International Journal on Emerging Technologies (Special Issue NCETST-2017) **8**(1): 121-123(2017)
(Published by Research Trend, Website: www.researchtrend.net*)*

ISSN No. (Print) : 0975-8364
ISSN No. (Online) : 2249-3255

`The Threat of Obfuscated Malware

Kamlesh Joshi[1], Himanshu Sangrola[1] and Rahul Palaria[2]
[1]B. Tech Scholar, Amrapali Group of Institutes, Haldwani, (Uttarakhand), INDIA
[2]Assistant Professor, Amrapali Group of Institutes, Haldwani, (Uttarakhand), INDIA

ABSTRACT: One of the most serious challenges that the computer industry faces is the threat of malware. Malware is a generic term for any malicious software. Today, malware is a multibillion-dollar industry in itself. Although a number of commercial software's are available in the market for the detection and deletion of malwares, certain malwares remain undetected. The malwares which evade detection by anti malware softwares are mostly encrypted or obfuscated. Thus, there is a potential threat to the security of our computer systems from these types of malware.
In this paper, we discuss the evolution of malwares from simple ones to encrypted ones, polymorphic and metamorphic ones. We also discuss the various obfuscation techniques involved in metamorphic malwares. Towards the end, we have covered some of the existing malware detection techniques.

Keywords: obfuscation, malware detection

I. INTRODUCTION

Malwares are responsible for a majority of crime over the internet and can be considered as a major reason behind the growing number of cyber crimes. Malwares can be classified into viruses, worms, backdoors, Trojans, etc. according to their goals and propagation methods. There are also a number of softwares (anti viruses) available in the market for the detection and removal of these malwares. Malware detection is a core component of these anti virus softwares. Signature-based malware detection is one the techniques employed by these antivirus softwares in which signatures of known malwares are matched to the malware being detected. However, some malwares are able to evade this signature based detection because they are obfuscated. This paper deals with such types of malware.
This paper is divided into six sections. Section 2 covers the evolution of malwares from encrypted ones to polymorphic ones. Section 3 covers the most metamorphic malwares. The various obfuscation techniques are discussed in the next section. Section 5 covers the various metamorphic malware detection techniques.

II. ENCRYPTED AND POLYMORPHIC MALWARES

In earlier days, encryption was used as a technique to obfuscate the content of malware. These encrypted malwares had decryption engine and thus a portion of the malware code was left unencrypted. Thus, this type of malware could be easily be detected by the signature of the unencrypted portion of the code. After this came oligomorphic malwares, which employed multiple decryption algorithms. This made signature based detection difficult [1].
After this, came polymorphic malwares. These encrypted malwares were capable of mutating their decryption engines in each generation [2]. These malwares create variants of themselves using different encryption mechanism in each generation resulting in different decryption engines. Thus, it becomes difficult for the malware detectors to detect the signature of the decryption engines. The body of a polymorphic malware consists of malicious code and encryption-decryption code.

III. METAMORPHIC MALWARES

Metamorphic malwares can create an entirely new variant after reproduction and this new variant produced is in no-way similar to the original variant [3]. These malwares do not make use of encryption unlike polymorphic malwares. Instead, these malwares make use of code obfuscation techniques. Since these malwares have do not produce variants having same body, they can easily evade signature based detection.
Each time any metamorphic malware runs, it changes the opcode loaded into memory and then writes a new version of the malware back to the infected host file. The malware maintains its malicious behavior without ever having the same sequence of native opcodes in memory.
These malwares contain a morphing engine which is responsible for obfuscating the whole malware. The Morphing engine consists of subcomponents namely

Disassembler, Shrinker, Permuter, Expander and Assembler.

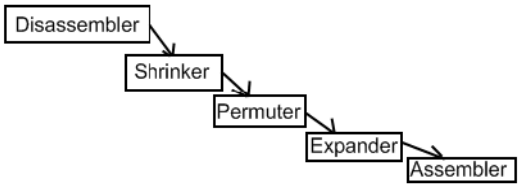


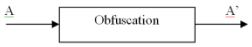
Fig.1. Parts of Metamorphic Engine.

Disassembler converts the machine language to assembly language. Then the code is converted into an equivalent code using various code obfuscation techniques. The obtained code is then shuffled using a permuter. At last, the machine code is generated using an assembler [4]. The new variant is totally different from its parent in appearance but has the same functionality as its parent.

IV. OBUSCATION

Obfuscation is the technique to hide the information such that others cannot find the true meaning. Software vendors make use of obfuscation so that the software would be difficult to reverse engineer. Just as this technique can be used by software vendors to protect software against attackers it can be used by malware writers to hide malicious content.

Consider a piece of code A subjected to obfuscation. The corresponding obfuscated code is A'.



The obfuscated code A' holds the functionality of code A and is difficult to reverse engineer in comparison to code A.

V. OBFUSCATION TECHNIQUES

In the following section, we cover the various malware obfuscation techniques discussed in [4] and [5].

Dead-Code Insertion

This technique involves the addition of extra lines of code at any random position of the program. These lines are ineffective. Though, the addition of these lines change the appearance of the program, the behavior of the program remains the same. Dead Code insertion includes insertion of nop instructions.

Code Transposition

In this technique, the sequence of instructions in the code is reordered without making a change in the functionality of the code. In this technique, the instructions are shuffled in a random manner The next step is to recover the execution order of the code. This is done by making use of jumping instructions. This is done in such a manner that the functionality of the code remains the same even though the control flow is changed.

Subroutine Reordering

In this type of code obfuscation the order in which the subroutines appear in the code is reordered. In such a manner that it does not impact the functionality of the malware. This technique can generate n! different variants, where n is the number of subroutines. The malwares employing this type of code obfuscation can be detected through signature detection.

Register Reassignment

Register reassignment is another simple obfuscation technique in which either the name of the variables or the registers are changed. This results in different opcodes being generated. The detection of such type of malware requires a wild card search algorithm. This algorithm ignores register changes. This technique renaming provides different memory traces for each variant. This makes it difficult for virus detectors.

Instruction Substitution

This technique of code obfuscation is based on the fact that an operation can be done in a number of ways just like in mathematics or Boolean algebra. In this technique, the obfuscated code is generated by replacing some instructions with other equivalent ones in the original code. This technique of code obfuscation can create a totally different variant.

Code Integration

This technique involves the decompilation of the malicious code. Decompilation breaks the program into manageable objects. The same malware is then integrated with some or all these objects. This process is called knitting. After that these infected objects are reintegrated to make a program which looks like the original code. In this way it becomes difficult to detect this type of obfuscation.

Generally, malware writers make use of a combination of different obfuscation techniques to make their malware detection proof. Also, a number of virus generation toolkits are available in the market for the coding and obfuscating malwares.

VI. METAMORPHIC MALWARE DETECTION TECHNIQUES

There are a number of techniques involved in the detection of malwares. In this section, we have covered some of these techniques discussed in [6].

Signature based detection

This is one of the most popular and effective way of detecting malware. In this technique, a database is maintained which holds the signatures of all known malwares. In detecting any malicious code, its signature

is matched with the signatures stored in the database. If the signature of the concerned code matches with any of the signatures stores in the database, an alert is generated.
Though, Signature detection is fast and simple, it is not effective against some polymorphic and most of the metamorphic malwares.

Heuristics based detection
Recent virus detectors use signature detection along with heuristics. This help in reducing the number of false alarms. Heuristics is partially dependent on the behavior of the target malware. Heuristic based detection is good when combined with another reliable detection technique.

Behavioral based detection
Behavior based objects detection is a type of dynamic analysis techniques. In this detection, behavior of the concerned code is analyzed during runtime. If the behavior seems "bad", it is flagged as malware and corrective action is taken. Behavior based detection requires "templates" of bad/suspicious behavior. The behavior of the malware is its signature. Thus behavior based detection technique is a kind of signature based detector except that the signature in this case is the functionality of the malware.

Semantic based detection
This detection technique is based on static analysis. In this, detection does not require the execution of the malware. The malware code is used to determine the malicious nature of the code. A signature is created based on the semantic property of the code. Semantic based detection can detect malwares obfuscated by techniques such as Subroutine Reordering, Register reassignment, dead code insertion. However, this detection cannot be used to detect malwares obfuscated by other techniques.

Hidden Markov Model based detection
Profile Hidden Markov Models (PHMM), known for determining relations between DNA and protein sequences, can also be applied for malware detection. Though, PHMM can detect malwares including the metamorphic ones, they need a test data in order to train them. Also, the process of filtering the data, disassembling them, training and scoring the whole dataset is time consuming.

Similarity Analysis
In this technique, the program executable is decompressed and disassembled. Each disassembled program represents a vector of functions. Each function is represented as an array of vector of functions. The similarity between the functions of a program P and its variant is computed using cosine similarity measure or other methods. This value is then compared with the threshold value to check whether executable is malicious or not.

VII. CONCLUSION

The rat-race between malware writers and anti-malware technologies has made Malware a billion dollar industry in itself. As anti-malware technologies have evolved over the years, the malware writers too changed their tactics accordingly. Obfuscated malware still continues to be threat to the industry as well as the home users. Thus, this problem requires a serious attention from all stakeholders. To address this problem efficiently it is important to design anti- virus systems that detect all obfuscated malwares. It is equally important that this detection is done at a reasonable speed and with precise accuracy.

REFERENCES

[1]. Ilsun You, Kangbin Yim "Malware Obfuscation Techniques: A Brief Survey" 2010 International Conference on Broadband, Wireless Computing, Communication and Applications.
[2]. Shiv Kumar Agarwal, Vishal Shrivastava "BASIC: Brief Analytical Survey on Metamorphic Code" *International Journal of Advanced Research in Computer and Communication Engineering,* Vol. **2**, Issue 9, September 2013
[3]. Vinod P., V. Laxmi, M.S. Gaur "Survey on Malware Detection Methods"
[4]. Mila Dalla Preda, "Code Obfuscation and Malware Detection by Abstract Interpretation" Phd thesis Universit`a di Verona.
[5]. Ashwini Venkatesan" Code Obfuscation and Virus Detection" Master's Theses and Graduate Research San Jose State University.
[6]. Govindaraju, Aditya, "Exhaustive Statistical Analysis for Detection of Metamorphic Malware" (2010).*Master's Projects.* Paper 66.