Metamorphic Virus: Analysis and Detection

IT19029214 | Hapuarachchi J.C

Faculty of Computing
Sri-Lanka Institute of Information Technology (SLIIT)

Secure Software Systems

Abstract- Metamorphic Virus is a virus that can change its code and the signature pattern in each performed iteration. Metamorphic viruses can be identified as one of the biggest threats to digital world and this is said to be more harmful and dangerous than any other typical malware available. Because of the ability to change the code and the signature pattern, metamorphic viruses are hard to get detected by signature-based virus scanners. Detecting Metamorphic virus is a very challenging task. To avoid from getting detected, metamorphic virus uses many different techniques and mechanisms. In this research, an in-depth research of metamorphic virus is presented. Other than an in-depth analysis of the mechanisms that metamorphic viruses use to avoid detection is also presented. In this research before going directly to metamorphic virus an introduction of general viruses is given first. Then the ways of detecting a computer virus, at last an analysis of metamorphic virus and ways to detect metamorphic virus are presented. The research paper was done by doing a thorough research on the public resources available on the internet.

Acknowledgement- I would like to express my gratitude to the lecturers in charge, Dr. Lakmal Rupasinghe and Ms. Chethana Liyanapathirana who guiding us through the completion of this research paper. I would also like to thank and appreciated the lecture panel who oversee Secure Software Systems Module and to providing all the necessary information which were required to make the research successful.

Index Terms- Metamorphic Virus; Metamorphic Malware; Malware Detection Techniques; Metamorphic Virus Detection mechanisms, Metamorphic malware analysis

INTRODUCTION

Computer Virus is also same like a flu virus. A flu virus spreads from one person to another person. Same as that computer virus also can spread from host to host. A computer virus is designed in such a way to spread from a host to another host very easily. A computer virus is simply a computer program (Evgenious Konstantinou, 2008). This computer program consists of a malicious code that is designed to do malicious activities. Everyday hundred thousand of computers and devices gets infected by viruses (Vinod, Jaipur, Laxmi, Gaur, 2009). One of the main reasons for this immense number is because most of the computer users, keep downloading unsecured files from the internet. No matter how many times public get aware about these types of viruses, they will keep downloading it blindly. There are two terms that most of the people misunderstand the two terms Virus and Malware. Virus is only one kind of malware. There are various types of malware as Virus, Worm, Trojan Horse, Spyware etc. A computer virus can replicate only if the computer program that have the virus gets executed. A computer virus does not have the ability to replicate itself, this needs the help of a host program (Jeffrey Hortan, Jennifer Seberry, 1997). There is another kind of malware that does not need a host program to replicate, instead it can get replicated by themselves. These are known as worms. The damage caused by different kind of viruses varies. Some viruses may cause immense damage to the infected host like deleting/corrupting the existing files, overwriting the files or the entire hard disk or even can crash the entire system. Some viruses do not cause this much of damage instead they will perform activities to annoy the user, such as Pop advertisements, or just typing some random text on the screen. This kind of viruses are not harmful but though they are not harmful they will not stop reproducing or replicating (Evgenious Konstantinou, 2008).

The first Virus was said to be identified in 1986. This is said to create in a way to infect floppy disks. Though floppy disks are not popular these days, in the past, one of the main storage devices used by public is floppy disks. From that time onwards, various kinds of computer viruses were found. Over the past 2 to 3 decades, devices and the computers that gets infected by viruses increased gradually. On the internet there are publicly available sources that states about the viruses that infected large number of computer and caused huge

damage to the companies and organizations in the past (Evgenious Konstantinou, 2008). As an example, a virus called as Melissa was discovered in 1999. This virus was first released in 26th of March 1999.At that time, this virus spread so rapidly. Within a matter of minutes, email systems got overloaded. This virus was mainly spread using emails. Since the email servers got overloaded because of this Melissa virus, it did cause a huge damage which is close to almost $80 million (Pradeep, 2006). Not only Melissa there are so many examples of viruses that caused severe damage such as ILOVEYOU, Code Red etc. Even in the present day, there are so many viruses and other kind of malware that cause immense damage to the organizations and public. Another reason why malware have become common is because the virus writers make or develop virus developing kits which are available on internet. These are used by the people or individuals who do not have that much knowledge on programming languages to create their own viruses.

A typical computer virus has a process that follows. This is known as Computer Virus life cycle. In this cycle there are 4 phases. The first phase is Dormant phase. In this phase the virus does not do anything, it will just lie idle in the computer until someone activates or click on the file that the virus lies on. The next is Propagation phase. In this phase the virus will start spreading and will start copying to other files of the computer. In the Trigger phase virus changes from dormant to activation. In the last phase, the virus will do the malicious thing, which is releasing the payload (Evgenious Konstantinou,2008).

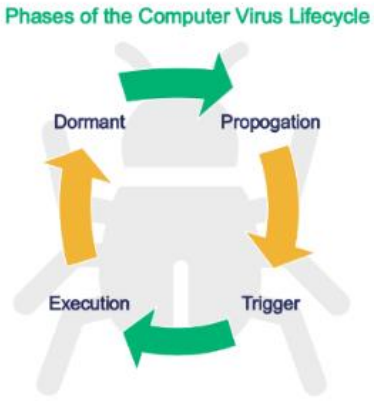

Figure 1: Computer Virus Lifecycle

The viruses can be detected using signature-based virus scanners. Signature based scanners have many numbers of malware signatures (A unit value for an identified malware) stored in their database. When scanning the target, virus scanner will create a unique signature for each of the files that are scanned and then the detected signatures are matched and compare with the signature malware available in the database. If the signatures match, that means it is a malware (Evgenious Konstantinou, 2008).

Metamorphic viruses avoid signature-based detection. Virus writers wanted to create a virus that is not detectable by signature-based virus scanning and the result is metamorphic virus. Detecting metamorphic virus is very challenging because this virus changes its code, signature every single time a system gets infected or in each generation. The drawback of signature-based scanning is that even a small change in the virus will make the scan fail so easily. Since the signature of metamorphic viruses changes each time, it infects a system, metamorphic viruses will not be got detected by signature-based virus detection scanners. Metamorphic viruses use the technique Obfuscation techniques to change the internal code. The obfuscation techniques can insert garbage code to the existing code, changing the pattern of the code instruction and renaming the existing variables or there are many other obfuscation techniques that contributes to change the virus body of metamorphic viruses. From these, the signature of the metamorphic virus also gets changed. Though the code and signature are changed, the behavior remains the same (Evgenious Konstantinou, 2008).

METHODOLOGY

A thorough research was done on the internet about Metamorphic viruses. Used publicly available information to gather and collect the required knowledge which are essential to conduct the research.

Introduction to malicious software

Virus

Computer Virus is a computer program that is designed in a way to spread from one host to another host. Computer virus can cause immense damage to the infected systems. The damage caused by viruses varies. Some viruses may cause immense damage to the system while some viruses will not cause that much of harm to the system (Evgenious Konstantinou, 2008). In 1987 a person named Fred Cohen who is an American computer scientist, and he was known as the person who invented computer virus defense techniques. He gave a definition to computer virus. He said that “We define a computer virus as a program that can infect other programs by modifying them to include a possibly evolved copy of itself”. In 1983 Fred Cohen wrote one of the first computer viruses. He wanted to

write a small program that can infect other devices while spreading from one host to another. This virus was stored and hidden in a legitimate program. The virus included legitimate programme is then stored on a floppy disk. The following is the pseudo code of the computer virus created by Fred Cohen.

```
program virus: =
{1234567;

subroutine infect-executable:=
 {loop:file = get-random-executable-file;
 if first-line-of-file = 1234567 then goto loop;
 prepend virus to file;
 }

subroutine do-damage:=
 {whatever damage is to be done}

subroutine trigger-pulled:=
 {return true if some condition holds}

main-program:=
 {infect-executable;
 if trigger-pulled then do-damage;
 goto next;}

next:}
```

code of the computer virus created by Fred Cohen.

Not only these kinds of viruses, but Fred Cohen also invented viruses that give a positive impact on the system. Such kind of virus is "Compression Virus". Compression virus is simply used to compress the executable files. During the present times, when considering about some of the viruses, Cohen's virus definition is not acceptable. Because the viruses called as Companion viruses do not modify the other files in the host, instead it will make a copy of the file that have the companion virus.

Files Infector Virus

File Infector Virus is the viruses that infects the files that are executable. The goal of this virus is to corrupt the file or make the executable file completely unusable. The viruses work by injecting or inserting the malicious code to the executable file. No matter what operating system used, either it is Linux, Mac OS, or windows any operating system can get effected by these types of viruses. Files Infector viruses are sometimes called as File Injector. The viruses infect the system with the extension .exe. If an executable file which is infected from this type of virus gets executed, it will overwrite partially or completely the infected file and then it will infect the host system and even can infect the entire local network (Mark Ludwig, 1995). An example of File Infecting Virus is Win32.Sality.BK.

Macro Virus

Macro Virus is a computer virus which is written using Macro language. Macro viruses usually spreads to email attachments or phishing emails. These viruses can access the user's address book and send email to everyone associated with the user's address book. When a specific receiver receives the email that have the macro virus attached, the macro virus spread to every document in the user's computer. Most of the receivers blindly open such kind of emails, because these emails look like they are received from a legitimate person. Another important thing about macro virus is that these viruses do not run-on operating systems instead these run-on applications. A macro can do many different malicious things. These viruses mainly aim at files. They can create/delete files, insert new files, Corrupt/modify existing files and so on. A famous example of macro virus is Melissa which was found to be released first in 1999. Macro viruses will also make the computer slower, display unwanted messages on the screen and macro viruses will make some files to ask for passwords to open it, which do not require passwords (Prof-Takialddin Al Smadi, 2007)

Worms

Another important malware type is Computer Worms. Computer Worms are somewhat like viruses, yet there is a difference between them. The difference is that Computer viruses require a host to execute them. Without the activation of the host, viruses will not be triggered. But worms are different. Worms are standalone malicious programs that can self-replicate without an activation from the host. In the early days, computer worms did not do any damage to the computer. The only thing it did is reproduced themselves (Evgenious Konstantinou, 2008). But with the development of technology the computer worms also got developed and worms became more dangerous. Nowadays worms carry out payloads. These payloads can damage the entire system as well. In the world of computer worms there are many different worms that are designed to do various malicious activities. Some worms are designed in such a way to turn the computer to "Zombies" or even "Bots". Bots are used when performing a DDOS attack. Not only this, but there are also some worms that encrypt the entire hard drive of

the computer system and will demand the user to pay a ransom to the hard drive released and some worms aims at stealing sensitive information such as Financial Information, Logins to the Bank accounts etc. The First computer worm is Morris Worm which was created by Robert Morris (Evgenious Konstantinou, 2008).

Trojan Horse

Trojan horse is a type of malware that looks like a legitimate program but can take control of the computer or infected system. Since trojan horses looks like legitimate programs many people blindly tends to click on the infected program that looks like a legitimate program. When the user clicks the infected program, trojan horse will start spreading. The malware will spread to other parts of the system. The damage done by a trojan horse varies. It can either cause a small damage or a large damage that even can make the system crash (Daniel Fuentes, Juan A Ortega, Juan Antonio, Luiz Gonzales, 2010). Trojan horses have different types like.

- Backdoor Trojan
- DDOS Trojan
- Downloader Trojan
- Ransom Trojan
- Info stealer Trojan

Trojan horses do not only infect computers but also infect other devices like mobile phones, tablets etc. There is a special kind of Trojan that targets only the android devices. This is known as Switcher Trojan. Switcher Trojan will try to infect the device of the user to attack the wireless networks that are connected to the mobile device. Since the damage done by trojan horse varies, the users should follow countermeasures or controls to prevent from trojan horse attacks. As the basic control, one can perform a scan that scans for trojan horse viruses. If this scanner scans for trojan horse viruses regularly there is a possibility that trojan horse viruses will be detected. Another basic control a user can follow is to update the system as soon as a patch is available. Patched systems have less tendency to get infected by malware such as Trojan horses. One of the best controls for Trojan horse is to make public aware about trojan horse attacks. Users should be aware about the basics such as avoid downloading from unsafe websites, avoiding opening or clicks on links that are received from sources that seems suspicious., avoid clicking on random/pop-up advertisements (Daniel Fuentes, Juan A Ortega, Juan Antonio, Luiz Gonzales, 2010).

Spyware

Spyware is a software that is designed to do malicious activities like stealing of sensitive data/information. Using Spyware an authorized person can observe the activities of a particular user. Using the gathered information an unauthorized person or attacker can track the activities performed by the user and attackers might sell these gathered sensitive data (Evgenious Konstantinou, 2008). Mainly there are 3 activities that a spyware is supposed to do.

- Infiltrate
- Monitor and capture data.
- Send Stolen data

Simply a spyware can display confidential information of a user to an unauthorized person or attacker. Oftenly spware collect information such as;

- Login credentials
- Pin numbers
- Usernames and passwords
- Credit card numbers
- Browsing history

A spyware gets masked to avoid looking like something malicious. For this purpose, most of the spware masquerades themselves to look like a normal download or a normal legitimate program. Just like other malware types. The damage cause by spware varies. Some spywares may only be used to gather financial or confidential information while some spyware may collect confidential information and use that information collected to cause severe damage to the system data (Evgenious Konstantinou, 2008).

Virus detection mechanisms

With the development of viruses, various mechanisms were introduced to detect different kind of viruses. The main concept behind anti-virus software is virus detection mechanisms. These mechanisms can be categorized as; First Generation Scanners, Second Generation scanners, Code Emulation mechanisms and Algorithmic scanning methods. Other than these the virus detection mechanisms can also be categorized as Static Detection Methods and Dynamic Detection Methods.

First Generation virus scanners

First Generation Virus scanners are simple virus scanners. They use methods that are simple to detect the infected viruses. These scanners will look for the pre-defined string in the virus code. These virus scanners

use malware signatures to detect the viruses. The main drawback of First-Generation scanners is that this will only detect the known viruses. There is another category of first-generation virus scanner. They use a different technology from malware signatures. The other category will record the length of the program code and will look for any changes in the length. Most of the First-Generation Virus Scanners are Static Detection Virus Methods. Some of the First-Generation virus scanners are *String Scanning scanners, Wildcard scanners, Mismatch scanners, Generic Detection scanners and Bookmark scanners* etc (Mustafa Irshad, Haider M. al-Khateeb, Ali Mansour, Moses Ashawa, and Muhammad Hamisu, 2018).

String Scanning scanners – This is one of the simplest scanners used by virus scanners or antivirus software to detect various virus types. The main method used by this type of viruses is to search a specific sequence of bytes or simply strings that are likely to be available in virus but that are not available in other programs. This sequence of bytes is known as signature of a virus. These signatures are stored in the databases of virus scanners. Malware signature is used to find whether the specific signature pattern is available in any of the program codes (Pankaj Kohli, Bruhadeshwar Bezawada, 2008). Below table depicts sequence of bytes in various viruses. If the below strings are detected in a specific program code, that means the program is infected with the specified virus.

Virus Name	Signature
W32/Beast	83EB 0274 1683 EBOE 740A 81EB 0301 0000
Accom.1280	89C3 B440 8A2E 2004 8A0E 2104 BA00 05CD 21E8 D500 BF50 04CD
Die.448	B440 B9E8 0133 D2CD 2172 1126 8955 15B4 40B9 0500 BA5A 01CD
Xany.979	8B96 0906 B000 E85C FF8B D589 D303 E864 FFC6 8602 0401 F8C3

Table 1: Virus name and Signature

Wildcard Scanners – The scanners that use wild cards can skip sequence of bytes or strings. These help in excluding specific bytes.

? – Represent a single character

*- Represent one or more characters.

In the following virus signature '?' represent to exclude that specific character when matching with the program code.

B440 ??E8 0133 D2CD 2172 1126 8955 15B4 40B9 0500? A5A 01CD

Mismatch Scanners – This is the type of virus detection technique which is used by IBM antivirus scanners and this technique is initially solely used in antivirus scanners. Mismatch scanners does not consider about the order of bytes of sequence. As an example, 23 44 55 67 88 22 23 33 will match any of the following byte sequences with a mismatch value 2.

23 B1 55 44 67 88 C5 23 33 22

23 55 D2 44 S1 88 67 22 33 23

23 22 D2 23 S3 67 88 55 33 44

The major drawback of Mismatch scanning technique is that when compared with other virus scanning method, this method is relatively slow (Mustafa Irshad, Haider M. al-Khateeb, Ali Mansour, Moses Ashawa, and Muhammad Hamisu, 2018).

Second Generation virus scanners

With the advancement of computer viruses, first generation viruses failed to detect advanced viruses. As a result of these second-generation virus scanners came into being. Second Generation Virus Scanners used more advanced technologies than First Generation Virus Scanners. With the introduction of second-generation scanners more advanced viruses were able to be detected (Mustafa Irshad, Haider M. al-Khateeb, Ali Mansour, Moses Ashawa, and Muhammad Hamisu, 2018). Examples of Second-Generation Virus Scanners are Smart Scanners, Skeleton Detection Scanners, Nearly Exact Identification Scanners. In the early days, it was a really difficult task to create a virus. Virus creation required expertise knowledge. Virus writers are mostly expertise in programming languages. However, with time, virus construction kits came into being. These construction kits enable even the people who does not have that much knowledge in programming to create viruses and then another important kit came into being which is mutator kit.

These mutator kits made the virus looked different from its previous/original form. Because of this difference the typical String scanning scanners which used malware signatures to detect viruses failed to detect these advanced kinds of viruses. The main technique used by Mutation kits to change the virus form is by inserting junk instructions to the virus source code. To detect these kind of viruses, Second Generation Virus Scanners were developed.

Smart Scanners –Smart scanners were able to ignore the junk instructions inserted by mutator kits. The ignored junk instructions were not stored as malware signatures. To detect the viruses, smart scanners only use a part of a virus code that does not have any junk instructions or any other unnecessary data and also these kinds of scanners have the ability to ignore the spaces and tab spaces which are entered by the mutator kits to change the original form of the virus (Essam Al Daoud, Iqbal Jebril, Belal Zaqaibeh, 2008).

Skeleton Detection Scanners – Skeleton Detection Scanners are also somewhat like smart scanners. These also ignore unnecessary junk instructions. But the difference is that Skeleton detection scanners will parse the program code line by line and will remove the unnecessary junk instructions or any other tab spaces or white spaces. When all these junk instructions are dropped, the remaining will be like a skeleton. This skeleton part will be scanned by the virus scanner (Essam Al Daoud, Iqbal Jebril, Belal Zaqaibeh, 2008).

Nearly Exact Identification Scanners – Though this is a second-generation scanner this uses techniques introduced First Generation Virus Scanners. A checksum will be calculated by using constant bytes in the body of virus. In this scenario variable bytes in the virus body are removed and a map is created using the constant bytes. Exact identification scanners are very much successful when compared with other virus scanners and this is a guaranteed method to detect viruses. The major drawback of Exact Identification Virus is its slowness. When compared with other virus scanners it is comparatively slow. The slowness is since it is time consuming to build constant byte map for viruses with lengthy codes (Essam Al Daoud, Iqbal Jebril, Belal Zaqaibeh, 2008)..

Metamorphic virus

Metamorphic Virus is one of the biggest threats to digital world. Unlike other viruses, Metamorphic viruses are hard to detect. The main difference of Metamorphic Virus when compared to other viruses is that the ability to change the code in each iteration. Since the mechanisms used in metamorphic virus to avoid detection, metamorphic virus is said to be more advanced than any other virus type. To avoid detection Metamorphic viruses mainly use code obfuscation techniques. The main goal of Metamorphic virus is to change the virus body while the functionality of virus remains same. To do so, obfuscation methods like insertions of garbage code, shrinking of code, expansion of the code register usage exchange were mainly used. Due to the advanced nature of metamorphic virus, if the right security tools are not in place, metamorphic virus can be more sophisticated and cause severe damage to the infected systems. If a particular metamorphic virus remains in an infected system for a long time, many variants of the virus will get produced which will make the virus detection even harder. There is no specific medium that is used to distribute metamorphic viruses, mainly they are distributed via email attachments. Other than that, virus can get entered to the system if the system user downloads something from a compromised website. Metamorphic virus always makes sure that the new generation of the virus is not like the previous generation or the parent body. Metamorphic viruses avoid all the first-generation virus scanners and avoid dynamic virus scanners like emulators. Virus writers who create metamorphic viruses are very aware about the weaknesses available in the virus detection scanners. The knowledge about the weaknesses were used by the virus writers to make metamorphic virus more advance and bypass virus detections performed by virus scanner. Below figure shows the evaluation of viruses including metamorphic virus.

Figure 2: Malware Evolution

Anatomy of Metamorphic Virus

Metamorphic virus works according to an anatomy of Metamorphic engine. Using metamorphic engine, the virus performs all the mutations accordingly. Metamorphic engine is also known as mutation engine.

A metamorphic engine should include the essential components such as.

1. Disassembler
2. Code Analyzer
3. Code Transformer
4. Assembler

Any kind of metamorphic engine should contain the above units. After infecting a particular system, the main goal of metamorphic virus is to changes its virus body. To do this, first the virus must find out where the virus code is available in the infected system. After finding out the location, the virus code will be converted into a set of assembly instructions and this is done by using a disassembler. The next components which is code analyzer mainly contributes to the functions performed by code transformer. Code analyzer helps code transformer by providing the necessary information like variables in the code, subroutines available, the process flow or the flow diagram of the virus and some other necessary information. The next component is Code transformer. This is the most important component of metamorphic engine. This can also be identified as the heart of metamorphic/mutation engine and sometimes called as obfuscator. The mutation part is mainly done by code transformer. Code transformer will obfuscate the code and change the code using obfuscate techniques like register exchange, changing the order of sub routines, changing the order of instructions. And now the virus code is changed, and the last part is done by the assembler. Assembler is used to convert the mutated code into machine binary code (Marco Campion,· Mila Dalla Preda, Roberto Giacobazzi, 2020).

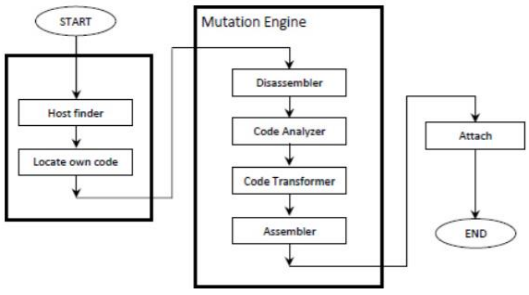

Figure 2: Process of Metamorphic virus

Obfuscation techniques

With the development of technology, obfuscation techniques also parallelly improved. These are mainly used to change and prevent viruses from getting detected. Obfuscation techniques make the virus body looks complicated but still the functions remain same. The most common obfuscation techniques used by virus writers are,

- Junk code insertion
- Register/Variable substitution
- Instruction replacement
- Instruction permutation

Junk code insertion – This is one of the most common and simplest methods used by virus writers to mutate the virus code. This can happen effortlessly because this is a simple act of entering junk codes or instructions to the original virus code. After the insertion of junk code, the virus code will look different from the original form. The junk codes are designed in a way that they won't affect the functionality of the virus. Junk codes will not change the values and simply these codes do not perform any operations (Ilsun You, Kangbin Yim, 2010).

Eg:

Instruction → ADD Reg , 0 Operation → Reg = Reg + 0

Instruction → MOV Reg , Reg Operation → Reg ← Reg

Register/Variable Substitution – This is another obfuscation technique used by the virus writers to change the virus code. In Register Substitution, registers are exchanged and in variable substitution variables are exchanged. If this obfuscation technique is used, this will help in avoiding detection from string detection virus scanners. In 1998 a virus called W95.Regswap used Register/Variable substitution as obfuscation technique. It changed the byte sequency, but it did not affect the functionality of the virus or did not made any changes to the virus functionality. Since the virus body was changed, different malware signatures were generated for each virus generation (Ilsun You, Kangbin Yim, 2010).

Example: 2 Versions of W95.Regswap

Version 1:

5A	POP edx
BFO4000000	mov edi,0004h
8BFH	mov esi,ebp
B80C000000	mov eax,000Ch
81C288000000	add edx,0088h
8BIA	add ebx,[edx]
899C8618110000	mov [esi+eax*4+00001118],ebx

Signature-
5ABF840000088B75BU0C00000881C2880000008B
1A899C8618180088

Version 2:

5A	pop	edx
BB04000000	mov	ebx,0004h
8BD5	mov	edx,ebp
BF0C000000	mov	edi,000ch
81C088000000	add	eax,00088h
8B30	mov	esi,[eax]
89B4BA181100	mov	[edx+edi*4+00001118],esi

Signature-
58BB840000088BD5BF0C00000881C0880000008B3
0899 4BA18180088

Instruction replacement – This technique will replace the instruction with instructions that perform the same function as the replaced instruction. There can be multiple ways of coding the same function. The following example shows different assembly instructions for doing the same task (Ilsun You, Kangbin Yim, 2010).

Exampe:

mov edx,0

and edx,0

sub edx,edx

Instruction permutation – In this kind of permutation technique, instructions are reordered. The ordering will not affect the functionality of the virus. Due to the rearrangement and reordering of instruction, binary sequency will look different from the previous generations. Instruction will also focus on independent instructions which the reordering will not affect the functionality of the virus.

Analysis of Metamorphic virus detection techniques

Though metamorphic virus avoided most of the typical virus detection techniques, still there are virus techniques that are capable of detection metamorphic viruses.

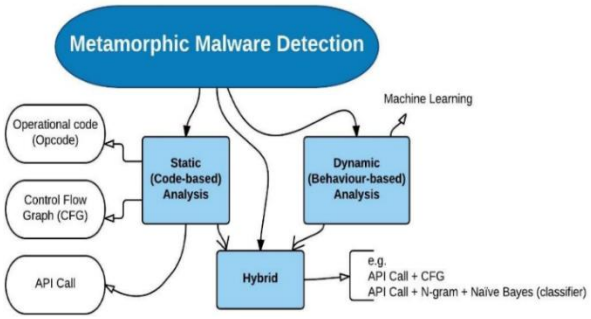

Figure: approaches to metamorphic malware detection

Operational code (Opcode) – A typical computer program is a set of instructions and data. Instruction is a set of machine language instructions that a particular processor understands and executes. These instructions always come with a pair. The pair consist of opcode and operations. Opcode is also known as operational code. Opcode will specify what to be performed and operations will provide information to the action specified by the opcode. This method was introduced by Wong and Stamp (Wong, Stamp, 2006). This study also helped in performing other malware detection studies which were performed by Stamp and Toderici (Stamp, Toderici, 2013) and Deshpande,Stamp and Park (Deshpande, Stamp, Park, 2014). Wong and Stamp analyzed the metamorphic virus and introduced Hidden Markov Model (HMM). HMM model is used for metamorphic virus/malware detection. HMM model mainly focuses on the opcode assembly sequences available in the virus source code. HMM is trained with the help of assembly opcodes or operational codes to distinguish the opcodes in the virus. HMM focus their attention of the statistical features available in the metamorphic virus because detecting signatures are useless when it comes to metamorphic viruses. Using the statistical features, HMM can detect the viruses that belongs to the same family. HMM is trained to detect different variants of the same virus family. HMM is trained for detecting the different variants that will be generated by the same metamorphic engine (Wong, Stamp, 2006).

Alam (Alam, 2014) proposed another method to detect metamorphic virus. This method too used opcode as the basis of detection. When compared with other metamorphic malware detection mechanisms, this can be considered as a unique approach. Alam introduced SWOD-CFWeight which stands for Sliding Windows of Differences and Control Flow Weight. SWOD-CFWeight uses a language which is called as MAIL which stands for Malware Analysis Intermediate Language. SWOD-CFWeight acts as an intermediate

language. The purpose of using SWOD-CFWeight is to transform the assembly language instructions. SWOD-CFWeight detected metamorphic virus by focusing on the differences available in the opcodes and focused the attention on information flow on the program.

Control Flow Graph (CFG) – Control Flow Graph is one of the mechanisms that is very old, and it is used for the purpose of malware detection for the past years. As the name suggests itself, this is a Flow Graph which is used to represent the control flow of a particular malware. This uses the notations that a typical flow diagram uses which are nodes, processes, straight lines etc. To develop a CFG every statement is considered. Though the metamorphic virus code gets modified or mutated in each generation, it will not affect CFG (Paul and Mishra, 2014). CFG will simply travel across every possible path that a particular program will take during the execution of the program. Even in recent days, still studies are done on CFG. A study done by Agrawal (Agrawal, 2012) used a method called MAA which stands for Malware Abstraction analysis. The difference of this study from the previous study is that, in the original CFG, every statement in the virus code is taken into consideration. But in here it is different. He eliminated or avoided the low-level syntaxes like function calls, returns and even conditional statements are not taken into consideration.

API Call – In this method a call graph is developed. The executable code is transformed to a call graph. The call graph consists of nodes, edges, etc. In the call graph, nodes are used to represent system calls and edges are used to represent system call sequences. Though mainly this method develops a call graph, this also utilizes the relationship that the API calls have with other API calls. This basically uses a combination of API call and Call graphs which is known as API Call graph to detect metamorphic virus (Mustafa Irshad, Haider M. al-Khateeb, Ali Mansour, Moses Ashawa, and Muhammad Hamisu, 2018). An API call graph is developed using call sequences. Then using the created call graph, a code graph was created. These code graphs are compared with the other code graphs that are generated based on various program files. For the testing purpose three obfuscation techniques were used namely Code insertion, Code replacement and Code reordering.

Hybrid Approach- As mentioned earlier, metamorphic viruses are designed in a way to avoid detection from typical signature-based malware scanners. Metamorphic viruses easily avoid detection from signature-based malware scanning by changing the structure, virus code in each generation. This became a problem for the people who designed virus scanners. Even the virus scanners that detect viruses based on behavioral pattern, also did not become effective when it comes to metamorphic viruses. The virus scanners that are designed to detect viruses based on behaviors are not efficient and it required a lot of processing. As a solution for this, Eskandari and Hashemi (Eskandari and Hashemi, 2011) thought that a hybrid approach would be more effective to detect metamorphic virus. For the hybrid approach, Eskandari and Hashemi selected API Call and Control Flow Graph (CFG). This hybrid approach is done using three phases. They are.

1. Disassembly
2. API Call Graph Generation
3. Feature generation

In the first phase, which is disassembly phase, the executable code is disassembled. The purpose of disassembly is to remove the unnecessary statements available in the executable. Disassembly is usually done using a pre-processing algorithm. From only the necessary statement a CFG will be generated. In the next phase, which is API Call Graph Generation, API Call Graphs are generated. A labelling algorithm is used for the generation of API Call Graphs. The next phase, which is Feature generation phase, as the name suggests using the generated API call graphs features of the virus is generated (Eskandari and Hashemi, 2011).

Conclusion

During the recent decade, computer virus has become a term that is common. Each day immense number of systems get infected by various malware types. No matter what controls are taken, it is not possible to prevent all the possible virus threats to a particular system. In early days since to create a virus, expertise knowledge was required, there were only a limited number of viruses. With the introduction of malware construction kits, malware became more common. With the evolution of viruses, metamorphic viruses came into being in the end of 1990s. Metamorphic virus had become one of the biggest threats to the digital world. The obfuscation techniques used by metamorphic malware- makes the malware able to avoid detection from typical signature-based virus scanners. When it comes to metamorphic virus, signature-based virus scanners are useless because of the ability that metamorphic virus to change the virus code or the virus structure. Though signature-based virus scanners fail to detect virus scanners, there are various techniques that can be used to detect

metamorphic viruses such as using opcodes, API Call graph, Control Flow Graphs, etc. With the introduction of various obfuscation techniques, metamorphic viruses got developed. Even in the future, new obfuscation techniques will be found which can be used in metamorphic viruses to make the virus more sophisticated as well as more complex future and contrary to that new techniques will be introduced in future to detect metamorphic viruses.

AUTHORS

Author – Hapucarachchi J.C, BSc(hons) in Information Technology | Specializing in Cyber Security | SLIIT | it19029214@my.sliit.lk

REFERENCES

[1] Mustafa Irshad, Haider M. al-Khateeb, Ali Mansour, Moses Ashawa, and Muhammad Hamisu, “Effective Methods to Detect Metamorphic Malware: A systematic review”, 2018.

[2] Wing Wong, “Analysis and Detection of Metamorphic Computer Viruses”, 2006.

[3] Evgenious Konstantinou, “Metamorphic Virus: Analysis and Detection”, 2008.

[4] Soumyadeep G. Dastidar, Subhrangsu Mandal, Ferdous A. Barbhuiya, and Sukumar Nandi, “Detecting Metamorphic Virus Using Hidden Markov Model and Genetic Algorithm”, 2015.

[5] Jeffrey Horton, Jennifer Seberry, “Computer Viruses: An introduction”, 1997.

[6] Ashwin Kalbhor, “A Tiered Approach to Detect Metamorphic Malware with Hidden Markov Model”, 2014.

[7] Mahmood Fazlali, Mohammed Mahdi Dehshibi, Peyman Khodamoradi, “Metamorphic Malware Detection using opcode frequency rate and decision tree”, 2016.

[8] Babak Bashari Rad, Suhaimi Ibrahim, Maslin Masrom, “Camouflage In Malware: From Encryption to Metamorphism”, 2012.