Contemporary Malicious Code Detection-Techniques

Attributes :

Dr. M.E Ezema

Computer Science Department

University Nigeria Nsukka

And

Prof H.C. Inyiama

Department of Computer and Electronics Engineering

Nnamdi Azikiwe University Awka

Anambra State Nigeria

Abstract

The major security threat that is predominant today is Cyber Warfare. What could constitute this threat? There is no doubt, this warfare is not the exchange of nuclear weapons or missiles but deployment of malicious code across or within dispersed internetwork points in order to gain access/ infiltrate and cause serious damages to confidential data in the systems. Now what constitute this threat is malicious executables especially new, hidden malicious executables often arriving as electronic mail attachments. Malicious Code is any code added, modified or removed from the system software in order to intentionally cause havoc or subvert intended function of the system. This threat has continued to grow geometrically as internet grows and constantly accelerates the trends of interconnectedness in distribution of these malware. The vulnerability of systems are weighed and exploited by rogue programmers (Authors of Malware). This paper introduces taxonomy of malicious code, types such as Computer Viruses, Trojan Horses, Logic Bombs and Worms. More importantly, it also focuses on the techniques such as signature- based and non-signature- based technologies employed to tackle this exploitation of confidential access or destruction of information resources In addition, it is an important prerequisite for the development of removal tools that can thoroughly delete malware from an infected machine.

Keywords: Vulnerability, Malicious Code, Malware, Heuristics.

1.0 Introduction

Confidentiality and integrity of information resources in computer systems cannot claim to have optimal protection and restricted access control without deploying detection techniques or approaches. What causes breach, damage, infiltration of information in computer systems is referred to as Malicious Code or rather **Malware.** [1] As internet and e-mail become an ever increasing part of our 21st-century lives, the myriad dangers and risk that come with them are increasing too, make sure you know how to detect and deal with threats that face us.[2] These malware according to Gary et al could take the form of code, scripts (ActiveX), active contents, computer viruses, Trojan Horses, root kits, spyware, key loggers, dialers, malware and rogue security software etc. Malicious code is typically disseminated across the internet either by electronic mail or web pages.

The authors (programmers) of malware are continually writing new malicious code to expose and exploit vulnerability of systems as a result of progressive growth of internet connectivity and complexity of systems. Their motives could be to infiltrate, damage or to gain unwarranted access to government confidential information resources and/or to increase the marketability of their products (anti – malware, anti-spam, antivirus).[3] According to Policy Summary No 29 article on malicious code, "creation of malicious code is a lucrative business sponsored by organized crime groups"; whose intent is to disrupt computer systems. They believed such threats which cause damages to system Operating system and hardware could lead to high demand of their products and definitely yield high profits.

2.0 Taxonomy of Malicious Code.

Malicious code basically has it classification that uniquely identifies each programmed threat in a computer system. Taxonomy is a system of classification allowing one to uniquely identify the threats as presented in the figure 1 below ; [4] threats can be divided into two categories: **Parasitic** (dependent) and **Non- Parasitic** (independent) threats. Parasitic or dependent code hugely rely on it hosts,(except viruses) to carry out its operation and it is a fragment of programs that cannot exist exclusively of some actual application program, utility or system program. Non- Parasitic are referred to as replicate, a self- contained program that duplicates itself or produce a replica of it.These fragments of programs are to be activated by a host program when trying to perform some specific functions. The classes of malware are better expressed by the diagram in figure 1 below.

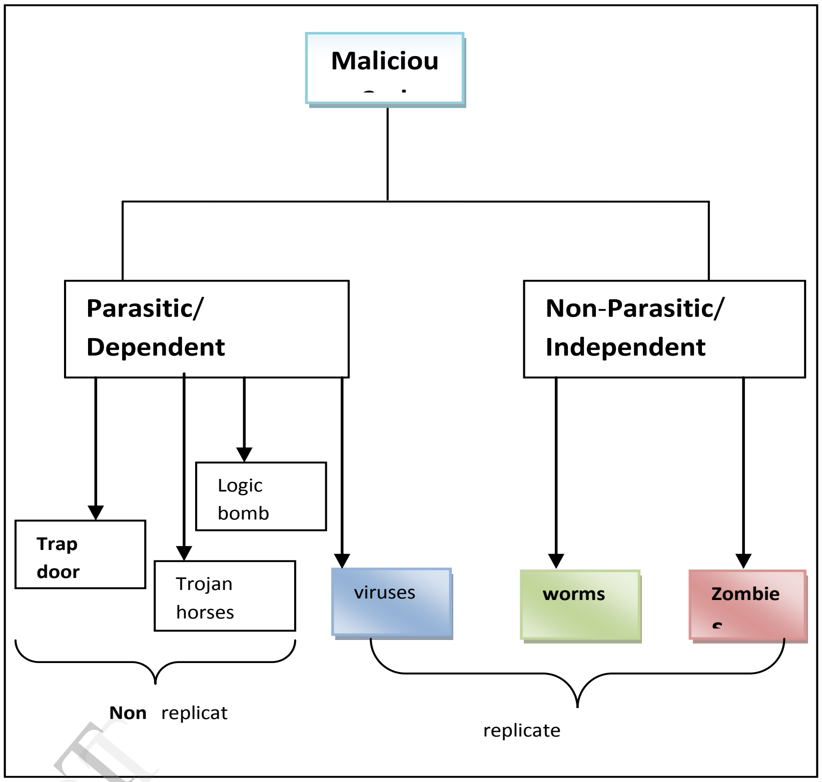

Fig 1: Taxonomy of Malicious Code

2.1 Types of non parasitic Malware

Based on the above classification of malware, one could easily identify the various types of malware. They are explained briefly below.

2.1.1 Viruses

Viruses are programs written to change the way computer systems or mobile devices work without the knowledge of the owner. It attaches itself to another computer programs to cause a mischievous acts such as erasing and modifying system's files and also replicates itself. The various forms of virus:

- **Program File Virus**: this type of virus infects the program files like .EXE, .COM, .SYS etc.
- **Boot Sector**: this type of Virus infect the system area of disk such as hard drives and removable disks, when the disk is booted;

- **Macro Virus:** this type of virus attacks the data files and they attach themselves to documents and files which are platform - independent.
- **Stealth Virus:** this virus uses ***stealth techniques*** to hide itself from detection from anti-virus software;
- **E-mail Virus:** a virus spread by email message as an attachment.

2.1.2 Worms

This is a piece of code that produces replicas or fully functional copies of itself and travel through a computer network or across the internet**.** [5] Nicholas opined worm as a self-propagating computer virus embedded in a file that creates copies of itself, which in turn create numerous copies as they travel via a computer network. Examples: Command.exe, All Users.exe, Hot Girl.src, Hotmailpass.exe, Temp.exe etc.

2.1.3 Zombies and Botnets

Zombie is a computer program that is attached to the internet to compromise and manipulate the information in the computer without the knowledge of the owner. Botnet refer to a network of zombie computers that has been taken over and put under the control of remote intruder. This zombie of computer in the botnets can consist of Pc's at home, schools, businesses and government, scattered around the world.

2.2 Types of Parasitic Malware

2.2.1 Logic Bombs

This is a program code that is written and attached to another program; it is activated when a certain conditions are met. For instance, a time bomb will attack a system and could cause serious damage like erasing system files if a license key or very important program is not found in the system.

2.2.2 Trojan Horses

This is non- replicating program that actually performs illicit functions when executed by host programs. Attackers use this malicious program to steal a user's password information such as credit card PIN or to destroy programs on a hard disk. Trojan horses consists of two parts, the Client and the Server. The server is the part that is installed on the victim's computer. When it is installed, it paves way for the remote attacker to access the victim's files as if he is sitting very close to the system.

2.2.3 Trap Door

Trap door is a code that serves as a secret entry point into a program that is intentionally attached in a program code.

2.2.4 Other forms of Parasitic Malware are;

Spyware

This is a malware that hides on one's computer with the purpose of collecting personal information and giving a feedback to the attacker. It is also software that secretly sends information about a user to third parties without notifying the user or seeking his consent. This information could be user's online activities, file accessed on the computer, e.t.c,.

Root kits

This software is specifically written to alter the standard functionality of an operating system on a computer system in mischievous and stealthy manner. By altering, a rootkit allows an attacker to act like a system administrator on the victim's computer. Rootkits may be used to install other type attacker tools such as Trapdoors and keystroke loggers. Examples: LRK5, Knark, Adore etc.

.

Virus Hoax

Virus hoax is a pseudo- warning program code that alerts the computer users the existence of virus. This is usually received in form of e-mail message which suggests the reader to send the message to others. Eventually, this will results in proliferation of email that may overloads the systems.

3.0 Detection Code Techniques

There are approaches used in detecting malicious code in computer systems. These approaches or techniques are categorized as signature- based and non – signature-based techniques.

3.1. Signature- Based Approach

Basically, the most primitive and earliest technology used in detecting malicious code is Signatures.**[6] Bodgan** envisaged that signature contains a segment of code that acts as an identifier for single malicious program. Using this technology is relatively old and repetitive which requires little explanation. This paper explained in detail the two commonest detection scheme used for different signatures.

3.1.1**Detection Schemes Based on GLFSR(Generalized linear Feedback Shift Register)**

This particular scheme is constructed as stated by **Bodgan et al (2009)** and it should be noted that by the method applied in which is a pattern generator with n = (δ, x, m) outputs. However, the general form of GLFSR is represented as feedback polynomial given in equation (1) below.

$\Phi(x) = x^m + \Phi_{m-1}x^{m-1} + \ldots + \Phi_1 + \Phi_0$ - - - - - - - - - - - - - - - - - - - (1)

This equation defined above stated that every coefficient of the polynomial equation defines the feedback connection specific to a GLFSR.

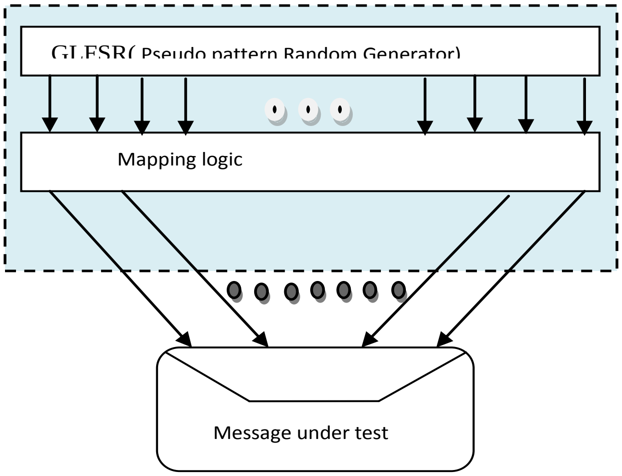

Fig 2. Verification of every process

However, the aim of this is to acknowledge if a message is compromised by certain (single) attack and such attack can be identified as its signature. [7] Thus, the outputs from the GLFSR as indicated by **Sukwong** are being transformed by a dedicated mapping logic into signatures of the attacks.

3.1.2**. Group Detection Scheme**

- Another signature- based adopted according to **Bodgan** was proposed as *column approach* for matching of the Pseudo Random Pattern Generator(PRPG) pattern into target patterns. This scheme introduce steps to be taken to obtain a unique detection scheme for multiple attacks. The steps to follow is as follows;

1. Design a composite pattern by using a GLFSR functioning as PRPG engine
2. Propose a dictionary of attackers signatures that should be detected by the outcome scheme
3. Establish a minimal set of PRPG patterns(Target Patterns)

4. Obtain the Target Patterns by applying a column matching process

5. Design an efficient combinational circuit with the PRPGin that the combined pattern generator produces all signature from the above constructed dictionary.

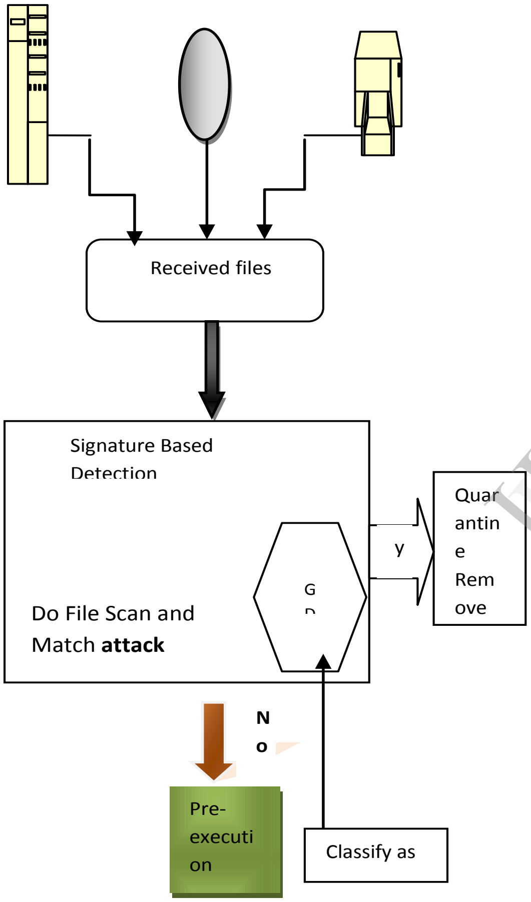

Fig 3. Integration of GDS in an Intrusion Detection System

3.2 Non- Signature-Based Approach

The formal approach used to combat malicious code weren't so strong to defend systems against metamorphic viruses. Non–signature–based approach uses latest defensive mechanisms such as "Heuristics", Proactive Detection, "Behavioral Detection", and Host Intrusion Protection Systems{ HIPS.) [8] However, a model was proposed according to **Alisa** grouped this approach as Technical and Analytical Components. Truly, these components may not be separated at module or algorithm level within every malicious code.

3.2.1 Technical Component

This component of a malicious detection scheme is defined as a data collection system that provides the data that need to be analyzed. **Alisa** stressed that malicious code can be evaluated as a long string of data, as a series of instructions, or by the effect it has on the Operating System. These different views help to explain why there are so many different methods and possibilities as stated in the diagram. These methods are listed in terms of increased level of abstraction. To combat malicious programs, greater and higher levels of abstraction are required. On this note, the lists of methods are explained briefly below.

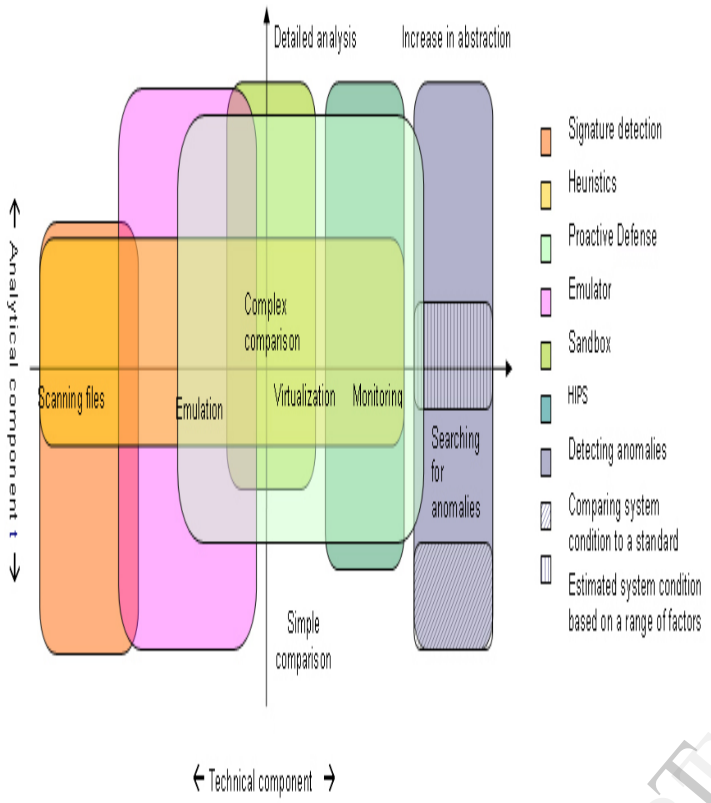

Fig 4 A Model for Accessing Methods of Malicious Code Detection.

- **Scanning Files:** Scanning simply refers to extracting data from files, structuring that mass of bytes in a specific manner and hauling those structured byte to analytical component. This "analysis" was a simple comparison of byte sequences in file against known signatures.
- **Emulation:** Emulation approach is a step between treating a program as a collection of byte (scanning) and processing a program as particular sequence of actions. Emulator's major function is to focus on its events rather than inanimate bytes of data per second. Emulators are used in many, and possibly all, major antivirus products. They may be used as a basic, core-level protection engine or as "insurance" for more abstract and sophisticated engine, such as **Sandbox.**
- **Virtualization – The Sandbox**: This is the logical extension of emulation and a sandbox is one form of virtualization. Sandbox provides a certain rules in which the Operating System acts on. One such rule might be a ban in modifying the system directory. For instance, if a program tries to modify the system directory, it may be fed into virtual copy of system directory so that it can continue to operate without impacting the Operating system. The line in between emulation and virtualization may be nice one, but it is a clear one. Emulation occurs in a fully contained, controlled and separate environment. Virtualization occurs in the real world but under careful rules and guidance.
- **Monitoring System Event**: While the emulator or sandbox observes each program separately, monitoring system events is the next level of abstraction. It involves the simultaneous observation of all programs to understand their impact on the operating system. Monitoring of system events is used as the technology component in several major antivirus products and as a main component in individual system.
- **Scanning for System Anomalies**: Scanning for system anomalies is the most
abstract method used to collect data about a potentially infected system. This method relies in basic principles:
    a) An operating system, together with the programs running within that system, is an integrated system;
    b) If malicious code is run in the environment, then the

system will have an "unhealthy" status. This differs from a system with a "healthy" status, in which there is no malicious code.

c) The operating system has an intrinsic system status".

To detect malicious code effectively, using the system anomalies method, a relatively, complex analytical system such as expert system or neural network is required. However, due to complexity, the system anomalies technology is still classified as an emerging technology.

3.2.2 Analytical Component

Having explored technical component, we now focus on analytical component. This component indicates the degree of sophistication in decision making algorithm which varies above transverse the vertical axis. Analytical component can be divided into three different categories:

- **Simple Comparison**

Technology that falls into this category is purely on binary- a clear "Yes or "No". For example, an identification of malicious code by locating a specific byte sequence. Another higher level example is identifying a suspicious program through the use of single action that it takes, such as creating a record in a critical section of the system.

- **Complex Comparison**

This technology has to do with multiple comparisons of objects into corresponding samples. The template for these comparisons can be flexible and the results will be probability-based. For example, identification of malicious code using several code byte signatures, each of which is non- rigid

4.0 **Actual Technology**

Most of the security software today like Kaspersky, Panda Cloud are equipped with a Host Intrusion Protection Systems(HIPS), Proactive technology or non-signature tech. A user's understanding of HIPS based in a user- friendly definition describes HIPS as "as a monitor that analyses systems events for malicious code and the description is one that could mean almost anything in the security world, such as emulator engine that is equipped with heuristic analysis system.

4.1 **Heuristics**

Heuristics is a combination of research methods capable of detecting what was previously unknown. It was first and foremost type of analytical component in protection software. Outside a specific context, in terms of problem- solving,it closely resembles an "unclear" method used to resolve "unclear" problems.[9] However, **Symantec Security Company** viewed heuristics as a distinct technology- one that would identify a malicious code by using flexible assigned byte template. In other words, it implies when the system is working with files and at the same time using complex comparison approach. When talking about heuristic detection, developers of security software are usually referring to a protection system with analytical component that uses a *fuzzy* search to find a solution. This is equivalent of saying that the analytical component involved uses either complex analysis or any expert system.

Conclusion

This paper treats on malicious code detection and its techniques such as traditional method: signature-based and non-signature –based technologies. This piece of work will serves as a note of clarification to those who need to grasp what malicious codes are and what form they could appear. These techniques provide defensive methods in ensuring effective combat of these programmed threats in computer systems.

References

1. www.kaspersky.com/threats
2. G, McGraw and G, Morrisett (2000), “Attacking Malicious Code”: A Report to the infosec Research Council, IEEE Network.
3. Policy Summary No. 29, Malicious Code: “Importance of Information Security”, http:// www.cio.gov.bc.ca/cio/information security/index.page?.
4. Mohammad Heidan (2004), “Malicious Code in Depth”, Babisandez@yahoo.com.
5. Nicholas Weaver, Vern Paxson et al, “Large Scale Malicious Code”: Research Agenda.
6. R. Bodgan, (2009). *A Data Perspective on Information Transmission Over Distributed Systems*. Publication place: PolitethnicaPublishing.
7. O.Sukwong, H. Kim and S. Hoe, (2011) Commercial Antivirus Software Effectiveness: An Empirical Study- Computer, 44(3): 63-70.
8. Alisa. Schevchenko, (2008) “White Paper”: Malicious Code Detection Technologies, www.kaspersky.com
9. Symantec global internet security threat report [Online]. Available: http:// www.symantec.com/about/news/release/article.jsp?prid=2009041301.