# All About Malwares (Malicious Codes)

**Hossein Rouhani Zeidanloo, S. Farzaneh Tabatabaei, Payam Vahdani Amoli and Atefeh Tajpour**
Faculty of Computer Science and Information System,
University of Technology Malaysia(UTM) , Kuala Lumpur, Malaysia

**Abstract -** *Malware, short term for malicious software, is a software which is developed to penetrate computers in a network without the user's permission or notification. Malware is a common term for a variety type of malicious software. In general, Malwares include Worm, Botnet, virus, Trojan horse, Backdoor, Rootkit, Logic bomb, Rabbit and Spyware. Despite many works that have been done in the area of Malware, still there is not any distinct classification which differentiates different kind of Malwares and explains each of them thoroughly. In this paper, we define each of them in detail and emphasize their differences. We also conclude our studies in this area with providing a diagram which gives a comprehensive overview about Malware. Among the diverse forms of malware, botnet and worm are the most widespread and serious threat which occur commonly in today's cyber attacks. Therefore, we concentrate more on them and their communication topologies.*

**Keywords:** Malware; Worm; Botnet; Virus

## 1 Introduction

Malware is a common term for all types of malicious software, which in the area of computer security means: "Software which is used with the intention of violating a computer system's security policy". There are many other definitions for Malware, but all of them have some area in common in which Malware is malicious codes that has the potential to harm the machine or network on which it executes. [1]

The term "software" here is a broad area which malicious software may use of executable code, scripts and etc. Actually the machine that its security policy is violated is considered as "target or victim" for malware. It is better we use the term "originator" of the malware to signify the issue who initially launched the malware with the intention of attacking one or more targets. Depending on the kind of malware, the set of targets may or may not be explicitly known to the originator.

Malwares include Worm, Botnet, virus, Trojan horse, Backdoor, Rootkit, Logic bomb, Rabbit and Spyware. Despite many works have been done in the area of Malware, but still there is not any distinct classification which differentiates different kind of Malware and explains each of them carefully. In this paper, we try to define each of these Malware carefully and provide a complete reference for those researchers who want work in the area of Malware. The remainder of the paper is organized as follows: Section II describes Malwares in details; Section III describes our Classification for Malwares. The paper concludes in Section IV.

## 2 Malwares

Malware is usually classify into a number of type, based on the way in which it is introduced into the target system and the sort of policy breach which it is intended to cause. The traditional classification was introduced by Peter Denning in the late 1980s [2, 3].

### 2.1 Worms

Worm is one of the malicious software which has independent structure and distribute from one computer to another by replicating automatically copies of itself via a network, without the use of infected files or human action. It means that worms have self-replication and self-contained properties. Self-replication means that it has the ability to copy itself and self-contained means that worm has the ability to execute without the need to attach to another program. [4]

The points that distinguish worm from virus are: (i) Its capability to replicate copies of itself automatically without any human action, (ii) unlike a virus, worm does not need to attach itself to an existing program. As an example, we can say that when a worm installed in computer system, it could send out thousands of its copies to everyone listed in computer email address book.

Worm could have very harmful effect on systems in the network, such as could consume too much system memory or system processor (CPU) and cause many applications to stop responding. [5,6] Worms may be based on executable code, interpreted code, scripts, macros, etc.

A worm typically consists of three parts:

- *Identifier:* Code used to identify possible targets, i.e. other hosts which it can try to infect.
- *Transmitter:* Code used to transfer the worm to the targets.
- *Payload:* Code to be executed on the target. The payload is optional, and it may or may not have a damaging effect on the target.

#### 2.1.1 Identifying the Targets

Identifying and searching the targets could be categorized into two groups: (i) information found locally on the host which worm installed (ii) Organized explore of the network.

Local information usually could be finding in configuration files which contain addresses of other computers to be contacted for different purposes. As an example, worms which spread via e-mail look in personal e-mail address books or search through text files which might contain e-mail addresses. Organized searching of network is usually based on port scanning to enable worms to discover the ports that are open and can be contacted. The searching techniques for finding open ports in the network produces characteristic network activity and consequently man methods have been designed to detect these activities and stop the spread of such worms.

### 2.1.2 Transmitting the Worm

As soon as appropriate possible targets have been discovered, the worm will attempt to use its special dissemination technique to send itself to these new hosts and get its code executed on them. Actually each specific kind of worm uses different method for its propagation in the network but in general we can say that all worms have some common characteristics for their propagation which we categorize into four steps. (i) infect one system (ii) find additional systems in the system that already infected to target and infect them. It could use IP addresses or email addresses that exist in the infected system (iii) target those additional systems that found and try to transmit worm to them. Transmitting of worm could happen via email, web clients, network file system and many other ways. (iv) execution of malicious code on the infected systems. It can be possible via user intervention, directly from command-line, web clients and many other ways.

### 2.1.3 Payload

Worm itself is not dangerous because it is just a carrier and move around. The payload of worm is the part which has malicious program and could harm the computer systems in the network. However there are some cases that worms without payload still have malicious effects. A good example of that was W32/Slammer worm which by spreading across the network used lots of network resources and cause Denial of service.

Some worms are just developed to evaluate how worms can be distribute, or actually have a useful function. One of the very first worms was developed at Xerox Palo Alto Research Center in the early 1980s in order to distribute parts of large calculations among workstations at which nobody was currently working [7].

Worms with a malicious payload can have approximately any consequences on the target hosts. Some well-known examples are:

a) To abuse the targets in order to cause a Distributed DoS attack on a selected system.
b) Website damage on the targets, which are chosen to be web servers.
c) Installation of a keylogger to track the user's input, typically in order to gain passwords, credit card numbers or other confidential information, and to transmit these to a site chosen by the originator of the worm.
d) Installation of a backdoor, providing the originator with access to the target host.

## 2.2 Botnet

Nowadays, the most serious manifestation of advanced malware is Botnet. To make distinction between Botnet and other kinds of malware, we have to comprehend the concept of Botnet. For a better understanding of Botnet, we have to know two terms first, Bot and BotMaster and then we can properly define Botnet.

*Bot* – Bot is actually short for robot which is also called as Zombie. It is a new type of malware [8] installed into a compromised computer which can be controlled remotely by BotMaster for executing some orders through the received commands. After the Bot code has been installed into the compromised computers, the computer becomes a Bot or Zombie [9]. Contrary to existing malware such as virus and worm which their main activities focus on attacking the infecting host, bots can receive commands from BotMaster and are used in distributed attack platform.

*BotMaster* – BotMaster is also known as BotHerder, is a person or a group of person which control remote Bots.

*Botnets*- Botnets are networks consisting of large number of Bots. Botnets are created by the BotMaster to setup a private communication infrastructure which can be used for malicious activities such as Distributed Denial-of-Service (DDoS), sending large amount of SPAM or phishing mails, and other nefarious purpose [10, 11, 12]. Bots infect a person's computer in many ways.

Bots usually disseminate themselves across the Internet by looking for vulnerable and unprotected computers to infect. When they find an unprotected computer, they infect it and then send a report to the BotMaster. The Bot stay hidden until they are announced by their BotMaster to perform an attack or task. Other ways in which attackers use to infect a computer in the Internet with Bot include sending email and using malicious websites, but common way is searching the Internet to look for vulnerable and unprotected computers [13]. Based on our understanding, we could say that the activities associated with Botnet can be classified into three parts: (1) *Searching* – searching for vulnerable and unprotected computers. (2) *Dissemination* – the Bot code is distributed to the computers (targets), so the targets become Bots. (3) *sign-on* – the Bots connect to BotMaster and become ready to receive command and control traffic.

The main difference between Botnet and other kind of malwares is the existence of Command-and-Control (C&C) infrastructure. The C&C allows Bots to receive commands and malicious capabilities, as devoted by BotMaster.

BotMaster must ensure that their C&C infrastructure is sufficiently robust to manage thousands of distributed Bots across the globe, as well as resisting any attempts to shutdown the Botnets. However, detection and mitigation techniques against Botnets have been increased [14, 15, 16, 17, 18].

Recently, attackers are also continually improving their approaches to protect their Botnets. The first generation of Botnets utilized the IRC (Internet Relay Chat) channels as their Common-and-Control (C&C) centers. The centralized C&C mechanism of such Botnet has made them vulnerable to being detected and disabled. Therefore, new generation of Botnet which can hide their C&C communication have emerged, Peer-to-Peer (P2P) based Botnets. The P2P Botnets do not suffer from a single point of failure, because they do not have centralized C&C servers [19]. Attackers have accordingly developed a range of strategies and techniques to protect their C&C infrastructure. Therefore, considering the C&C function gives better understanding of Botnet and help defenders to design proper detection or mitigation techniques. According to the C&C channel we categorize Botnets into two different topologies: a) Centralized; b) Decentralized.

### 2.2.1.1 Centralized Model

The oldest type of topology is the centralized model. In this model, one central point is responsible for exchanging commands and data between the BotMaster and Bots. Many well-known Bots, such as AgoBot, SDBot, Zotob and RBot used this model. In this model, BotMaster chooses a host (usually high bandwidth computer) to be the central point (Command-and-Control) server of all the Bots. The C&C server runs certain network services such as IRC or HTTP. The main advantage of this model is small message latency which cause BotMaster easily arranges Botnet and launch attacks.

Since all connections happen through the C&C server, therefore, the C&C is a critical point in this model. In other words, C&C server is the weak point in this model. If somebody manages to discover and eliminates the C&C server, the entire Botnet will be worthless and ineffective. Thus, it becomes the main drawback of this model. A lot of modern centralized Botnets employed a list of IP addresses of alternative C&C servers, which will be used in case a C&C server discovered and has been taken offline.

Since IRC and HTTP are two common protocols that C&C server uses for communication, we consider Botnets in this model based on IRC and HTTP. Figure 1 shows the basic communication architecture for a Centralized model. There are two central points that forward commands and data between the BotMaster and his Bots.

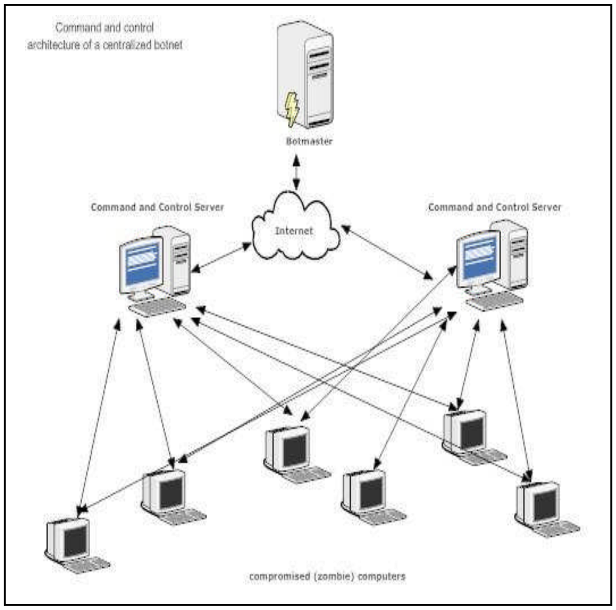
Command and control
architecture of a centralized botnet
Botmaster
Command and Control Server
Internet
Command and Control Server
compromised (zombie) computers

Figure 1. Command and control architecture of a Centralized model

#### 2.2.1.1.1 Botnets based on IRC

The IRC is a form of real-time Internet text messaging or synchronous conferencing [20]. The protocol is based on the Client-Server model, which can be used on many computers in distributed networks. Some advantages which made IRC protocol widely being used in remote communication for Botnets are: (i) low latency communication; (ii) anonymous real-time communication; (iii) ability of Group (many-to-many) and Private (one-to-one) communication; (iv) simple to setup and (v) simple commands. The basic commands are connect to servers, join channels and post messages in the channels; (vi) very flexibility in communication. Therefore IRC protocol, as shown in Figure 2 is still the most popular protocol being used in Botnet communication [12].

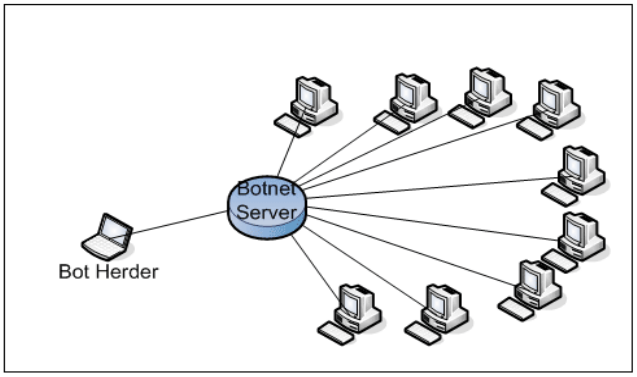

Figure 2. IRC based Botnet

#### 2.2.1.1.2 Botnet based on HTTP

HTTP protocol is another popular protocol used by Botnets. Since IRC protocol within Botnets became well-known, more internet security researchers gave attention to monitoring IRC traffic to detect Botnet. Consequently, attackers started to use HTTP protocol as a Command-and-Control communication channel to make Botnets become more difficult to detect. The main advantage of using the HTTP protocol is hiding Botnets traffics in normal web traffics, so it can easily bypasses firewalls with port-based filtering mechanisms and avoid IDS detection. Usually firewalls block incoming/outgoing traffic to unwanted ports, which often include the IRC port. There are some known Bots using the HTTP protocol, such as Bobax [21], ClickBot [22] and Rustock [23].

### 2.2.1.2 Decentralized

Due to major disadvantage of Centralized model– Central Command-and-Control(C&C)–attackers started to build alternative Botnet communication system that is much harder to discover and to destroy. Hence, they decided to find a model in which the communication system does not heavily depending on few selected servers and even discovering and destroying a number of Bots.

As a result, attackers exploit the idea of Peer-to-Peer (P2P) communication as a Command-and-Control (C&C) pattern which is more resilient to failure in the network. The P2P based C&C model will be used dramatically in Botnets in the near future, and definitely Botnets that use P2P based C&C model impose much bigger challenge for defense of networks. Since P2P based communication is more robust than Centralized C&C communication, more Botnets will move to use P2P protocol for their communication.

In the P2P model, as shown in Fig. 3, there is no Centralized point for communication. Each Bot keeps some connections to the other Bots of the Botnet. Bots act as both Clients and servers. A new Bot must know some addresses of the Botnet to connect there. If Bots in the Botnet are taken offline, the Botnet can still continue to operate under the control of BotMaster.

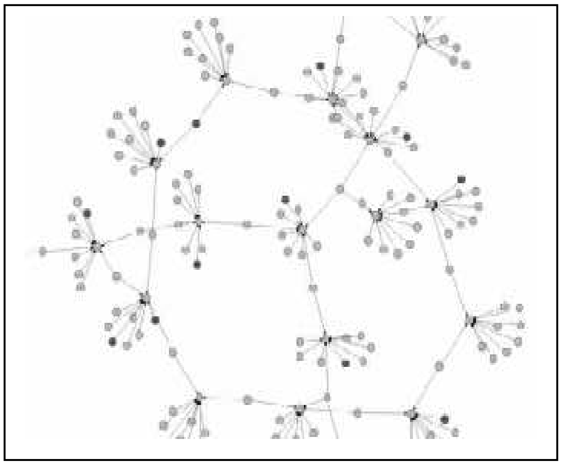

Figure 3. Example of Peer-to-peer Botnet Architecture

The first known Botnet which utilized P2P model was *Slapper* worm that appeared in 2003[24]. Other famous P2P Botnets include Sinit [25], Phatbot [26], Nugache in 2005[27], Spamthru in 2006 [28] and Storm worm in 2007.

P2P Botnets aim at removing or hiding the central point of failure which is the main weakness and vulnerability of Centralized model.

Some P2P Botnets operate to a certain extent decentralized and some completely decentralized. Those Botnets that are completely decentralized allow a BotMaster to inject a command into any Bots, and have it either be broadcasted to a specified node. Since P2P Botnets usually allow commands to be injected at any node in the network, the authentication of commands become essential to prevent other nodes from injecting incorrect commands.

## 2.3 Virus

Virus is a computer program which transmits from one computer to another computer by attaching itself to another program. The program that the virus attaches itself to is one of the victim's programs or files. [29] There are many different way for transmitting virus to other computers such as by sending infected file as an email attachment or by embedding copies of infected files into removable medium such as a CD, DVD or USB drive. Viruses can increase their chances of spreading to other computers by infecting files on a network file system or a file system that is accessed by another computer.[30,31] One of the crucial differences between virus and worm is capability of worm for automatically spreading itself to other computers in the network by exploiting computer's security vulnerabilities.

A virus usually consists of two parts: (i) Insertion code (ii) Payload

- *Insertion code:* this is a code which is responsible to insert a copy of the virus into other files on the infected computer. This part is obligatory for all kind viruses.
- *Payload:* this is a code which is responsible for malicious activities that virus may perform. This part is completely optional and just relies on the purpose of designing virus. The payload of virus could range from the virus that overwrites or deleted files on the system such as overwriting the Flash Bios of system or overwriting special media file types to a new type of virus that is not designing for damaging files on the system, but allowing backdoor access to the system for stealing sensitive information such as passwords.

There are some policies which designers have to take into consideration for efficient designing virus. Since designing efficient antivirus is also relies on understanding of these policies, we are categorizing these policies into four main groups which are:

a) Transmitting policy: the conditions that have to be provided for spreading virus. It usually depends on which part of computer has to be infected. As an example, virus tries to infect computer's boot program.
b) Infection policy: it relies on the files that have to be chosen from targeted computer to be infected. As an example, some virus are rely on the use of macros, therefore; infection policy is based on finding files in victim computers which support these macros.
c) Code insertion policy: the regulations for inserting code into targeted files. Therefore, good nderstanding

of files layout which contain executable programs is needed. The simplest policy is to insert the code at the beginning or end of the files.

d) Execution policy: the policy that is used for executing virus and its infected code in victim computer.

## 2.4 Rootkit

A Rootkit is an automated software package which can be used by hacker to mask intrusion and to gain administrative ("root") privileges on a computer or computer network. In other way, we can say that Rootkit is a collection of tools for several purpose, such as gathering information about the system and its environment by using network sniffers, providing a backdoor into the system which enable hacker to gain access to system at some later time, mask the fact that system has been compromised and many other similar purposes. Rootkit usually install a handler which can remove the audit records and other records of the rootkit. [32]

The main point that we have to take into consideration is that the purpose of Rootkit is not to get access to victim computer, but to preserve existing access by hiding its malicious resources and other techniques which can be used. It means that other malware such as worms and Trojans are utilizing Rootkit to hide their presence in victim computer in order stay longer.

The main malicious activities that Rootkit can do are:

- Provide unauthorized access to victim computer
- Mask malicious resources( e.g. processes, files, open ports, registry keys)
- Clean logs of the victim computer system which make the trace of hacker much complicated

In general, we can classify Rootkit into two groups which are: (i) user mode Rootkit and (ii) kernel mode Rootkit

- *User mode Rootkit:* in this mode Rootkit substitute certain system files which are used to extract information from the system. It means that Rootkit in this mode needs a variety of binaries to be manipulated. Today's common rootkit usually run in user mode.
- *Kernel mode Rootkit:* in this mode Rootkit place the malicious code inside the kernel by altering the kernel.

## 2.5 Trojan horse

Trojan horse is malicious software that can be hiding in a victim computer. In contrast to worm and virus, Trojans do not have their own on-board replication and spreading capability. Therefore, maybe it is better we say Trojan horse is a virus which cannot be replicated. There are many ways for infecting victim computers by Trojans such as downloading from a remote site, but recently Trojans use worm and virus for penetrating into victim computers.

There is special kind of Trojan which can be controlled remotely and receive commands from attackers. This capability of Trojan makes it similar to bot. However, the main difference between remote-access Trojan program and botnet is that botnet is a group of compromised computers which can be controlled remotely under a same command-and-control mechanism.

Trojans similar to worms can have approximately any consequence on the target hosts. Some well-known examples are:

a) To delete processes or files which are belonging to critical applications.

b) To modify the host files which can redirect access to security update site to another site containing malicious code or even blocking such access.

c) To steal confidential information from the infected computers.

d) To open ports on the infected computers. This can be done remotely by hacker or by Trojan code itself.

We categorize Trojan horse into two main groups: (i) General Trojan (ii) Remote-Access Trojan

- *General Trojans:* this type of Trojans has wide range of malicious activities. They can threaten data integrity of victim machines. They can redirect victim machines to a particular web site by replacing system files that contain URLs. They can install several malicious software on victim computers. They can even track user activities, save those information and then send it to attacker.
- *Remote-Access Trojans:* we can claim that they are the most dangerous type of Trojan. They have special capability which enable attacker to remotely control victim machine via a LAN or Internet. This type of Trojan can be instructed by attacker for malicious activities such as harvesting confidential information from the victim machine.

## 2.6 Backdoor

Malware which is installed by attackers who have compromised a system to allow them to get access to the victim computer without going through any normal authentication and login procedures. It means that backdoor facilitate attackers subsequent return to the compromised system. [33]

Actually backdoors are the most widespread type of Trojans which can be categorized in the group of Remote-access Trojans. Therefore, we can conclude that backdoor also has the ability to remotely be controlled via a LAN or internet.

Backdoors are used by attacker for many purposes such as stealing confidential information, executing malicious code, deleting data, rebooting the victim machine and more importantly can include the machine in Botnet.

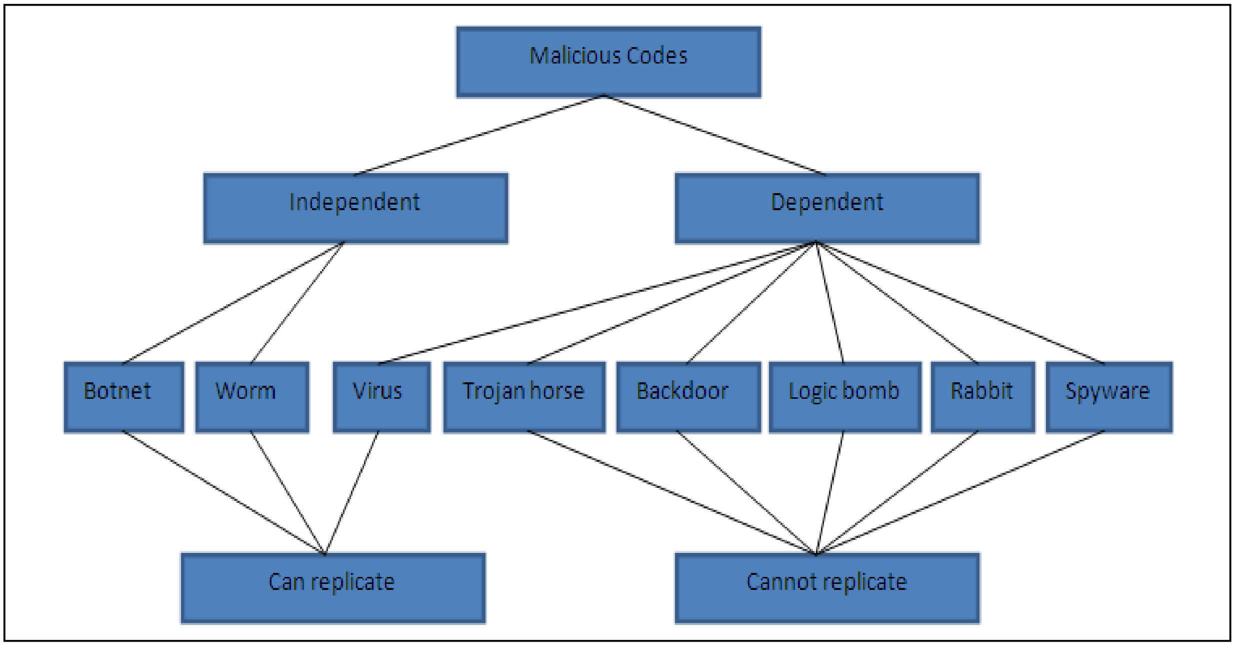

Figure 4. Classification of Malwares

### 2.7 Logic bomb

A logic bomb, also called slag code, is programming code which lies inactive until a particular piece of program logic or specific event activates it. There are some common activators which are the arrival of a specific date or time, certain message from the programmer, creation or deletion of some specific information and even can be activated when something does not happen. Actually logic bomb cannot replicate itself. It means that a logic bomb just infect intended victims.

Logic bomb can be used in many areas but the classic use for it is to guarantee payment for software or makes free software trials possible for a period of time. It means that if payment did not make by a specific date or the period of time has been passed, the logic bomb can be activated and takes proper measures such as software automatically delete itself or even the logic bomb delete certain information on the system.
Viruses, worms and even Trojans can contain logic bombs that execute a special payload when some certain condition is met.

### 2.8 Rabbit

Rabbit, also called Bacteria, is a malicious code that mainly designed to use up large amounts of system resources such as message buffers and file space by creating many instances of itself or run many times simultaneously. Similar to logic bombs and unlike worms, Rabbit do not necessarily spread over the network.

### 2.9 Spyware

It is a software program which penetrates to the victim machines and secretly without permission of the user sends personal information to the third party. [34] The information that Spyware can steal and sends to the third party usually are user ID and password, crucial documents and key strokes of the user.

## 3 Classification

As we mentioned earlier, there is not any clear definition and classification for different kind of malwares. In this part, we classify Malwares based on two important characteristics of Malwares which are (i) dependency (ii) replication.

The group of Malwares which needs victim's host program to execute is classified as dependent malwares and the group which does not need victim's host program to execute their code is classified as independent Malwares, as shown in Figure 4.

The group of Malwares which has the ability to replicate in the network are classified as Can replicate Malwares and the rest of them are in Cannot replicate group.

## 4 Conclusions

Malware is malicious software which developed to penetrate computers in a network without the permission or notification of users. In this paper, we proposed an inclusive classification of malware and tried to investigate the operation and distribution principles of them. Because the most serious threat which occurs commonly in today's cyber attacks are

Botnets and worms, we tried to concentrate more on them and their communication topologies.

# 5 References

[1] M.D. Preda, M. Christodorescu and S. Jha, S. Debray, A Semantics-Based Approach to Malware Detection, *ACM SIGPLAN-SIGACT symposium on principles of programming* languages, University of Verona, University of Wisconsin, University of Arizona, 2007.

[2] P. Denning. The science of computing: Computer viruses. *American Scientist*, 76(3):236–238, May 1988.

[3] P. Denning. *Computers under Attack: Intruders, Worms and Viruses*. Addison-Wesley, Reading, Mass., 1990.

[4] SP 800-61, Computer security incident handling guide, NIST, 2004.

[5] C. E. Pelaez, J. Bowles, "Computer Virus", IEEE, 1991

[6] E.H. Spafford, "The internet worm incident", In ESEC'89 2nd European Software Engineering Conference, Coventry, United Kingdom, 1989.

[7] J. F. Shoch and J. A. Hupp. The ”Worm” program – Early experience with a distributed computation. *Communications of the ACM*, 25(3):172–180, 1982

[8] P. Barford and V.Yagneswaran, "An Inside Look at Botnets". In: Special Workshop on Malware Detection, Advances in Information Security, Springer, Heidelberg (2006).

[9] N. Ianelli, A. Hackworth, Botnets as a Vehicle for Online Crime, CERT, December 2005.

[10] E. Cooke, F. Jahanian, and D. McPherson, "The zombie roundup: Understandinng, detecting, and disrupting botnets," Proc. Of Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI'05), June 2005.

[11] Honeynet Project, Know your Enemy: Tracking Botnets, March 2005. http://www.honeynet.org/papers/bots

[12] M.A Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "A multifaceted approach to understanding the botnet phenomenon," *6th ACM* SIGCOMM on Internet Measurement Conference, IMC 2006, 2006, pp. 41-52.

[13] Crimeware:Bots. Web publication, Available at URL: http://www.symantec.com/norton/cybercrime/bots.jsp

[14] J. R. Binkley, S. Singh, "An algorithm for anomaly-based botnet detection," Proc. of 2nd Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI'06), July 2006, pp 43-48.

[15] F. Freiling, T. Holz, and G. Wicherski, "Botnet Tracking: Exploring a Root-Cause Methodology to Prevent Distributed Denial-of-Service Attacks," Proc. of 10th European Symposium On Research In Computer Security (ESORICS'05), 2005

[16] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee, "Bothunter: Detecting malware infection through ids-driven dialog correlation," Proc. Of the 16th USENIX Security Symposium, 2006.

[17] G. Gu, J. Zhang, and W. Lee, "BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic," Proc. of the 15th Annual Network and Distributed System Security Symposium (NDSS'08), San Diego, CA, February 2008.

[18] A. Karasaridis, B. Rexroad, and D. Hoein, "Widescale botnet detection and characterization," Proc. of Hot Topics in Understanding Botnets (HotBots'07), April 2007.

[19] Duc T. Ha, Guanhua Yan, Stephan Eidenbenz, Hung Q. Ngo. "On the effectiveness of structural detection and defense against P2P-based Botnet," Proc. of the 39th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'09), June 2009.

[20] J. Oikerinen , D. Reed. Internet Relay Chat protocol. May 1993. Web publication. Available at URL: http://tools.ietf.org/html/rfc1459#section-1

[21] J.Stewart.Bobaxtrojananalysis. http://www.secureworks.com/research/threats/bobax/

[22] N. Daswani, M. Stoppelman and the Google Click Quality and Security Teams, "The anatomy of ClickBot.A.," Proc. of the 1st Workshop on Hot Topics in Understanding Botnets (HotBots 2007).

[23] K. Chiang and L. Lloyd, "A case study of the rustock rootkit and spam Bot," Proc. of the 1st Workshop on Hot Topics in Understanding Botnets (HotBots 2007). 2007.

[24] Iv´an Arce and Elias Levy. An analysis of the slapper worm. IEEE Security and Privacy Magazine, 1(1):82–87, 2003.

[25] J. Stewart. Sinit P2P trojan analysis. Web publication. Available at URL: http://www.secureworks.com/research/threats/sinit,2003

[26] J. Stewart. Phatbot trojan analysis. Web publication. Available at URL: http://www.secureworks.com/research/threats/phatbot/,2004

[27] S. Stover, D. Dittrich, J. Hernandez, and S. Dietrich. Analysis of the storm and nugache trojans: P2p is here. *;login:*, 32(6), 2007.

[28] SecureWorks. SpamThru trojan analysis, October 2006. http://www.secureworks.com/analysis/spamthru/

[29]Vesselin Vladimirov Bontchev, *Methodology of Computer Anti-Virus Research*, PhD thesis, University of Hamburg, Germany, 1998

[30] http://www.bartleby.com/61/97/C0539700.html

[31]http://www.actlab.utexas.edu/~aviva/compsec/virus/whatis.html

[32] Stephen M. Specht, Ruby B. Lee, Distributed Denial of service: taxonomies of attacks, tools and countermeasures, Princeton architecture laboratory for multimedia and security, technical report, 2003.

[33] Yin Zhang, Vern Paxon, detecting backdoors, Cornell University, AT&T center for internet research at ICSI, 2000.

[34] Tzu-Yen Wang, Shi-Jinn Horng, Ming-Yang Su, Chin-Hsiung Wu, Peng-Chu Wang and Wei-Zen Su, A Surveillance Spyware Detection System Based on Data Mining Methods, *IEEE Congress on Evolutionary Computation*, 2006