Chapter 96
Advancing Malware Classification With an Evolving Clustering Method

Chia-Mei Chen
Department of Information Management, National Sun Yat-sen University, Kaohsiung, Taiwan

Shi-Hao Wang
Department of Information Management, National Sun Yat-sen University, Kaohsiung, Taiwan

ABSTRACT

This article describes how honeypots and intrusion detection systems serve as major mechanisms for security administrators to collect a variety of sample viruses and malware for further analysis, classification, and system protection. However, increased variety and complexity of malware makes the analysis and classification challenging, especially when efficiency and timely response are two contradictory yet equally significant criteria in malware classification. Besides, similarity-based classifications exhibit insufficiency because the mutation and fuzzification of malware exacerbate classification difficulties. In order to improve malware classification speed and attend to mutation, this research proposes the ameliorated progressive classification that integrates static analysis and improved k-means algorithm. This proposed classification aims at assisting network administrators to have a malware classification preprocess and make efficient malware classifications upon the capture of new malware, thus enhancing the defense against malware.

INTRODUCTION

Malware has been one major issue in network security (Zhuang, Ye, Chen, & Li, 2012; Nikolopoulos & Polenakis, 2016), and received studies have uncovered that malware can be divided into distinctive families according to specific extractable features, where within-family malware are usually mutations and fuzzification from the same origin. In order to detect malware, security administrators rely on hon-

DOI: 10.4018/978-1-7998-2460-2.ch096

eypots and intrusion detection systems to capture malware from the internet. Despite the enlightening discovery and efforts in malware collections, network security practices remain focused on immediate malware that are attacking at the moment instead of tracing their origins for more timely reaction. This is possibly due to the fact that malware have various types, such as source code, binary executing code, shell scripts, Perl scripts, and so on, so that classification is a complex and challenging task. Moreover, due to the proliferation of network and advanced capabilities of developers, malware propagation speed and structure complexity are on the increase. In another word, network security has not fully exploited the potential for malware classifications, thus efficient and timely classification is needed (Altaher, Almomani, Anbar, & Ramadass, 2012).

As an attempt to enhance malware classification, this research integrates static analysis with improved k-means algorithm (MacQueen, 1967) to analyze the malicious code. On the one hand, improved internet usage and bandwidth incur dramatic increase in the number of malware that security administrators often find it very challenging to respond, so a systematic classification is expected (Wen & Yang, 2017). On the other hand, even though honeypots and intrusion detection systems have been useful tools for security administrators to guard against internet attacks, the enormous amount of malware of various types dazzle security administrators. Besides, ways of internet attack vary greatly, increasing the difficulty of network protection. Jointly considering the aforementioned challenges, this research proposes a method to classify malware based on their features and categorize them into different families so that security administrators are able to shorten time needed to decide on effective responses.

In order to achieve the research goal of developing a malware classification approach that is more dynamic, malware source code are the target of analysis. Generally speaking, binary files must be "coded" by source code; hence, source code can present the behavior of the entire system. Source code can precisely describe the behavior and function of application programs (Annervaz et al., 2013). And there are some researchers who believe that the analysis and handling of source code will become more and more important in the future (Harman, 2010). In this case, direct investigation of source code can reveal undetected malware behavior types via binary code analysis (Huang, 2013; Yang, 2012). There are some researchers who find that most malware source code were from duplication and modification rather than creations of innovation or new drafts (Park, Zhang, Reeves, & Mulukutla, 2010). To sum up, by delving into the source code, we are able to categorize malware into different families to ease future identification of responses.

The importance of classifying malware into different families is receiving more attention in recent years (Huang, 2013; Yang, 2012) studies. It is noticed that a diversity of malware are often extensions of other existing malware families, presenting similar structures and functions. For instance, shut-down antivirus systems connect to IRC (Internet Relay Chat) channel and setup backdoor. Besides, there are identical or similar functions and structures in specific behaviors. According to literature, malware keep evolving and attacks are dynamic and persistent events (Christian, Lim, Nugroho, & Kisworo, 2010). In this vein, by clustering malware source code, similar behaviors can be found from new malware, thus enabling one to properly categorize the newly found malware. Ye (Ye et al., 2009) also believes that the results from clustering discovered malware are beneficial for analysts to understand and interpret malware (Ye et al., 2009). Therefore, malware clustering is critical and helpful for forensics.

When it comes to classification methods, our choice of k-means algorithm is under the consideration of balancing time and effort constraint and malware discernment. For security administrators, there is a variety of work to do on a daily basis, but once a malware hits one computer in the organization, it spreads in a hard-to-keep-up manner, so the crucial task is to identify the best or most feasible mecha-