Chapter 13
Metamorphic Malware Analysis and Detection Methods

Vinod P.
Malaviya National Institute of Technology, India

V. Laxmi
Malaviya National Institute of Technology, India

M.S.Gaur
Malaviya National Institute of Technology, India

ABSTRACT

The term 'malware' is collectively used for any program which accesses the system through surreptitious (often unauthorized) means, with malicious intent, resulting in data loss and/or corruption. Some examples are viruses, worms, trojans, botnets etc. Malware is becoming a world-wide epidemic as one infected computer system may compromise all networked systems. Millions of computers connected to the Internet exchange useful data and information and are exposed to malware threats. Malware programs may apply different techniques for unauthorized access, but all of them compromise the system in one way or another. In order to protect from the threats imposed by the malware, we need to understand the techniques used by them in exploiting system vulnerabilities and build an effective detection system. This contribution chapter surveys various malware types, infection mechanisms, detection techniques and metamorphic viruses. This chapter also presents a Longest Common Subsequence (LCS) based methodology for metamorphic malware detection.

Malware or malicious code is software causing some unwanted and unauthorized activities on the system in a stealthy manner without the knowledge of the user. Malware activation makes the system vulnerable to malicious activities of the attacker. Malware makes its way to the system because of the lack of security awareness amongst users. It spreads through network vulnerabilities such as email attachments. Malware can be classified as viruses, Trojans, botnets, adware, spyware, rootkits etc. Some of the activities of notable malware are follows:

- Deletes important files from the system.
- Logs every keyboard input and sends this information to outside computer system.

DOI: 10.4018/978-1-60960-123-2.ch013

- Steals sensitive information by spying on various user activities.
- Brings down the performance of the machine by slowing down its speed.
- Sends storms of spam mails and implements many backdoors to leak user information.

Virus writers target limitations of anti-virus products to create new variants. The traditional method used by the Anti-virus products (AV) is signature based detection. Signature is a binary pattern that uniquely identifies a virus. This method imposes some problems as (a) it fails to detect encrypted code (b) lacks knowledge of the program semantics (c) the size of signature repository increases with time and (d) fails to detect obfuscated malware.

This chapter discusses various types of malware, infection mechanisms, detection and complex obfuscation techniques. The organization of the chapter is as follows: Section 1 presents a brief history of computer viruses followed by malware evolution in Section 2. In Section 3, infection mechanisms are presented. Section 4 discusses anatomy of metamorphic viruses. Section 5 covers Metamorphism techniques. Section 6 details existing malware detection techniques. Section 7 introduces metamorphic malware detection methods. Section 8 highlights our proposed method. Finally, concluding remarks and future work are discussed.

1. EARLY MALWARE

Computer viruses were the earliest malware. A virus spreads by attaching itself to a host program. A typical virus consists of three parts – *infection mechanism, triggering mechanism* and *payload insertion*. Pseudo code of a virus illustrating these three components is shown in Figure 1(a). The virus first searches for infectable data/device on the victim machine. If trigger returns 'true', payload is delivered. The malicious payload performs intentional or unintentional damage to the host application or machine.

1. 1 Boot−Sector Infector

When the machine is powered on, ROM based BIOS performs *"power on self test"* and searches for boot device. Once the boot device is identified, BIOS reads boot block(s) and transfers control to the code in the boot block code. This step is called the *primary boot*. The primary boot loads secondary boot code which understands the file system structure. This secondary boot code is responsible for loading the operating system kernel.

A boot sector infector (BSI) virus infects the boot block. BSI relocates the original boot block to a specific location and the boot block is replaced by the virus code. After infection, control is transferred to the boot block so as to avoid any suspicion of infection. Choosing a specific location

Figure 1.(a) Pseudo code of computer virus and (b) infection mechanism of the virus

```
def virus():
        infect()
        if trigger() is true:
                payload()
```

(a) Pseudo code of computer virus

```
def infect() :
        repeat k times :
        target = select_target ()
        if no target
                return
        infect_code (target)
```

(b) Infection Mechanism