A Survey on Malware and Malware Detection Systems

Imtithal A. Saeed
Faculty of Computing
Universiti Teknologi Malaysia,
81310 UTM
Johor Baharu Campus, Johor,
Malaysia

Ali Selamat
Faculty of Computing
Universiti Teknologi Malaysia,
81310 UTM
Johor Baharu Campus, Johor,
Malaysia

Ali M. A. Abuagoub
College of Computer
Engineering & Sciences,
Salman bin Abdulaziz
University,
Alkharj, KSA

ABSTRACT

Over the last decades, there were lots of studies made on malware and their countermeasures. The most recent reports emphasize that the invention of malicious software is rapidly increasing. Moreover, the intensive use of networks and Internet increases the ability of the spreading and the effectiveness of this kind of software. On the other hand, researchers and manufacturers making great efforts to produce anti-malware systems with effective detection methods for better protection on computers. In this paper, a detailed review has been conducted on the current situation of malware infection and the work done to improve anti-malware or malware detection systems. Thus, it provides an up-to-date comparative reference for developers of malware detection systems.

Keywords

Malware, Malware Detection Systems, Antivirus.

1. INTRODUCTION

A more recent report from McAfee says "malware continues to grow" [1]. Thousands of new malware appear very quickly, reports from G Data and King soft Laboratory said[2, 3]. In contrast, researchers and manufacturers evolve new methods to produce improved techniques for building anti-malware [4-8]. The techniques used for creating malicious software can be categorized, in this review, into groups depending on creation and obfuscation techniques, invocation methods, platform, spreading and propagation techniques.

Malware detection system is a system used to determine whether a program has malicious intent or not [9]. Detection system includes two tasks, detection and analysis[10]. The malware detection system may or may not exist in the same system it is protecting [11]. And sometimes it's tasks divided into client and server, such as in cloud-based antivirus [8, 12]. Many improvements made on both aspects of detection and analysis [3, 10, 13-17].

In addition, technological solutions added to increase the effectiveness and the performance of malware detection systems. Such that the use of cloud computing [8], network-based detection system [18], web, virtual machine [19, 20], agent technology[21-27] or by the use of hybrid methods and technologies.

The main goal of this review paper is to investigate the current situation regarding malware and their detection systems. Moreover, the study includes analysis of the techniques and technologies used for building anti-malware.

The rest of the paper is organized as follows: Section 2 defines malware with their main. Section 3 describes the techniques used for the creation and obfuscation of malware. Section 4 discusses and compares malware classes. An extensive review of malware detection systems is presented in Section 5. Section 6 concludes the paper with remarkable comments.

2. MALWARE DEFINITION

The term malware comes from combining the two words malicious and software, and to be used to indicate any unwanted software. It was defined, generally, by G. McGraw and G. Morrisett as “any code added, changed, or removed from a software system in order to intentionally cause harm or subvert the intended function of the system” [28]. In [29] a virus has been defined as “ a generic term that encompasses Viruses, Trojans, Spywares and other intrusive code”.

Malware characterized by the ability of replication, propagation, self-execution and corruption of computer system. The corruption of computer system can affect information confidentiality, integrity and denial of services.

Replication is the important characteristics for most malware, as it ensures its existence. In some malware cases incessant replication makes exhaustion of computer resources (e.g. hard disk, RAM).

Invisibility property is used by many of malware types to evade themselves from being detected by anti-malware. It can be done by one of polymorphic or metamorphic techniques [29].

The common way for infecting a system (data or executable files, boot records of disk drives or exhausting network bandwidth) is to transfer malware from a polluted device to another uninfected one, using local or network filesystem. A malware make use of operating system vulnerabilities and software bugs, as few of software contain faults. It plants itself in to start its lifecycle at the same system or remotely controls the infection operation on another system.

3. MALWARE TECHNIQUES

For creating malware, attackers use various ways ranging from simple ordinary techniques that inserting a special piece of codes into a program file, to complex ones that use sophisticated algorithm to create obfuscated and polymorphic malware. The kind of malware produced by the ordinary techniques can be identified easily by extracting some unique characteristics to combine what called a signature.

In polymorphic malware there is variable malware in which syntaxes of mal-code mutate in each time of infection, but the semantic remain the same without change. Encryption techniques are the most common methods used in polymorphic malware.

Obfuscated malware include polymorphic and metamorphic malware, in which the original code transformed into a form that is functionally the same but is much more difficult to be understood. The obfuscation techniques that used are dead-code (which is inserting some number of code that accomplish nothing), code transportation (by inserting jumps in the program while its control flow remains the same), register renaming (by the mean of replacing the use of register in an instruction with another unused one) or toolkit paradigm where a set of variant malware, that generates one type of malware in each time of infection.

Remote execution of malware is done by hackers to achieve their intention remotely using the infrastructure of the Internet and benefiting from the existing methods of remote execution.

4. MALWARE CLASSES

Several malware classifications have been issued so far, depending on some of their characteristics. The purpose of such classifications is to facilitate the tracking of authorship, correlating information, identifying new variants [30]. However, in this paper, a kind of classification, depending on the use of networks and Internet, is made. Using of networks and the Internet is that they represent the execution environments of malware or as means of propagation. The idea behind such classification is because the use of networks and the Internet necessitate dealing with this type of software in special ways. That is like intrusion detection systems (IDS), prevention of SQL attacks, detecting worm spreading on LAN, real-time classification of malware etc. The classification made, is to categorize the major common malware types into groups depending on the network and web usage.

4.1 Network-based Malware

Spyware is a kind of malware that is installed secretly on a user computer for the purpose of collecting information about users without their knowledge [31]. Even reputable vendors of software like Microsoft and Google, intentionally, collect information of their users using spywares[10].

Adware is a short-cut of advertising-supported software. They are software packages that automatically play advertisements to user computer without desire. The objective of adware is to gain financial profit for their author. Adware are not harmful by nature, but they can be in the form of a pop-up window which can interrupt users thinking. Some adware may come with integrated spyware such as key loggers and other privacy-invasive software [32].

Cookies are some information stored on user's computer by their web browsers. The main purpose of cookie is to authenticate users depending on the information stored in, storing site preferences and server-based session. Cookies are sent as a field in the header of the HTTP response by a web server to a client, and then sent back unchanged by the browser to server in each time when requests sent to the server. Cookies are not executable, because they are text format file only, but may be permanent or not expired on specific date/time. Thus they are not harmful by themselves, but they can be used by other spywares.

Backdoors, also called trap doors, are malcode written into an applications or operating systems with the intention of granting programmers access to the system without requiring them to go through ordinary methods of authentications. They're written by experts or specialized developers for friendly usage. The security problem with trapdoors is the full access, getting in without authentication, because these programs can be used remotely by enemies to make attacks.

Trojan horse is a code that appears to be a useful program, but actually it steals information or corrupts data [11, 32].

Sniffers are computer programs that can intercept and record traffic over a network. They capture each packet to decode and gain raw data, showing the values of various fields in the packet and analyzing its contents. Sniffer code can be used as initial steps toward intrusion attack.

Spam also known as junk email, is a software package that broadcast identical messages to numerous targeted recipients by email. Spam can delay system as numerous mails come, further it can lead to consuming bandwidth. In some cases it is used instead of adware. However in United States, spam was declared to be legal by the CAN-SPAM Act of 2003, provided the message restricted to certain specifications [33].

Botnet is a collection of infected computers (contains bot software embedded in it) that have been taken over by hacker and used to perform malicious functions, without the hackers having to log into the client's computer. Botnet can make DoS attack as many clients' bots, under control of hacker bot, having a role of attack [11, 31].

4.2 Ordinary Malware

Virus is any software code that has the ability to replicate itself, during infection, into any other application software or a document. Viruses can do harmful functions on a user machine; it can make destruction to the whole system. Virus code is attached in an application program using one or more of three methods (pre-pending, embedding and post-pending). This type of malwares use local file system to locate malicious code from infected device to uninfected one [11, 31, 32].

Example: Autorun.inf file which resides in a removable storage media for the purpose of playing the disk automatically. This file is targeted by malware developers, to put their malcode in, instead of the original code. When the removable disk enters, the operating system starts searching for "autorun.inf" and run it. This thing ensures infection inevitably no way. Generally, basic type of viruses can successfully be detectable by signature-based scanners, if signatures are provided.

Worm is any software code that has the ability of self-replicating on victim computer. Worms are independent; they don't need for a host program to start lifecycle. Worm can consume network bandwidth, preventing legitimate users from using it. Worm has the property of creating new copies of themselves to increase the spread rate in a system. AV scanners can make use of this characteristic to detect a malware, i.e. when there are several files of the same attributes, this might be a sign of malware infection [11, 31, 32].

Logic bomb is a software program which remains quiet until a specific condition is met. The most common activator for a logic bomb is a date and time. The logic bomb checks the system date and time, regularly, to see whether it must be activated. If so, the logic bomb activates and executes its code [31]. From the previous discussion, it is obvious that malware those depend on networks dominate the current state of malware infection. Also they can have big impact representing in the disclosure of confidential data, preventing online services and sabotage files. Table 1 summarizes malware classification and their properties.

Table 1. A comparison of major malware families

Factors of comparison	Malware family	Spyware	Adware	Cookies	Trapdoor	Trojan horse	Sniffers	Spam	Botnet	Logic bomb	Worm	Virus
Creation techniques	Pattern	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
	Obfuscated	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
	Polymorphic	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
	Toolkit	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Execution environment	Network	✔	✔	✔	✔	✖	✔	✔	✔	✔	✔	✖
	Remote execution through web	✔	✔	✔	✔	✔	✔	✔	✔	✖	✖	✖
	PC	✖	✖	✖	✖	✖	✖	✖	✖	✔	✔	✔
Propagation media	Network	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
	Removable disks	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
	Internet downloads	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Negative impacts	Breaching confidentiality	✔	✖	✔	✖	✔	✔	✖	✖	✖	✖	✖
	Inconveniencing users	✖	✔	✖	✖	✖	✖	✔	✖	✖	✖	✖
	Denying services	✖	✖	✖	✔	✖	✖	✔	✔	✔	✔	✔
	Data corruption	✖	✖	✖	✔	✖	✖	✔	✔	✔	✖	✔

MALWARE DETECTION

A malware detection program D is the computational function that works in a domain which contains a collection of application programs 'P', and a collection of malicious and benign programs. The detector program 'D' analyzes the programs 'p' which belongs to the set of application programs 'P' to find whether it is a benign (normal program), or a malware (malicious program) [11]. Formally, we can write the above definition as below:

$$\boldsymbol{D(p)} = \begin{cases} \boldsymbol{malicous} & \boldsymbol{if\ p\ contains\ malicous\ code} \\ \boldsymbol{bengin} & \boldsymbol{otherwise} \end{cases}$$

The previous function represents the main function of a malware detection program. The detection program determines the identity of a program by analysis or by identification. But sometimes this function may result in, false positive, false negative or undecidable objects depending on efficiency of the function 'D'. So the function could be rewritten as follows:

$$\boldsymbol{D(p)} = \begin{cases} \boldsymbol{malicous\ if\ p\ contains\ malcode} \\ \boldsymbol{bengin\ \ if\ p\ is\ a\ normal\ program} \\ \boldsymbol{undecidable\ \ if\ D\ fails\ to\ determine\ p} \end{cases}$$

Undecidability is for zero-day malware (new unknown malware), as classification methods fail to determine the identity of a program [34]. False positive is a malware detected while it is not malware and false negative is a benign program detected while it is not benign[3].

4.3 A Brief History of Detection Systems

The first program invented to work as an anti-malware was Flushot Plus by Ross Greenberg in 1987. It was used to prevent viruses and Trojans from making unwarranted changes to files. In 1989 John McAfee released his VirusScan™ program, which could detect and repair several viruses at once [35]. Wisdom & Sense (W&S) was a statistics-based anomaly detector developed in 1989 at the Los Alamos National Laboratory [36]. W&S was rule-based on statistical analysis used with anomaly detection. In 1990, the Time-based Inductive Machine (TIM) used as anomaly detection with inductive learning of sequential user patterns in Common Lisp on a VAX 3500 computer [37]. The Network Security Monitor (NSM) used masking on access matrices for anomaly detection on a Sun-3/50 workstation [38]. The Information Security Officer's Assistant (ISOA) was a prototype that deployed variety of strategies including statistics, a profile checker, and an expert system [39].

4.4 Mechanism of Malware Detection

Software companies develop detection systems products at laboratories and keep track of new programs, analyzing them, putting the valid software in white list and malicious software in black list. For the undecidable software, so-called gray list, the scanners operate them in a controlled environment for more classification [3].

When analysis of a program in the gray list results in new malware, company releases online updates for new malicious software. Then users can update their product databases by using remote access through Internet connection.

4.5 Detection Techniques

4.5.1 Signature-based and anomaly-based techniques

All malware scanners, basically, use signature-based and anomaly-based techniques for detecting identities of programs. However, there are methods using these techniques: dynamic methods that use run-time information of a malware, when it is executed in a memory; static methods those are done by extracting features from static malware, when it is in a disk, and hybrid methods that use combination of dynamic and static methods[32].

To identify maliciousness of a file using signature-based techniques, scanner software evaluates its information to a vocabulary of virus signatures in a database to see whether a signature found there. The advantage of such techniques is its

effectiveness. But the main disadvantage with signature-based techniques is that they cannot defend against unknown malware [3] .

Anomaly-based systems detect any kind of misusing computer that fall out of the ordinary activity of a computer system, while signature-based systems detect malwares that have a fingerprint in their databases [37, 40]. Anomaly-based detect computer malicious software by monitoring system activities and classifying it as either normal or anomalous .The pivotal difference between signature-based and anomaly-based is using classification to detect a malware, instead of using patterns [41].

4.5.2 Heuristic based techniques

Artificial intelligence (AI) was used with signature-based and anomaly-based techniques to enhance their efficiency. Neural networks(NNs) have been adopted for their adaptability to environmental changes and their ability of prediction [42]. Fuzzy logic is an artificial intelligence approach derived from fuzzy theory, which use approximation for logic rather than precise classical logic. Genetic algorithm is another machine learning-based technique used in malware detection process for deriving classification rules and selecting appropriate features or optimal parameters for optimum solution. It applies principles of evolutionary biology such as inheritance, mutation, selection and combination. The main advantage of this technique is the derivation of solutions from multiple directions with no need for prior knowledge about system behavior [43].

Statistical and mathematical techniques are used in malware detection by applying statistical and mathematical models on the information of system activities such as network connections, bandwidth, memory usage, system call used by objects etc. [7, 42].

4.6 Malware Detection Technologies

Host-based intrusion detection systems monitor dynamic behavior and state of specific computer system to see if there are any internal or external activities defraud the system policy. This kind of malware detection systems idiomatically named (“in-the-box”) because they reside in the same host that they are monitoring [13].

Network-based intrusion detection systems (IDS) are used to sniff all the packets on network nodes for analysis. In this type a single sniffer module placed in each network segment to monitor traffic in that segment. In contrast distributed-network-based intrusion detection system has multiple modules placed in each node to monitor traffic in those nodes [44]. Network-based malware detection systems idiomatically named (“out-of-the-box”) because they reside outside the host that they are monitoring [19].

There are hybrid intrusion detection systems; used with mixture of host-based and network-based capabilities. This type of IDS consists of multiple subsystems locating on separate nodes in the network for monitoring and gathering data from these nodes. The data collected by these subsystems is sent to the main system for analysis and classification [45]. Regarding effectiveness issue both host-based and network-based detection systems have their drawbacks, while host-based protects effectively internal system but it is susceptible to external attack, network-based can prevent external attack but it can’t protect inside host [45].

According to [19] a virtual machine (VM) is defined as an efficient isolated duplicate of real machine with characteristics of conformity with the original system, efficiency and full control of system resources. Virtual machine-based malware detection systems are constructed on the basis of the mentioned concept. There exist three classes of VM used by malware detection systems; Sandbox is the first one where computer resources have to be reached through specific API provided by the VM where system receives information of a suspicious executable program from a user, analyzes its behavior by performing it in a controlled environment (sandbox) and sends analysis reports back to the user who has issued the information. Secondly emulation where simulating the entire computer system for running the guest operating system and the VMM provides an execution environment for programs that are identical to the original machine with exception of differences caused by the availability of system resources or by timing dependencies while efficiency is the core characteristic of emulators. An emulator is a piece of software that acts as a hardware (i.e. CPU emulator simulates CPU functionality using software). The emulator does not directly execute a code; instead instructions are intercepted by the emulator, translated into corresponding sequence of instructions compatible with the targeted platform. System emulators are hidden to detection code so it is regarded as a suitable environment for malware analysis. Thirdly, in native system virtual machines, a virtual machine monitor (VMM) is a smaller piece of privileged code that privileging VM on the host computer. This characteristic makes it native VM with good performance, but liable to errors and tamper resistance [14].

Agent–based intrusion and malware detection systems depend on characteristics of agent technology such as autonomy, decentralization, platform independency, scalability and mobility. It is benefitting from the notion of no central station causes no central point of failure [21-23, 25, 46-49].

While the design of host-based IDS and distributed IDS suffer from a number of drawbacks that host-based IDS cannot detect outsider attacks but it is effective internally, the distributed IDS does not take care of internal attack but it is effective externally and agent-based system invented to combine characteristics of both host-based and distributed IDS [22].

Web-based scanning provided by vendors those maintain websites with detection capabilities for scanning the entire local computer systems, critical areas only, local disks, folders or files. Online scanning is good idea for those who don't want to run antivirus applications on their computers. Sometimes malicious software firstly attacks and disables any existing antivirus software then starts attacking. Turning to an online resource that isn't already installed on the infected computer could be reasonable solution [50].

Application protocol-based intrusion detection system (APIDS) is an intrusion detection system that focuses its monitoring and analysis on a specific application protocol. The system monitors the dynamic behavior and state of the application protocol. The system consists of a service or an agent that sits between group of servers, monitoring and analyzing the application protocol between them. A typical place for an APIDS would be between a web server and the database management system, monitoring the SQL protocol specific to the middleware/business logic when it interacts with the database . Anti-Spam systems they are used to prevent e-mail spam. Both end users and administrators have roles in treating spam, rather than embedded techniques used automatically by email server systems. Anti-spam techniques can be classified into four categories: those that require

actions by end-users, those that can be managed by e-mail administrators, those that can be automated by e-mail senders and those deployed by researchers and law enforcement officials [33].

Multi-agent P2P intrusion detection is an agent-based service-oriented system which puts in use of distributed security policy and distributed intrusion detection, on architecture that provides interactive environment to make a decision [21].

Special tools for virus removal are available to help remove stubborn infections or certain types of infection. Examples do include Trend Micro's Rootkit Buster and rkhunter tool to scan for rootkits on an Ubuntu Linux computer.

From the previous discussion, it is clear that the malware detection systems have evolved widely in the past few decades. They have evolved from static programs that work on static data analysis and regular algorithms to complex algorithms that work on sophisticated techniques based on statistical and mathematical models and artificial intelligence. Furthermore, the addition of technological solutions such as the use of cloud computing, virtual machines, network-based application and agent-based technology. Table 2 illustrates some of the malware detection systems.

Table 2. Summary of malware detection systems

Release title for malware detection system	Adopted Technique/ Technology	Characteristics	Advantages	Disadvantages	Release year	Reference no
Efficient signature based malware detection on mobile devices	Signature-based	Uses heuristic techniques	Effective in detecting known malware	Can't detect new malware	2008	[15]
Anomaly-based network intrusion detection: Techniques, systems and challenges	Anomaly-based	They are behaviour analysis, statistical analysis or AI analysis techniques	Detecting unknown new malware	High rate of false/negative and false/positive	2009	[40]
A specification based intrusion detection framework for mobile phones	Specification-based	Utilizing keypad or touch screen interrupts to differentiate between malware and human activity	Detecting unknown new malware	High rate of false/negative and false/positive	2011	[51]
Adaptive Rule-Based Malware Detection Employing Learning Classifier Systems: A Proof of Concept	Rule-based	Combination of a rule-based expert system with an evolutionary algorithm based reinforcement learning	Detecting unknown new malware	High rate of false/negative and false/positive	2011	[52]
A Heuristic Approach for Detection of Obfuscated Malware	Heuristic based	Series of static check on binary file's PE structure for common traces of obfuscation	Detecting Obfuscated new malware	Not all legitimate applications free PE structures.	2009	[17]
Combining file content and file relations for cloud based malware detection	Cloud-based application	Parametric component for file content information and a non-parametric component for file relation information.	Servers have fast response to users requests.	Feature integration degrades information quality.	2011	[8]
Intrusion Detection System (IDS): case study	Host-based	Monitoring internal or external activity that defrauds the system policy	Protect effectively internal system	Susceptible to external attack	2011	[53]
A Network Based Approach to Intrusion Detection and Prevention	Network-based	A sniffer in each network segment to sniff all packets	Prevent external attack	Can't protect inside host	2009	[18]
The use of distributed network-based IDS systems in detection of evasion attacks	Distributed hybrid	Multiple sniffer in each node	Protect effectively internal and external system	Lack of adaptation and scalability	2005	[45]
A virtual machine introspection based architecture for intrusion detection	Virtual machine	Smaller piece of privileged code	Performant	Liable to errors and tamper resistance	2003	[19]
Multi-Agent System for Intrusion Detection in MANET	Agent-based	Combine characteristics of both host-based and distributed IDS	Performant	-	2012	[47]

5. CONCLUSION

This paper presented a detailed review of the state of the art for malware, malware detection techniques and technologies. In particular, it provides an up-to-date comparative study for most of malware families as well as it summarizes a number of malware detection systems. Although the developing processes of malware and their detection systems are rapidly growing, this study can be considered as a key reference for the developers in the field.

6. REFERENCES

[1] Mcafee and Lab, 2013 Threats Predictions. 2013.

[2] Berkenkopf, R.B.S., G-Data Malware Report. 2010.

[3] Ye, Y., et al., Intelligent file scoring system for malware detection from the gray list, in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. 2009, ACM: Paris, France. p. 1385-1394.

[4] Rieck, K., Malheur A novel tool for malware analysis 2012.

[5] Pinz, C.I., et al., Improving the security level of the FUSION@ multi-agent architecture. Expert Syst. Appl., 2012. 39(8): p. 7536-7545.

[6] Ammar Ahmed E. Elhadi, M.A. Maarof, and A.H. Osman, Malware Detection Based on Hybrid Signature

Behaviour Application Programming Interface Call Graph. American Journal of Applied Sciences, 2012. 9 (3): p. 283-288.

[7] Kevadia Kaushal, P.S., Nilesh Prajapati, Metamorphic Malware Detection Using Statistical Analysis. International Journal of Soft Computing and Engineering (IJSCE), 2012. 2(3).

[8] Yanfang Ye, T.L., Shenghuo Zhu,Weiwei Zhuang,Egemen Tas,Umesh Gupta,Melih Abdulhayoglu, Combining file content and file relations for cloud based malware detection, in Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. 2011, ACM: San Diego, California, USA. p. 222-230.

[9] Christodorescu, M., et al., Semantics-Aware Malware Detection, in Proceedings of the 2005 IEEE Symposium on Security and Privacy. 2005, IEEE Computer Society. p. 32-46.

[10]Yin, H., et al., Panorama: capturing system-wide information flow for malware detection and analysis, in Proceedings of the 14th ACM conference on Computer and communications security. 2007, ACM: Alexandria, Virginia, USA. p. 116-127.

[11] Vinod, P., et al., Survey on Malware Detection Methods. 2009.

[12] Zeltser, L., what is cloud Anti-Virus and how it does work.

[13]Jiang, X., X. Wang, and D. Xu, Stealthy malware detection through vmm-based "out-of-the-box" semantic view reconstruction, in Proceedings of the 14th ACM conference on Computer and communications security. 2007, ACM: Alexandria, Virginia, USA. p. 128-138.

[14.] Automated dynamic binary analysis. 2007.

[15] Deepak Venugopal, G.H., Efficient signature based malware detection on mobile devices. Mob. Inf. Syst., 2008. 4(1): p. 33-49.

[16] Kolbitsch, C., et al., Effective and efficient malware detection at the end host, in Proceedings of the 18th conference on USENIX security symposium. 2009, USENIX Association: Montreal, Canada. p. 351-366.

[17] Zhou, S.T.a.M., A Heuristic Approach for Detection of Obfuscated Malware,. IEEE, 2009.

[18] Ahmed, M., et al. NIDS: A Network Based Approach to Intrusion Detection and Prevention. in Computer Science and Information Technology - Spring Conference, 2009. IACSITSC '09. International Association of. 2009.

[19] Garfinkel, T. and M. Rosenblum, A virtual machine introspection based architecture for intrusion detection. 2003: p. 191--206.

[20] Lagar-Cavilla, H.A., Flexible Computing with Virtual Machines. 2009.

[21] Gorodetsky, V., et al., Multi-agent Peer-to-Peer Intrusion Detection

Computer Network Security, V. Gorodetsky, I. Kotenko, and V.A. Skormin, Editors. 2007, Springer Berlin Heidelberg. p. 260-271.

[22] Ye, D., An Agent-Based Framework for Distributed Intrusion Detections. 2009.

[23] Ou, C.-M. and C.R. Ou, Agent-Based immunity for computer virus: abstraction from dendritic cell algorithm with danger theory, in Proceedings of the 5th international conference on Advances in Grid and Pervasive Computing. 2010, Springer-Verlag: Hualien, Taiwan. p. 670-678.

[24] Bijani, S. and D. Robertson, Intrusion detection in open peer-to-peer multi-agent systems, in Proceedings of the 5th international conference on Autonomous infrastructure, management, and security: managing the dynamics of networks and services. 2011, Springer-Verlag: Nancy, France. p. 177-180.

[25] Dong, H., et al. Research on adaptive distributed intrusion detection system model based on Multi-Agent. in Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference on. 2011.

[26] Ou, C.M., Multiagent-based computer virus detection systems: abstraction from dendritic cell algorithm with danger theory. Springerlink, 2011.

[27] Paritosh Das, R.N., A Temporal Logic Based Approach to Multi-Agent Intrusion Detection and Prevention. 2012.

28. McGraw, G. and G. Morrisett, Attacking Malicious Code: A Report to the Infosec Research Council. IEEE Softw., 2000. 17(5): p. 33-41.

[29] Xufang, L., P.K.K. Loh, and F. Tan. Mechanisms of Polymorphic and Metamorphic Viruses. in Intelligence and Security Informatics Conference (EISIC), 2011 European. 2011.

[30] EroCarrera. and P. Silberman, STATE OF MALWARE: FAMILY TIES. 2010.

[31] Egele, M., et al., A survey on automated dynamic malware-analysis techniques and tools. ACM Comput. Surv., 2008. 44(2): p. 1-42.

[32] Idika, N. and A.P. Mathur., A Survey of Malware Detection Techniques. 2007.

[33] Goldman, E., Dissecting Spam's Purported Harms. 2003.

[34] Webster, M., Algebraic specification of computer viruses and their environments. Selected Papers from the First Conference on Algebra and Coalgebra in Computer Science Young Researchers Workshop (CALCO-jnr 2005), 2005.

[35] Grimes, R.A., Malicious Mobile Code: Virus Protection for Windows. O'Reilly Media, 2001.

[36] Vaccaro, H.S. and G.E. Liepins. Detection of anomalous computer session activity. in Security and Privacy, 1989. Proceedings., 1989 IEEE Symposium on. 1989.

[37] Teng, H.S., K. Chen, and S.C. Lu. Adaptive real-time anomaly detection using inductively generated sequential patterns. in Research in Security and Privacy, 1990. Proceedings., 1990 IEEE Computer Society Symposium on. 1990.

[38] Heberlein, L.T., et al. A network security monitor. in Research in Security and Privacy, 1990. Proceedings., 1990 IEEE Computer Society Symposium on. 1990.

[39] Winkler, J.R., A Unix Prototype for Intrusion and Anomaly Detection in Secure Networks (1990). Proceeding. 13 th National Computer Security Conference, 1990.

[40] P. García-Teodoro, J.D.-V., G. Maciá-Fernández, E. Vázquez, Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security, 2009 28.

[41] Bolzoni, D. and S. Etalle, APHRODITE: an Anomaly-based Architecture for False Positive Reduction. 2006, Centre for Telematics and Information Technology, University of Twente: Enschede.

[42] Chandola, V., A. Banerjee, and V. Kumar, Anomaly detection: A survey. ACM Comput. Surv., 2009. 41(3): p. 1-58.

[43] Professor, S.M.B.a.R.B.V.a.A.P.a.A., Fuzzy Data Mining And Genetic Algorithms Applied To Intrusion Detection. In Proceedings of the National Information Systems Security Conference (NISSC), 2000.

[44] Kozushko, H., Intrusion Detection: Host-Based and Network-Based Intrusion Detection Systems. 2003.

[45] Basicevic, F., M. Popovic, and V. Kovacevic. The use of distributed network-based IDS systems in detection of evasion attacks. in Telecommunications, 2005. advanced industrial conference on telecommunications/service assurance with partial and intermittent resources conference/e-learning on telecommunications workshop. aict/sapir/elete 2005. proceedings. 2005.

[46] Gou, X., W. Jin, and D. Zhao, Multi-agent system for Worm Detection and Containment in Metropolitan Area Networks. Journal of Electronics (China), 2006. 23(2): p. 259-265.

[47] Mechtri, L., F.D. Tolba, and S. Ghanemi. MASID: Multi-Agent System for Intrusion Detection in MANET. in Information Technology: New Generations (ITNG), 2012 Ninth International Conference on. 2012.

[48] Silva, M., D. Lopes, and Z. Abdelouahab, A Remote IDS Based on Multi-Agent Systems, Web Services and MDA, in Proceedings of the International Conference on Software Engineering Advances. 2006, IEEE Computer Society. p. 64.

[49] Pinz, C.I., et al., Real-time CBR-agent with a mixture of experts in the reuse stage to classify and detect DoS attacks. Appl. Soft Comput., 2011. 11(7): p. 4384-4398.

[50] Steroids, S.o., Malware online scanners. accessed 12/4/2013.

[51] Chaugule, A., Z. Xu, and S. Zhu, A specification based intrusion detection framework for mobile phones, in Proceedings of the 9th international conference on Applied cryptography and network security. 2011, Springer-Verlag: Nerja, Spain. p. 19-37.

[52] Blount, J.J., D.R. Tauritz, and S.A. Mulder. Adaptive Rule-Based Malware Detection Employing Learning Classifier Systems: A Proof of Concept. in Computer Software and Applications Conference Workshops (COMPSACW), 2011 IEEE 35th Annual. 2011.

[53] Asmaa S. Ashoor, S.G., Intrusion Detection System (IDS): case study. 2011 International Conference on Advanced Materials Engineering IPCSIT, 2011.