Basic survey on Malware Analysis, Tools and Techniques

Dolly Uppal[1], Vishakha Mehra[2] and Vinod Verma[3]

[1,2] Department of Computer Engineering, Rajasthan Technical University, Kota
[3] Asst. Prof., Department of Mechanical Engineering, Govt. Engineering College, Ajmer

Abstract

The term malware stands for malicious software. It is a program installed on a system without the knowledge of owner of the system. It is basically installed by the third party with the intention to steal some private data from the system or simply just to play pranks. This in turn threatens the computer's security, wherein computer are used by one's in day-to-day life as to deal with various necessities like education, communication, hospitals, banking, entertainment etc. Different traditional techniques are used to detect and defend these malwares like Antivirus Scanner (AVS), firewalls, etc. But today malware writers are one step forward towards then Malware detectors. Day-by-day they write new malwares, which become a great challenge for malware detectors. This paper focuses on basis study of malwares and various detection techniques which can be used to detect malwares.

Keywords

Malware, obfuscation, normalization, Deobfuscation, oligomorphic etc.

Introduction

In present century malware attackers are playing a good game over malware defenders. Malware defenders come across through thousands of new malware samples every day. Malcodes are distributed over internet through untrusted websites at an alarming rate. Often malware enters into the system through the downloaded file. Once the malware enters the system it performs malware activity and corrupt the entire system.

Some of the malware can easily be detected and defended through antivirus scanners (AVS). But, today, the packers pack the malware in such a way that it plays hide and seek with the AVS and malware wins the game. So, it becomes a tuff job for the AVS to detect the malware. If the detector detects a malware in a non-infected file, it is considered as false positive. Also, if the scanner fails to detect malware in an infected file it is considered as false negative. A hit ratio is considered if the scanner detects the malware in an infected file.

This paper is a survey study of basics of malware. The paper is organized as follows: Section I describes the classification of malware. Section II briefly explains the techniques to detect malware following with section III about analysis tools of malware. Section IV describes the categories of malicious software. Section V include various obfuscation techniques with section

DOI:10.5121/ijcsa.2014.4110

VI highlights various deobfuscation techniques. Section VII gives brief conclusion about the paper.

I. Classification of Malware

Malware classification	Types of malware	Feature	Mode of operation	Damage caused
The contagious threat	Virus[5]	A form of malware that takes unauthorized control of the infected computer and cause harm without the knowledge of the user	Viruses attach themselves to a program such as executable file and its self-replicating capability spread the infection from one computer to another.	Cause denial of service Performance degradation
	Worms	Worms are standalone malicious software that can operate independently and don't hook itself to propagate	They exploit the security vulnerability by using computer [2] or network resources and spread themselves via storage devices such as USB devices, communication media such as E-mail	Cause network performance issues Consume large amount of memory of systems resources
The masked threat	Trojan	Malignant piece of software that conceal itself and behaves as a legitimate program to takes unauthorized control of the computer.	Trojan does not self-replicate instead downloaded through user interaction such as downloading a file from the internet.	Steal password or login details Electronic money theft Modify/delete files Monitor user activity
	Rootkit	Rootkits are the masking techniques for malware, basically designed to conceal the malicious	Can be installed through a software exploit or by a	Steal password Install Keyloggers

		intent of the program from the antivirus removal programs	Trojan	
The financial threat	Spyware	A software negatively affect a system by keeping track of user's activity without their consent and send back the sensitive information to creator	Can be installed with other software such as freeware or dropped by Trojans	Some sophisticated type of spyware captures entire network interface, digital certificate, encryption keys and other sensitive information.
	Keyloggers	Serious form of Spyware secretly record keystrokes, read cookies and files on the drive to gather personal details	Can be installed by another malicious program or when a user visited a infected site	Capture sensitive information such as username, password, credit card number or online banking details

II. Malware Detection Techniques

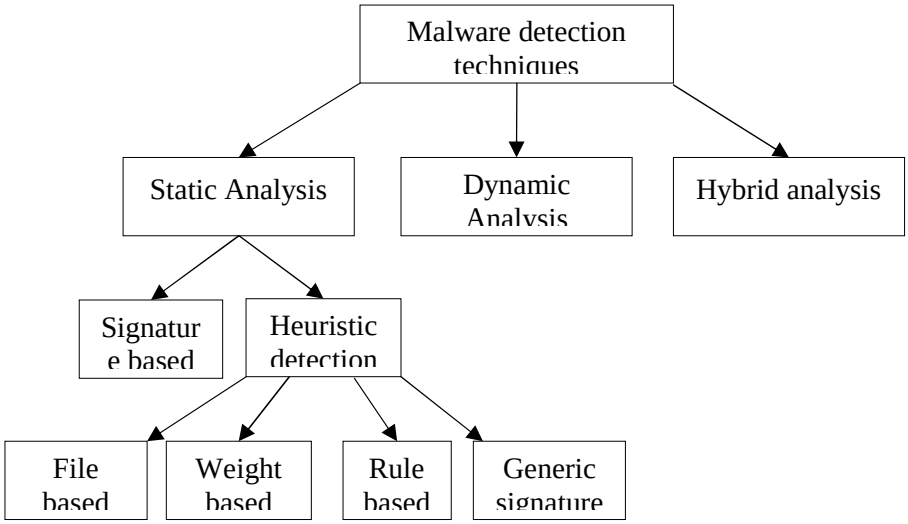

Figure 1. Hierarchal representation of various malware detection techniques

A. Static analysis detection technique

It is the procedure of analyzing software without executing it. During static analysis [9] the application is break down by using reverse engineering tools and techniques, so as to re-build the source code and algorithm that the application has created. Static analysis can be done through program analyzer, debugger and disassembler. Various static analysis techniques are as follows:

1. Signature based detection technique

This technique is also known as pattern matching or string or mask or fingerprinting technique. A signature is a bit of sequence injected in the application program by malware writers, which uniquely identifies a particular malware. To detect a malware in the code, the malware detector search for a previously specified signature in the code.

2. Heuristic detection technique

This technique is also known as proactive technique [4]. This technique is similar to signature based technique, with a difference that instead of searching for a particular signature in the code, the malware detector now searches for the commands or instructions that are not present in the application program. The result is that, here it becomes easy to detect new variants of malware that had not yet been discovered. Different heuristic analysis techniques are:

2.1 File based heuristic analysis

Also known as file analysis. In this technique, the file is analyzed deeply like the contents, purpose, destination, working of file. If the file contains commands to delete or harm other file, than it is considered as malicious.

2.2Weight based heuristic analysis

It is the much ancient technique. Each application is weighted according to the danger it may possess. If the weighted value exceeds the predefined threshold value, then the application contains malicious code.

2.3 Rule based heuristic analysis

The analyzer, here, extracts the rules defining the application. These rules are then matched with the previously defines rules. If the rules are mismatched, then the application contains malware.

2.4 Generic signature analysis

In this signature, variants of malware are detected. A variant of malware means, the malware are different in behavior but belong to same family like “identical twins”. This technique uses previously defined antivirus definition, to discover new variants of malware.

Advantage of Static Analysis

Static analysis is fast and safe; also it gathers the structure of code of program under inspection. If static analysis can calculate the malicious behavior in the application then this information can then be used for future security mechanism.

Disadvantage of Static Analysis

Static analysis does not take stand for analyzing the unknown malware. Also, the source code of many applications is not easily available. For doing static analysis, researchers must have a good knowledge of assembly language and also should have a deep understanding of functioning of operating system.

B. Dynamic analysis detection technique

The process of analyzing the behavior or the actions performed by the application while it is executing is called dynamic analysis [7]. Dynamic analysis can be done through monitoring function calls, tracking the information flow, analyzing function parameters and tracing the instructions. Generally a virtual machine or sandbox is used for this analysis; the doubted application is usually run into a virtual environment. If the application behaves unusually it is categorized as malicious. Nowadays, there are behavioral blocking software, which blocks malicious action of the program before their attack

Advantage of Dynamic Analysis

One can easily detect the unknown malware by simply analyzing the behavior of the application.

Disadvantage of Dynamic Analysis

This analysis takes time as the executing time of the application, so in some cases, it is not fast neither safe. Also, this analysis does not stand for the application which shows the different

behavioral changes by different triggering conditions. In short, it fails to detect multipath malware.

C. Hybrid analysis detection technique

This technique is the combination of both static analysis and dynamic analysis [6]. The procedure it follows it that it first checks for any malware signature if present in the code under inspection and then it monitors the behavior of the code. Hence this technique combines the advantages of both the above techniques.

III. Some static analysis and dynamic analysis tools

Dynamic Analysis Tools of Malware	Description
Process Explorer	Monitor currently running process
FileMon	Monitor file operation
RegMon	Monitor operation on registry
RegShot	Takes snapshot of the registry and associated files
TCPView	Displays all TCP and UDP open connections and the process that opened and is using the port
TDIMon	Logs network connectivity, but does not log packet contents
Ethereal	Packet Scanner that captures packets and supports the viewing of contents/payload

Table 1: Brief overview of some dynamic analysis tools of malware

Static Analysis Toolsof Malware [8]	Description
Bin Text	Extracts strings from executables, reveal registry keys and IRC,SMTP commands stored in string format
IDA Pro	Disassembles executables into assembly instructions
UPX	Executable packer used by malware writers
Proc Dump	Dumps code from memory
OllyDbg	Debugger that enables the user to attach to a process and insert breakpoints

Table 2: Brief overview of some static analysis tools of malware

IV. Categories of Malicious software

Encrypted Malware

This approach uses the concept of encryption to prevent signature based Antivirus Scanners [10]. This approach comprises of two main parts of an encrypted malware- encrypted main body and decryptor. Each encrypted malware is made different from its signature by using different key every time. The disadvantage to this approach is that the malware can be easily detected by antivirus scanners as the decryptor consists of same code pattern.

Oligomorphic Malware

The most advancement to encrypted malware is oligomorphic malware. The malware authors changes the decryptorevery time, hence generating thousands of new malware. But still, it can be detected by its signature, as decryptor can replicate itself finite number of times.

Polymorphic Malware

The polymorphic malware is same as oligomorphic malware, with a difference that, it generates infinite number of decryptor by using different obfuscation techniques. The basic function of polymorphic malware remains the same each time it is decrypted, only the code changes. The toolkit called Mutation Engine (ME) is used in order to change non-obfuscated code to polymorphic code. One part of the code remains the same with each iteration, hence it can be easily detected by Antivirus Scanner.

Metamorphic Malware

Metamorphic malware [1] is written again every time with the each iteration, which makes each new malware different from its previous once. They can easily pass through AVS. As AVS scanner are unable to match any signature henceforth. Metamorphic malware are not packed malware, hence, they never leave a signature in the memory to be matched. Authors generate then by enhancing different obfuscation techniques thoroughly and strictly.

V. Obfuscation technique

Obfuscation is the technique, generally used by malware authors, to make source code harder to read, understand and reverse engine, and to conceal the malicious intent of the malware [3].

Figure2. Obfuscation

A' contains malicious code which is difficult to reverse engine, but it holds functionality and performs comparable to A.

Basically six techniques are used for Code Obfuscation

Dead-Code-Insertion– It is the simplest form of code obfuscation technique which can be done by inserting NOPs (No Operation Performed) or some push statement followed by pop statement in the code.

Subroutine Reordering- As the name suggests, this technique randomly changes the order of subroutines in the program, hence generates K' different malwares, where K is the number of subroutines.

Code Transposition - In this, the order of instruction is changed by using statements like jmp and unconditional branch statements, which makes the code different from its naive code. Code transposition can be done in two ways. The first method is as described above. The second one generate the new variants by reordering the independent instructions, which is difficult to implement and harder to identify the malware.

Instruction Substitution – This technique replaces some of the code statements with the equivalent statements. For example MOV with either Push or Pop.

Code Integration – A new brief is inserted into the source code of the program in order to make the code malicious.

Register reassignment – The registers of the code is replaced by the unused registers. The program code and its behavior remains the same.

VI. Deobfuscation Techniques

Deobfuscation is the technique, generally used by malware detectors to make malicious or packed code easier to read, understand and reverse engine and to conceal the malicious intent of the malware.

Malware Normalization

It is the procedure of eliminating the obfuscated code from the program to enhance the capability of the malware detector. During this process, the malware goes through the normalizer and then it is matched with its ancestors present in the database. If it is matched, then it becomes a new signature and then stored in the database.

The PE code is passed through the decompression software, through which decompressed code is obtained, the code is then passed to the disassembler and disassembled code is passes through the normalizer. The code so obtained is the normalized code, which is send to the malware detector. The detector extracts the features of the malware and then is compared to the signatures of known malware present in signature repository. The comparison is done through comparison engine using sequence alignment algorithm. The possibility is that of matching maximum to maximum signature present in the repository to that of normalized code obtained. If the signature is not matched with the signature repository, then it is categorized as new signature and stored in repository.

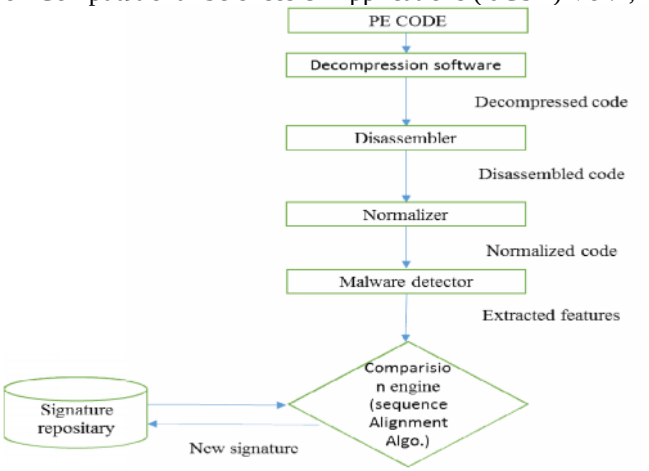

Figure 4. Deobfuscation of malware using Normalization

Similarity Analysis

It is the procedure of analyzing whether the code under inspection is the variant of same malware or not. Similarity analysis is done to capture a polymorphic malware by analyzing API sequence. The PE code of the program is passed to the decompressor, the obtained decompressed code is then sent to PE parser. From this binary parser, sequence of API calls is obtained, which is then used for similarity analysis. Also signature from database is used for similarity analysis. This analysis generates a value which is compared against the threshold value. If the obtained value is greater than the threshold value, then it is considered as a malware otherwise benign.

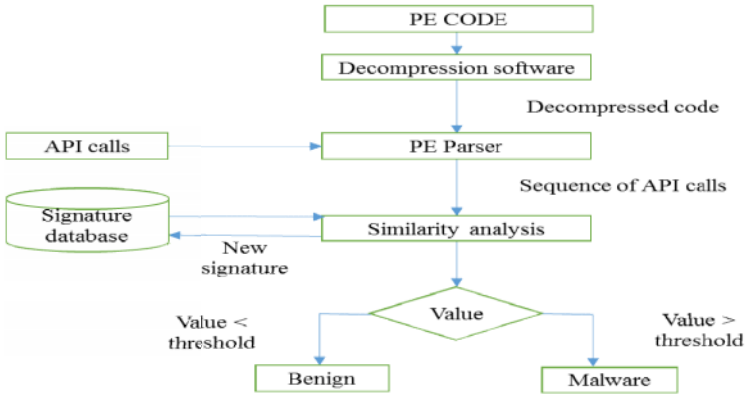

Figure 5. Measures of similarity analysis

CONCLUSION

In this paper we had surveyed a study about various types of malware and categories of malicious software. In particular, a light has been thrown on various obfuscation and deobfuscation techniques. Although the rate hazards of new malware are increasing at an alarming rate, this

paper provides a thorough study of tools for analyzing malware with a clear understanding of various countermeasures need to be adopted.

REFERENCES

[1] ArunLakhotia ,AdityaKapoor , Eric Uday , "Are Metamorphic Viruses Really Invincible ? Part 2" , Virus Bulletin, January 2005.

[2] Robin Sharp, An Introduction to Malware, Spring 2012. Retrieved on April, 10, 2013 http://orbit.dtu.dk/fedora/objects/orbit:82364/datastreams/file_4918204/content

[3] A. H. Sung, J. Xu, P. Chavez and S. Mukkamala: Static Analyzer of Vicious Executables (SAVE), Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC‟04), IEEE.J.Rabek, R.Khazan, S.Lewandowski and R.Cunningham. Detection of injected, dynamically generated, and obfuscated malicious code. In Proceedings of the 2003 ACM Workshop on Rapid Malcode, pages 76–82, 2003.

[4] G. McGraw and G. Morrisett. Attacking malicious code: A report to the infosec research council. IEEE Software, 17(5):33–44, 2000

[5] Muazzam AhmedSiddiqui:Data Mining Methods for Malware Detection: University of Central Florida, 2008.

[6] Robiah Y, SitiRahayu S., MohdZaki M, Shahrin S., Faizal M. A., Marliza R. "A New Generic Taxonomy on Hybrid Malware Detection Technique " (IJCSIS) International Journal of Computer Science and Information Security, Vol. 5, No. 1, 2009

[7] SavanGadhiya,KaushalBhavshar "Techniques for Malware Analysis" Volume 3, Issue 4, April 2013 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering

[8] Vasudevan, A., &Yerraballi, R., "SPiKE: Engineering Malware Analysis Tools using Unobtrusive Binary-Instrumentation". Australasian Computer Science Conference (ACSC 2006),2006

[9] Bergeron, J., Debbabi, M., Desharnais, J., M., E., M., Lavoie, Y., &Tawbi, N. (2001). Static Detection of Malicious Code in executables programs. International Journal of Req Engineering

[10] Mohammad NourSaffaf: Malware Analysis Bachelor‟s Thesis., Helsinki Metropolia University of Applied Sciences, May 27, 2009

Authors

Dolly Uppal

Scholar of Masters of Technology from Rajasthan Technical University, Kota, Rajasthan, India.

Passed out B.Tech.From Rajasthan Technical University, Kota, Rajasthan, India in 2012.

Research area: Malware Analysis

VishakhaMehra

Scholar of Masters of Technology from Rajasthan Technical University, Kota, Rajasthan, India.

Passed out B.Tech.From Rajasthan Technical University, Kota, Rajasthan, India in 2011.

Research area: Malware Analysis