EDOWA Worm Classification

Madihah Mohd Saudi, Emran Mohd Tamil, Siti Aishah Md Nor, Mohd Yamani Idna Idris and Kamaruzzaman Seman

Abstract—**Worms have become a real threat for computer users for the past few years. Worm is more prevalent today than ever before, and both home users and system administrators need to be on the alert to protect their network or company against attacks. It is coming out so fast these days that even the most accurate scanners cannot track all of the new ones. Indeed until now there is no specific way to classify the worm. To understand the threats posed by the worms, this research had been carried out. In this paper the researchers proposed a new way to classify the worms which later is used as the basis to build up a system which is called as the EDOWA system to detect worms attack. Details on how the new worm of classification which is called as EDOWA worm classification is produced are explained in this paper. Hopefully this new worm classification can be used as the basis model to produce a system either to detect or defend organization from worms attack.**

Index Terms - **Classification, payload, worm and worm analysis**

I. INTRODUCTION

Computer worm can caused millions dollars of damage by infecting hundreds and thousands of host in a very short period of time. A computer worm is a computer program or a small piece of software that has the ability to copy itself from machine to machine. It uses computer networks and security holes to replicate itself. Computer worm is classified as a highly threat to the information technology world. McCarthy [5] defines computer worm as a standalone malicious code program that copies itself across networks. Meanwhile Nachenberg [6] stated that the computer worm is a program that is designed to copy itself from one computer to another, dominate some network medium such as through email. The computer worm would infect as many machines as possible on the network. The prototypical computer worm infects a target system only once; after the initial infection, the worm attempts to spread to other machines on the network.

While there are thousands of variations of computer worms, the classification of computer worm can be done via several ways Nazario [11] proposed a function structure framework that consist six components. The components are reconnaissance capabilities, specific attack capabilities, a command interface, communication capabilities, intelligence capabilities and unused attack capabilities. The framework mainly predicts the future research on network worms. Another form of classification of computer worm Weaver et al [9], they classified computer worm into five major classifications: target discovery, carrier, activation, payload and attackers. As for Kienzle et al [19], they classified computer worm into three basic classes by propagation strategies. The computer worms are classified into e-mail worm, windows file-sharing worm and traditional worm.

Madihah Mohd Saudi is with the Faculty Science and Technology, Islamic Science University of Malaysia (USIM), Bandar Baru Nilai, 71800 Nilai, Negeri Sembilan, Malaysia (email: madihah@usim.edu.my).

Emran Mohd Tamil is with the Faculty of Computer Science and Information Technology, University of Malaya (UM), Kuala Lumpur, Malaysia (email: emran@um.edu.my).

Siti Aishah Md Nor was with the Faculty Science and Technology, Islamic Science University of Malaysia (USIM), Bandar Baru Nilai, 71800 Nilai, Negeri Sembilan, Malaysia.

Mohd Yamani Idna Idris is with the Faculty of Computer Science and Information Technology, University of Malaya (UM), Kuala Lumpur, Malaysia (email: yamani@um.edu.my).

Professor Dr. Kamaruzzaman Seman is with the Faculty Science and Technology, Islamic Science University of Malaysia (USIM), Bandar Baru Nilai, 71800 Nilai, Negeri Sembilan, Malaysia (email: drkzaman@usim.edu.my).

II. METHOD OF TESTING

In order to produce a new worm classification, the researchers' had conducted few testing and researches. A worm code analysis laboratory is build to test and analyze the worm. A controlled laboratory environment is built to conduct the testing. This laboratory is not connected to the real network. Three machines were used and connected in LAN using a hub. Figure 1 below illustrates the laboratory.

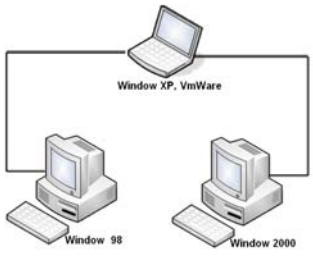

Fig. 1. Worm Code Analysis Lab Architecture.

A. Worm Analysis Process

For a new computer worm, the main purpose of analyzing it is to know the intention of the code. As for computer worm that has been released in real time, the worm analysis

process is for verification of what the worm source code intention and to verify as what has been published in anti virus website or CERT website or it might be a new worm feature. The analysis techniques can be divided into two techniques, either the static analysis or dynamic analysis. Before loading the computer worm specimen into the machine, researchers make sure all the preparation and the verification has been done. While conducting the analysis, all the process is document in writing. A written record of analytic techniques and the computer worm action is useful in understanding how the computer worm works, tracing through its function in a repeatable fashion and improving the worm analyst skills.

B. Loading Specimen

When all the preparation and the verification have been done, the lab is disconnected from any production network. The USB thumb drive is used to transfer the computer worm specimen onto the lab system.

Once the specimen already placed in the lab system, the analysis can be carried out. To determine the purpose and the capabilities of this piece of code, researchers can used either the static analysis or dynamic analysis.

1) Static Analysis

Static analysis is also known as white box analysis. It involves analyzing and understanding source code. If only binary code available, the binary code has to be compiled to get the source code. White box analysis is very effective in finding programming errors and implementation errors in software and to know the flow of the program. The static analysis looks at the files associated with the computer worm on the hard drive without running the program. With static analysis, the general idea of the characteristics and purpose of the computer worm can be analyzed. The static analysis phase involves antivirus checking with research, analyzing the strings, looking for scripts, conducting binary analysis, disassembling and reverses compiling.

a) Antivirus checking with research

When computer worm specimen already copied to the testing machine, researchers will run the antivirus to check if the antivirus installed detects anything. If the antivirus detects the computer worm, check the name of the computer worm and search it in any antivirus website for further information. If the computer worm is in compressed or archived form, researchers will open the archive to get its content. The researchers need to verify if the information available from the antivirus website is correct.

b) String Analysis

The strings that extracted from the computer worm could help the researchers to know more about the computer worm characteristics. A few tools such as TDS3 and Strings.exe (from Sysinternal) were used to extract the strings. The information that could be retrieved from the extracted strings are consist of the worm specimen's name, user dialog, password for backdoors, URLs associated with the malware, email address of the attacker, help or command-line options, libraries, function calls and other executables used by the malware.

c) Looking for script

The language written for the computer worm can be identified based on strings extracted from it. Table 2 below is some clues:

TABLE 1
SCRIPTING LANGUAGE

Scripting Language	Identifying Characteristics Inside the File	File's Common Suffix
Bourne Shell Scripting language	Starts with line !#/bin/sh	.sh
Perl	Start with line !#usr/bin/perl	.pl, .perl
JavaScript	Includes the word javascript or JavaScript, especially in the form <Script language = "JavaScript">	.js, .html, .htm
Visual Basic Script (VBScript)	Includes the word VBScript, or characters vb scattered throughout the file	.vbs, .html, .htm

d) Disassemble code

Disassemble and debugger is used to convert a raw binary executable into assembly language for further analysis. Researchers use the tools that have been listed in Appendix A to disassemble and debug the computer worm.

2) Dynamic Analysis

Dynamic analysis involves executing the computer worm and watching its actions. The computer worm is activated on a controlled laboratory system.

a) Monitoring file activities

Most computer worms read from or write to the file system. It might attempts to write files, altering existed programs, adding new files or append itself to the file system. By using tool such as Filemon all actions associated with opening, reading, writing, closing and deleting files can be monitored.

b) Monitoring process

The monitoring tool such as Prcview v3.7.3.1 or Process Explorer, displays each running program on a machine, showing the details of what each process is doing. With this kind of tool the files, registry keys and all of the DLLs that each process has loaded can be monitored. For each running process, the tool displays its owner, its individual privileges, its priority and its environment variables.

c) Monitoring Network activities

From a remote machine which will be in the same LAN with the infected testing machine, the port scanner, Nmap program and a sniffer will be installed. The port scanner

and Nmap program are used to monitor the listening port. A sniffer will be installed to sniff the worm traffic. All of the related tools like Ethereal, NeWT and TDS-3 use the sniffer. Using the sniffer, details of individual packets and all packets transmitted across the LAN can be monitored. As for the local network monitoring tool (TDIMon), it will monitor and records all request to use the network interface and show how the worm grabbed the network resources and used them.

The computer worm might have placed the network interface in promiscuous (broadcast) mode which allowed it to sniff all packets from LAN. To determine if the infected machine in promiscuous mode state of the interface, run the Promiscdetect.exe tool.

d) Monitoring registry access

The registry need to be monitored as the registry is the hierarchical database containing the configuration of the operating system and most programs installed on the machine. The monitoring registry access can be done by using the Regmon.

III. EDOWA CLASSIFICATION

After the analysis done in the laboratory, it leads researchers to produce a new classification for the EDOWA system. A proposal of the classification of worm is made. This classification is based on several factors: infection, activation, payload, operating algorithms and propagation. Figure 2 is an overview of the EDOWA classification.

Infection is the first step before a worm infects a computer. The activation is a mechanism that will activate a worm. Payload is a code that carries a destructive mechanism of a worm. Operating algorithms is a techniques used to avoid detection. Finally, the propagation mechanisms are how the worm spread to reproduce.

Sections below will elaborate more details on the EDOWA classification.

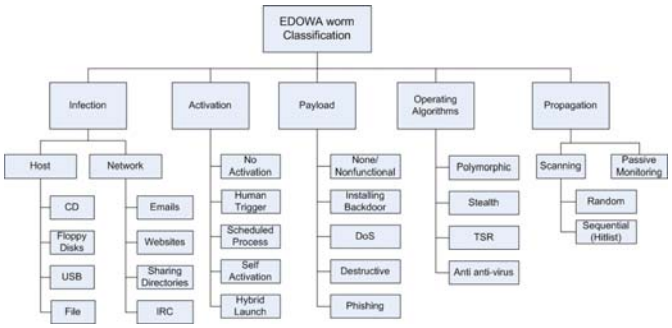

Figure 2. EDOWA worm classification

1) INFECTION

Infection is the phase on how a computer gets infected by a worm. From the research of eight classifications that is available, we found out that only one research made infection as the first phase. Albanese et al [14] says that infection refers to how a worm gains initial control of a system. Worms rely on two general methods to infect a host. Either they exploit an error in software running on a system, or they are the result of some action taken by a user. Here we proposed two techniques:

a) Host

Host is a mechanisms needed by the worm copy itself to a new systems that are not yet been infected. It cannot propagate autonomously across the network. Host computer worms where the original terminates itself after launching a copy on another host so there is only one copy of the worm running somewhere on the network at any given moment. It requires human help to move from one machine to another. CD, Floppy Disks, USB (thumb-drive and external hard disk) and File are the most common host available now.

b) Network

Network is the fastest way in moving worm. It consists of multiple parts, each running on different machines and possibly performing different actions also using the network for several communication purposes. Propagating from one machine to another is only one of those purposes. It can infect a computer without human interaction. Most simply copy themselves to every computer with which the host computer can share data. Most Windows networks allow machines within defined subgroups to exchange data freely, making it easier for a worm to propagate itself.

2) ACTIVATION

For this classification, activation is defined as a trigger mechanism of a worm. This phase is where the worm enters the host once it found a machine. According to Nazario [11] in his book "Defense and Detection Strategies against Internet Worms", stated these are used to launch an attack against an identified target system.

a) No Activation

Worm with no activation will just stay in the computer doing nothing. It just used up some hard disk space.

b) Human trigger

Human trigger is the slowest activation mechanisms. Usually this approach use worm that propagates using emails. The social engineering technique is used to attract user to click on the file to activate the worm [15]. According to Christoffersen et al [12], some worms are activated when the user performs some activity, like resetting the machine, logging onto the system and thereby running the login scripts or executing a remotely infected file. Evidently, such worms do not spread very rapidly.

c) Schedule Process

Based on Weaver et al [9], the second fastest worms activate is by using scheduled system processes. Schedule process is an activation that is based on specific time and date. Many desktop operating systems and applications

include auto-updater programs that periodically download, install and run software updates.

d) Self Activation

The worms that are fastest activated are able to initiate their own execution by exploiting vulnerabilities in services that are always on and available (e.g., Code Red [2] exploiting IIS Web servers) or in the libraries that the services use (e.g., XDR [13]). Such worms either attach themselves to running services or execute other commands using the permissions associated with the attacked service.

e) Hybrid Launch

Hybrid Launch uses the combination of two or more activation mechanism to launch a worm. ExploreZip [6] is an example of a hybrid-launch worm. ExploreZip send an e-mail that required a user to launch the infected attachment to gain control of the system. Once running on the computer system, ExploreZip would automatically spread itself to other computers over the peer-to-peer network. These targeted machines would then become infected on the next reboot without any known user intervention.

3) PAYLOAD

For this classification, payload is defined as a destructive mechanism of a worm. A payload is code designed to do more than spread the worm. Many worms have been created which are only designed to spread, and don't attempt to alter the systems they pass through.

a) No Payload

Worm with no payload does not do any harm to the computer system. This kind of worm will just propagate without infecting any destructive mechanisms to the computer.

b) Installing backdoor

Backdoor is a term used to describe a secret or undocumented means of getting into a computer system. Many programs have backdoors placed by the programmer to allow them to gain access to troubleshoot or change the program. Some backdoors are placed by hackers once they gain access to allow themselves an easier way in next time or in case their original entrance is discovered. Example of backdoor attack is the worm called Blaster [1] that used the backdoor mechanism to transfer the worm payload to newly infected systems.

c) Denial of services

A denial of service (DoS) attack floods a network with an overwhelming amount of traffic, slowing its response time for legitimate traffic or grinding it to a halt completely. The more common attacks use built-in “features” of the TCP/IP protocol to create exponential amounts of network traffic. Example of DoS attack is the well-known worm called Code Red [2] was programmed to unleash a denial-of-service attack on the Whitehouse.gov that targeting the actual Whitehouse.gov IP address.

d) Destructive

This will do harm to the machine or the host. According to Shannon et al [4], Witty worm deletes a randomly chosen section of the hard drive, which, over time, renders the machine unusable.

e) Phishing

Phishing [7] is a criminal activity using social engineering techniques. Phishers attempt to fraudulently acquire sensitive information, such as usernames, passwords and credit card details, by masquerading as a trustworthy entity in an electronic communication. eBay and PayPal are two of the most targeted companies, and online banks are also common targets. Phishing is typically carried out by email or instant messaging, and often directs users to give details at a website, although phone contact has been used as well. Attempts to deal with the growing number of reported phishing incidents include legislation, user training, and technical measures.

4) OPERATING ALGORITHMS

Operating algorithms is defined as a detecting techniques used by worms to avoid detection. Among the eight classifications that are available, we found out that only one research have operating algorithms in their classification. Albanese et al [14] classified it as survival. Operating algorithms are the mathematical and logical ways that a worm attempts to avoid detection. It can be categorized as:

a) Polymorphic

A polymorphic worm is a worm that changes all part of their code each time they replicate, this can avoid scanning software. Kruegel et al [16] paper defined polymorphic worms as a worm that is able to change their binary representation as part of the spreading process. It can be achieved by using self-encryption mechanisms or semantics-preserving code manipulation techniques. As a consequence, copies of a polymorphic worm might no longer share a common invariant substring of sufficient length and the existing systems will not recognize the network streams containing the worm copies as the manifestation of a worm outbreak.

b) Stealth

Stealth worm use a concealment mechanisms. It spread slow, evokes no irregular communication pattern and spread in an approach that makes detection hard. Cheetancheri [17] stated in his thesis that the goal of stealth worm is to spread to as many hosts as possible without being detected. However, once such a worm has been detected, manual means of mitigation are possible.

c) Terminate and stay resident (TSR)

Terminate and stay resident (TSR) worm exploit a variety of techniques to remain resident in memory once their code has been executed and their host program has terminated. These worms are resident or indirect worm, known as such because they stay resident in memory, and indirectly find files to infect as they are referenced by the user.

d) Anti anti-virus

Anti anti-virus will corrupt the anti-virus software by trying to delete or change the anti-virus programs and data files so the anti-virus does not function properly. According to Nachenberg [18], anti anti-virus or are usually called retroviruses, are computer viruses that attack anti-virus software to prevent themselves from being detected. Retroviruses delete anti-virus definition files, disable memory resident anti-virus protection and attempt to disable anti-virus software in any number of ways.

5) PROPAGATION

Propagation is defined as a worm that spread itself to another host or network. After researching the propagation issue, we strongly believe that there are two ways for a worm to reproduce itselfs: scanning and passive.

a) Scanning

Scanning is a method used by worms to find their victims. We strongly agree with the method proposed by Weaver et al. [9]. There are two possible scanning method that is random scanning and sequential scanning.

(1) Random scanning

It is the most popular method where the worm simply picks a random IP address somewhere in the Internet Address space and then tries to connect to it and infect it. Example of a random scanning worm is the Blaster [1] that picks a random number to determine whether to use the local address it just generated or a completely random one.

(2) Sequential scanning (Hitlist)

The worm releaser scans the network in advance and develops a complete hitlist of all vulnerable systems on the network. According to Staniford [8], the worm carries this address list with it, and spreads out through the list.

(3) Passive

Worms using a passive monitoring technique are not actively searching for new victims. Instead, they are waiting for new targets to contact them or rely on the user to discover new targets. Christoffersen et al [12] says passive worms tend to have a slow propagation rate, they are often difficult to detect because they generate modest anomalous reconnaissance traffic.

IV. CONCLUSION

This new classification is produced based on the research and testing that have been done in the laboratory. The classifications are divided into five main categories: Infections, Activation, Payload, Operating Algorithms and Propagation. Efficient Detection of Worm Attack (EDOWA) system is produced based on this classification. This EDOWA system is not discussed in this paper. Hopely this paper can be used as the basis model for worm classification and can be used for other upcoming research.

REFERENCES

[1] M. Bailey, E. Cooke, F. Jahanian, D. Watson and J. Nazario, "The Blaster Worm: Then and Now," *IEEE Security & Privacy*, vol.3, no.4, pp. 26-31, 2005.

[2] H. Berghel, "The Code Red Worm: Malicious software knows no bounds," *Communication of the ACM*, vol.44, no.12, pp.15-19, 2001.

[3] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford and N. Weaver, "Inside the Slammer Worm," *IEEE Security & Privacy*, vol.1, no.4, pp.33-44, 2003.

[4] C. Shannon and D. Moore, "The Spread of the Witty Worm," *IEEE Security & Privacy*, vol.2, no.4, pp.36-50, 2004.

[5] L. McCarthy, "Own Your Space: Keep Yourself and Your Stuff Safe Online (Book)," Addison-Wesley Professional, 2006.

[6] C. Nachenberg, "Computer Parasitology", *Proceedings of the Ninth International Virus Bulletin Conference*, September/October 1999, pp 1-25.

[7] A. Tsow, "Phishing With Consumer Electronics: Malicious Home Routers," *15th International World Wide Web Conference (WWW2006)*, Edinburgh, Scotland, May 2006.

[8] S. Staniford, President of Silicon Defense. "The Worm FAQ: Frequently Asked Questions on Worms and Worm Containment," *The Worm Information Center*, 2003.

[9] N. Weaver, V. Paxson, S. Staniford and R. Cunningham, "A Taxonomy of Computer Worms," *Proceedings of the ACM CCS Workshop on Rapid Malcode (WORM)*, pp.11-18, 2003.

[10] S. Staniford, V. Paxson and N. Weaver, "How to Own the Internet in Your Spare Time," *Proceedings of the 11th USENIX Security Symposium*, August 2002.

[11] J. Nazario, "Defense and Detection Strategies against Internet Worms" (BOOK), *Artech House Inc.*, 2003. Or a paper entitles "The Future of Internet Worm" by Nazario, J., Anderson, J., Wash, R., and Connelly, C. Crimelabs Research. 2001.

[12] D. Christoffersen and B.J. Mauland, "Worm Detection Using Honeypots (Thesis or Dissertation style)," Master dissertation, Norwegian University of Science and Technology, June 2006.

[13] CERT. CERT Advisory CA-2002-25 Integer Overflow in XDR Library, http://www.cert.org/advisories/ca-2002-25.html.

[14] D.J. Albanese, M.J. Wiacek, C.M. Salter, and J.A. Six, "The Case for Using Layered Defenses to Stop Worms (Report style)," UNCLASSIFIED-NSA Report, pp.10-22, 2004.

[15] C.C. Zou, D. Towsley and W. Gong, "Email worm modeling and defense," *Computer Communications and Networks, ICCCN 2004*, pp.409-414, 2004.

[16] C. Kruegel, E. Kirda, D. Mutz, W. Robertson and G. Vigna, "Polymorphic Worm detection using structural information of executables," *8th International Symposium on Recent Advances in Intrusion Detection (RAID)*, 2005.

[17] S.G. Cheetancheri, "Modeling a computer worm defense system (Thesis or Dissertation style)," Master dissertation, University of California, 1998.

[18] C. Nachenberg, "The Evolving Virus Threat," *23rd NISSC Proceedings*," Baltimore, Maryland, 2000.

[19] D.M. Kienzle and M.C. Elder, "Recent Worms: A Survey and Trends," *Proceedings of the ACM CCS Workshop on Rapid Malcode (WORM)*, pp.1-10, 2003.