Chapter 1
Optimal Features for Metamorphic Malware Detection

P. Vinod
SCMS School of Engineering and Technology, India

T. K. Ansari
SCMS School of Engineering and Technology, India

Jikku Kuriakose
SCMS School of Engineering and Technology, India

Sonal Ayyappan
SCMS School of Engineering and Technology, India

ABSTRACT

Malware or malicious code intends to harm computer systems without the knowledge of system users. These malicious softwares are unknowingly installed by naive users while browsing the Internet. Once installed, the malware performs unintentional activities like (a) steal username, password; (b) install spy software to provide remote access to the attackers; (c) flood spam messages; (d) perform denial of service attacks; etc. With the emergence of polymorphic and metamorphic malware, signature-based detectors are failing to detect new variants of these malware. The primary reason is that malicious code developed in new generation have different syntactic structures from their predecessor, thereby defeating any pattern matching techniques. Thus, the detection of morphed malware remains a complex open research problem for malware analysts. In this chapter, the authors discuss different types of malware with their detection methods. In addition, they present a proposed method employing machine learning techniques for the detection of metamorphic malware. The methodology demonstrates that appropriately selecting prominent features could improve the classification accuracy. The study also depicts that proposed methods that do not require signatures are effective in identifying and classifying morphed malware.

1. INTRODUCTION

Past few decades have shown tremendous increase in the use of computers that can invariably process small to big data. Likewise, we have also witnessed the popularity of Internet for usage for e-shopping, e-learning, e-reservation etc. In each of these applications online transactions is required to be performed. Vulnerabilities associated with the Internet, computer systems, softwares and operating systems are exploited by malware attackers and many black hat users to develop and launch sophisticated attacks. Mostly, attacks are created by recreating malicious programs (a.k.a

DOI: 10.4018/978-1-4666-6086-1.ch001

malware) using existing malware generation kits also known as *virus constructors*. Malware in general refer to all unwanted computer program (computer viruses, Trojans, rootkits, worms, adware, spyware etc.) that disrupt the normal functioning of the system. Emergence of free and open source software has shown increased market for malware writing which now have evolved into a profit making industry. The goal of these malicious software include activities like identity threats, consume system resources, and allow unauthorized access to the compromised systems. A common characteristics of malware is the capability to replicate and then propagate. Malicious programs make use of files, emails, macros, bluetooth or browser as a source of infection for its propagation.

Since the development of anti-virus (AV) software, signature scanning or pattern matching techniques are predominantly being used (Aycock, J 2006). Signature is a unique byte pattern or string capable of identifying a malicious code. Although, this method performs well in determining malware, however signature based scanning fail to detect unseen samples or *zero day malware attack*. Signature based techniques have some limitations on detection like (a) failure to detect encrypted code (b) lack of semantics knowledge of the programs (c) increase in the size of signature repository and (d) failure to detect obfuscated malware (Vinod et al, 2009). In order to circumvent the pattern based detection method, malware writers make use of complex obfuscation techniques to generate new strains. Obfuscation can take different forms (a) code packing (Yan, W. et al, 2008) (b) encryption of code using random decryptors (also known as *polymorphism*) and (c) complete code morphism which is referred as *metamorphism*. The basis of generating the metamorphic malware is to increase variability in the structure of code from one generation to another generation without affecting the functionality of programs.

Malware detection methods can be broadly classified as *static* and *dynamic*. With static analysis, the malware is detected by examining the code without its execution. Thus, static analysis is fast but may fail to detect parts of the malicious code that are executed only during runtime. During static analysis, the scanner checks for strings, file names, author signatures, system information, checksum etc. that differentiates malware from the benign program.

In dynamic analysis, samples are executed in a controlled environment. The scanners employing this method examine function/system calls, status of processor registers, flags, API parameters to determine if a program can be classified as malicious. Although, dynamic analysis is an improvement over static analysis where the detection time is usually very slow and therefore cannot be considered as the exclusive approach for malware detection. The main reason in dynamic analysis that the scanner tries to trace complete execution paths of the suspected sample. Infection of systems is the primary risk associated with dynamic analysis. To avoid this, malware scanners use virtualization or emulation based techniques. This reduces the efficiency as execution time is increased. Dynamic analysis may not succeed if malware incorporates *Anti-VM* and *Anti-emulation* checks.

Data mining methods are also gaining prominence in the detection of malware. In this method, a classification algorithm is used for modelling malware and benign behaviors/structure. The classifier is subjected to diverse malicious and benign patterns for the categorization of unseen samples (malware or benign). Recently machine learning techniques have gained popularity for the detection of malcode (Nir, N, 2012; Eitan, 2009; Asaf, S, 2009; Dima, S, 2009). Features of different category (Vinod, P., 2013) such as opcode *n-gram*, API call sequence, Portable Executable metadata or strings extracted from functions are used for the purpose of classification. In this chapter, we introduce different approaches encompassing data mining techniques for the identification of synthetic malware. For increased accuracy in the detection process, we discuss in detail the feature