ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

BASIC: Brief Analytical Survey on Metamorphic Code

Shiv Kumar Agarwal[1], Vishal Shrivastava[2]

Research Scholar, Department of Computer Science and Engineering,

Arya College of Engineering and Information Technology, Jaipur, Rajasthan, India[1]

Professor, Department of Computer Science and Engineering,

Arya College of Engineering and Information Technology, Jaipur, Rajasthan, India[2]

Abstract: This paper discusses a variety of obfuscation techniques used by metamorphic malware to change the structure of new variant. Mutation engine is associated with each metamorphic malware which is responsible to make the changes in the structure of new variant using variety of obfuscation techniques. In most of the metamorphic malware, the size of mutation engine remains a little bit small in order to bypass detection. The main objective of malware writers is to make code as complicate as possible along with modern obfuscation techniques. As per our survey, there is not an efficient algorithm to deal with these metamorphic codes. The conventional signature based algorithm used by most of antivirus software is even failed against the metamorphic families.

Keywords: Metamorphic malware, Obfuscation, Polymorphic, Mutation engine.

I. INTRODUCTION

Malware is short for 'malicious software' and is another term for 'computer viruses'. A malware is virus software spread through malicious programs or software known as malware. Malware is designed to delete, block, modify or copy data, or disrupt the performance of computers or computer networks. "Malware" is the general term covering all the different types of threats to your computer safety. The term malware includes viruses, worms, trojan horses, rootkits, spyware, keyloggers and more.

Once malwares enter to the system, they start to find the vulnerabilities within the operating system then perform unintended operation in the system. Most of the malwares basically attack on performance of the system, data integrity and privacy [1]. They also play the major role in denial of service attack [1], [2]. These malwares are also capable to infect other executable files and data. Malwares depending on their behaviour collect the information about host and harm the host computer without consent of the owner.

Today, there are some malware families known as modern malware families, which have the capability to change the signature of new variant in each generation by using a variety of code obfuscation techniques.

This paper introduces the obfuscation techniques commonly used in the polymorphic and metamorphic malware. For this goal, we firstly overview the history of the malwares that have been developed to defeat signature based antivirus scanners. Then, the malware obfuscation techniques are introduced with examples.

This paper is organized as follows. In section 2, we describe the polymorphic and metamorphic malwares. Section 3 explores the obfuscation techniques commonly used by polymorphic and metamorphic malwares, and then section 4 discusses the future trends. Finally, we conclude in section 5.

II. MALWARE TYPES

Malware is a collective term which includes Virus, Worms, Trojan horse with some other malcodes. The behaviour of malcodes and detection methods change over the year. Detection techniques employed by researchers depend on the behaviour and structure of malwares. In this work, we are addressing some of the well known malwares and their behaviour.

A. Viruses

Initially, viruses were developed with the intention to stay in boot sector and floppy disks. So that whenever any infected system starts booting, these viruses get activated and start their execution inside the system. They are required human interaction to spread out from one computer to another computer. Today internet is widely used throughout the world. Therefore, virus writers develop such viruses

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

those are having the capability to take the advantage of internet for their movement from one place to another place. Suppose that any external device is infected through the viruses. When any user uses such external device for the purpose of storing and retrieving some data or information from the device then host computer also gets infected due to malicious nature of viruses [3]. Viruses are also having the ability to reproduce themselves and infecting other programs and data.

B. Trojans

Trojan horse is a malicious program by nature. When user clicks on the link or attachment comes with any email or try to download some data over the internet which appears as per the user interest and looks very familiar to the user then Trojan horse acquire some space inside the user system and starts its execution without user interaction. Even the user of that system does not have the knowledge about such malicious activity which is happening inside his system. The main purpose of Trojan horse is to get some sensitive information from the infected system.

C. Worms

Worm is a malicious program, having the ability to reproduce itself over the network. Worm does not require any human interaction to perform replication. In other way, we may also considered it as a macro which may resides in a Word file and replicate itself in network. This file moves from one to another place and infect all nodes or systems appeared in its path.

D. Spyware

Spyware is a malicious program which continuously monitors the host computer and collects the sensitive information about the user like pages of user interest on the web which are more frequently accessed or visited by the user. Besides that it also collects the sensitive information about the user such as details of credit card number, email password, key pressed by user etc.

E. BotNets

BotNet is remotely controlled software such as robots or bots. They allow an attacker to take the complete control over the infected machine. Bots are centrally controlled by the IRC(Internet Relay Chat) protocol. Bots are generally used to transmit spam/spyware remotely. In general, Bots are connected through a central hub. In this type of configuration to manage the various connections over a single server is very difficult. Therefore, this type of structure cannot be extent at large level. But in case of hierarchical structure, it can be extent at high level. Where Bot master is connected through hundreds of Bots and each Bot is further connected to many more bots.

F. Logic Bombs

Logic Bomb is not having the capability to reproduce itself. Once it installed into the system, it waits for some trigger, incident or any external event like arrival of particular date and time or creation or deletion of any information, prior to perform any damage or any malicious activity.

Signature based detection technique used by these anti-virus software was very popular and successful until malware writers started to write the most advanced malware. There are two categories of malicious software programs (malware) that have the ability to change their code as they propagate.

G. Polymorphic malware

Polymorphic malware is a computer program that reproduces and causes harm to the computer. Polymorphic viruses are an extension of encrypted viruses where the decryption key is different with each virus. But the decrypted virus code is the same in spite of the different decryption key. Antivirus programs that incorporate code emulation techniques can detect polymorphic viruses [4]. Polymorphic malware also makes changes to code to avoid detection. It has two parts, but one part remains the same with each iteration, which makes the malware a little easier to identify.

Polymorphic malware generate different variants of itself while keeping the inherent functionality as same. This is achieved through polymorphic code. It is a style of code that mutates keeping the original algorithm the same [5]. The small section of polymorphic malware code containing the key generator and Encryption-decryption module is responsible for morphing the malware and creating variants that do not have the same signature. The problem of polymorphic malware is that the decryption block remained mostly the same in all variants.

H. Metamorphic malware

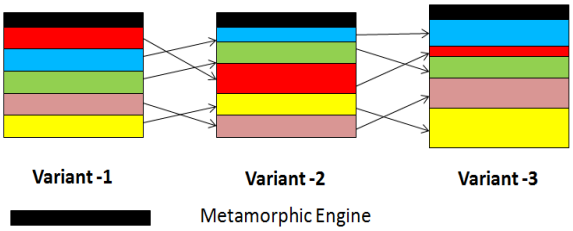
Metamorphic viruses are more powerful than polymorphic viruses. Unlike polymorphic viruses, they do not decrypt to the same virus code. Metamorphic viruses change the structure of their code without affecting the functionality. The changed code is recompiled to create a virus executable that looks different from the original [4]. Such modification is achieved by using several metamorphic techniques.

Unlike, polymorphic malware, metamorphic malware contain a morphing engine. This engine is responsible for obfuscating the whole malware. The body of a metamorphic malware can be broadly divided into two parts namely Morphing engine and malicious code. Code structure entirely different in each variant using obfuscation technique but same in behaviour and functionality as shown in figure1. Mutation engine embedded with malicious code responsible for code obfuscation.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

Figure 1: Different forms of metamorphic malware



III. OBFUSCATION TECHNIQUES

Code obfuscation is used by software vendors to hide their proprietary code, to increase the difficulty for reverse engineering the code. Many malware writers take it as advantage and obfuscate their program using obfuscation transformations so that the malicious intention could not be exposed. The obfuscated code performs comparable to the original program and retains similar functionality [3].

Code obfuscation is a technique used to make code hard so that nobody can understand the logic adopted behind the code [1]. Metamorphic malwares use code obfuscation techniques as opposed to encryption used by polymorphic viruses. The resulting code after obfuscation has the same functionality. In order to avoid antivirus scanners, malwares evolve their body into new generations through the obfuscation technique. This section introduces the obfuscation techniques commonly used in the polymorphic and metamorphic malware.

A. Dead code insertion

Dead-code insertion is a simple technique that adds some ineffective instructions at some random positions to a program to change its appearance, but keep its behaviour same [2] [6] [8]. An example of such instructions is nop which is generally inserted by the mutation engine for obfuscating the original code as shown in the table. However, the signature based antivirus scanners can defeat this technique simply by just deleting the ineffective instructions prior to analysis.

The dead code insertion adds code to the program without changing its functionality. In order to make the detection more complex, following sequences are also added by malware writers in the generated variants [9].

- Combination of push reg and pop reg.
- Combination of inc x and dec x instructions.
- Statements like xor reg, reg and mov 0, reg.

B. Register renaming

Register reassignment is another simple technique that switches registers from generation to generation while keeping the program code and its behaviour same [6] [8]. As shown in Table 1, both the codes are having same functionality but different signature [10]. For example if register ecx is not used in the entire exist range then it could be replaced by register eax. This technique uses different registers for new infections but continues to use the same virus code. W95/Regswap is a virus that uses the register usage exchange technique [11].

Table 1: Register renaming technique

Original Code	Code after Register Renaming Obfuscation
MOV EAX, [X]	MOV ECX, [X]
MOV EBX, [Y]	MOV EAX, [Y]
ADD EAX, EBX	ADD ECX, EAX
MOV [X], EAX	MOV [X], ECX

Semantic based detection technique is more useful in this case. Because all generated variants using this type of obfuscation technique have same semantic. Wildcard searching is also capable to make this technique useless which ignores the register changes.

C. Subroutine Reordering

Subroutine reordering obfuscates an original code by changing the order of its subroutines in a random way [6].This technique can generate n! different variants, where n is the number of subroutines. For example, Win32/Ghost had ten subroutines, leading to 10! = 3628800 different generations [6]. In this technique, the subroutines are reordered and branch instructions are used to connect them to maintain the functionality. The order of subroutines is different for each infection.

D. Instruction Substitution

In this technique some instructions within the program are replaced by the instructions, which are having the same functionality as shown in Table 2. Instruction substitution evolves an original code by replacing some instructions with other equivalent ones [8]. Sometimes, programmers can perform an action in different ways of coding.

Table 2: Instruction substitution technique

Instruction	Equivalent Instruction
mov eax, 18	mov eax 10 add eax 8
mov edx 36	mov edx 48 sub edx 12
mov [ecx+5],	add ecx 2

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

ebx	mov [ecx+3], ebx
mov ebx, 4 add ebx, 3	xor ecx, ecx sub ecx, -7

The instruction substitution transformation replaces some instruction in the program with the instructions which are functionally equivalent. This is the most difficult of all obfuscation class to be de-obfuscated. To handle such type of detection the malware scanner should maintain a dictionary of equivalent instructions. Therefore; this is a great opportunity for the virus programmers to utilize this possibility in metamorphic engines. Win95/Bistro is a virus that uses this technique to transform its code [11].

E. Code Transposition

Code transposition reorders the sequence of the instructions of an original code without having any impact on its behaviour [7]. There are two methods to achieve this technique. The first method randomly shuffles the instructions, and then recovers the original execution order by inserting the unconditional branches or jumps. In this technique conditional or unconditional jump (branch) instruction are inserted in such a way that original functionality is maintained but structure of control flow is changed.

This technique modifies the structure of the program in form of physically reordering of the program codes, while preserving the execution order or flow of the program running using conditional jumps or unconditional branches. It may be done at the level of instructions or modules. The Win95/Zperm virus is a very good example of this technique. Clearly, it is not difficult to defeat this method because the original program can be easily restored by removing the unconditional branches or jumps.

On the other hand, the second method creates new generations by choosing and reordering the independent instructions that have no impact on one another as shown in Table 3. Because it is a complex problem to find the independent instructions, this method is hard to implement, but can make the cost of detection high. With the help of this method, malware writers create the more sophisticated variants of existing virus. Thus, all the variants of same family show same program behaviour but having different code structure.

Table 3: Code transposition technique

Original Code	**Code with Subroutine Permutation**

Function1:
MOV EAX, [X]
Function2:
MOV EBX, [Y]
Function3:
ADD EAX, EBX
MOV [X], EAX
	Function2:
	MOV EBX, [Y]
	Function1:
	MOV EAX, [X]
	Function3:
	ADD EAX, EBX
	MOV [X], EAX

F. Code Integration

Code integration is a sophisticated technique used by metamorphic virus to generate new body structure during each generation. In this technique, the virus first decompiles the executable file, divides the code into different fragments, inserts virus code, and compiles the entire code again to generate new executable code. This makes it hard to detect the virus, and even more difficult to repair the executable [10], [11]. This obfuscation technique is introduced by the Win95/Zmist malware. As one of the most sophisticated obfuscation techniques, code integration can make detection and recovery so difficult.

IV. FUTURE TRENDS

As shown in the advanced malwares such as Zmist, the malware obfuscation technologies have become sophisticated and complex. Clearly, such a tendency is expected to be retained based on the growth of the hardware and software technologies. Also, they will be revised to be suit for the popular infrastructures such as web and smartphone. In this section, we describe the future trends in the malware obfuscation techniques. As per our observation, we required an algorithm with the combination of both static and dynamic analysis in order to deal with these obfuscated malware.

V. CONCLUSION

In this paper, we briefly surveyed the malware obfuscation techniques such as dead-code insertion, register reassignment, subroutine reordering, instruction substitution, code transposition and code integration, which have been mainly used by polymorphic and metamorphic malwares to evade antivirus scanners. As a future trend, these obfuscation techniques will be more sophisticated and

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

complex while being combined with one another. Especially, to handle these obfuscation techniques, we required an algorithm with the combination of both static and dynamic analysis.

ACKNOWLEDGMENT

This research work has been carried out with the support of R&D cell, Department of computer science and engineering, Arya college of engineering and I. T., Jaipur. We thank to all researchers in field of metamorphic malware to get motivated and encouraged to do further work in this field.

REFERENCES

[1] Ilsun You and Kangbin Yim, Malware obfuscation techniques: A brief survey, In Broadband, Wireless Computing, Communication and Applications.

[2] A. Balakrishnan and C. Schulze, Code Obfuscation Literature Survey, http://pages.cs.wisc.edu/~arinib/writeup.pdf, 2005.

[3] P. Vinod, V. Laxmi, M.S. Gaur, GVSS. Phani Kumar, Metamorphic virus detections through static code analysis, URL http://amrita.edu/cyber-workshop/proceedings/icscf09_submission_54.pdf.

[4] W. Wong, Analysis and detection of metamorphic computer viruses http://www.cs.sjsu.edu/faculty/stamp/students/Report.pdf

[5] Polymorphic Code, http://en.wikipedia.org/wiki/Polymorphic_code

[6] W. Wong and M. Stamp, Hunting for Metamorphic Engines, Journal Computer Virology, vol. 2, no. 3, pp. 211-229, Dec. 2006.

[7] M. Christodorescu and S. Jha, Static Analysis of Executables to Detect Malicious Patterns, Proceedings of the 12th conference on USENIX Security Symposium, Vol. 1, pp. 169-186, Aug. 2003.

[8] E. Konstantinou, Metamorphic Virus: Analysis and Detection, RHUL-MA-2008-02, Technical Report of University of London, Jan. 2008. http://www.rhul.ac.uk/mathematics/techreports

[9] Aditya Govindaraju. Exhaustive statistical analysis for detection of metamorphic malware, Master's thesis, 2010. URL http://scholarworks.sjsu.edu/etd_projects/66.

[10] Szor P., & Ferrie, P. (2005). Hunting For Metamorphic. http://www.symantec.com/avcenter/reference/hunting.for.metamorphic.pdf

[11] P. Szor. The Art of Computer Virus Research and Defense, Addison-Wesley Professional, 2005.