:1044 to 24.31.155.221:139
From ANUBIS:1045 to 24.31.83.9:139
From ANUBIS:1048 to 24.31.190.164:139
From ANUBIS:1051 to 24.31.4.202:139
From ANUBIS:1052 to 24.31.29.164:139
From ANUBIS:1050 to 24.31.164.186:139
From ANUBIS:1046 to 24.31.250.155:139
From ANUBIS:1053 to 24.31.61.83:139
Analysis Report for 251616a9205e376778b261330b11da9b - submitted on
10/30/13, 15:38:16 UTC
TCP Connection Attempts:
From ANUBIS:1054 to 24.31.19.30:139
From ANUBIS:1057 to 24.31.23.250:139
From ANUBIS:1056 to 24.31.89.248:139
From ANUBIS:1049 to 24.31.246.162:139
From ANUBIS:1064 to 24.31.250.155:139
From ANUBIS:1069 to 24.31.236.229:139
From ANUBIS:1070 to 24.31.10.162:139

```

From ANUBIS:1066 to 24.31.229.203:139
From ANUBIS:1075 to 24.31.10.162:139
From ANUBIS:1076 to 24.31.236.229:139
From ANUBIS:1077 to 24.31.229.203:139
From ANUBIS:1080 to 24.31.35.95:139
From ANUBIS:1078 to 24.31.75.237:139
From ANUBIS:1083 to 24.31.25.217:139
From ANUBIS:1086 to 24.31.75.237:139
From ANUBIS:1084 to 24.31.25.217:139
From ANUBIS:1085 to 24.31.35.95:139
From ANUBIS:1090 to 24.31.180.201:139
From ANUBIS:1089 to 24.31.64.157:139
From ANUBIS:1091 to 24.31.180.201:139
From ANUBIS:1047 to 24.31.159.162:445
From ANUBIS:1093 to 24.31.64.157:139
From ANUBIS:1094 to 24.31.21.53:139
From ANUBIS:1059 to 24.31.155.221:445
From ANUBIS:1097 to 24.31.19.200:139
From ANUBIS:1058 to 24.31.93.208:445
From ANUBIS:1055 to 24.31.190.164:445
From ANUBIS:1100 to 24.31.29.52:139
From ANUBIS:1098 to 24.31.63.194:139
From ANUBIS:1099 to 24.31.146.102:139
From ANUBIS:1101 to 24.31.181.140:139
From ANUBIS:1096 to 24.31.159.162:139
From ANUBIS:1061 to 24.31.83.9:445
From ANUBIS:1062 to 24.31.29.164:445
From ANUBIS:1063 to 24.31.164.186:445
From ANUBIS:1104 to 24.31.166.55:139
From ANUBIS:1105 to 24.31.85.178:139
From ANUBIS:1102 to 24.31.55.181:139
From ANUBIS:1060 to 24.31.4.202:445
From ANUBIS:1108 to 24.31.86.178:139
From ANUBIS:1109 to 24.31.178.166:139
From ANUBIS:1067 to 24.31.61.83:445
From ANUBIS:1113 to 24.31.211.10:139
From ANUBIS:1068 to 24.31.19.30:445
From ANUBIS:1112 to 24.31.71.63:139
From ANUBIS:1071 to 24.31.23.250:445
From ANUBIS:1107 to 24.31.29.52:139
From ANUBIS:1072 to 24.31.89.248:445
From ANUBIS:1110 to 24.31.21.53:139
From ANUBIS:1106 to 24.31.20.130:139
From ANUBIS:1073 to 24.31.246.162:445
From ANUBIS:1116 to 24.31.96.91:139
From ANUBIS:1074 to 24.31.250.155:445
From ANUBIS:1111 to 24.31.59.224:139
From ANUBIS:1115 to 24.31.19.200:139
From ANUBIS:1122 to 24.31.249.50:139
From ANUBIS:1119 to 24.31.179.36:139

