DepSim: A Dependency-Based Malware Similarity Comparison System*

Yang Yi¹², Ying Lingyun¹³, Wang Rui², Su Purui¹, and Feng Dengguo¹²

¹ State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
² State Key Laboratory of Information Security, Graduate University of Chinese Academy of Sciences, Beijing 100049, China
³ National Engineering Research Center for Information Security, Beijing 100190, China

{yangyi,yly,wangrui,supurui,feng}@is.iscas.ac.cn

Abstract. It is important for malware analysis that comparing unknown files to previously-known malicious samples to quickly characterize the type of behavior and generate signatures. Malware writers often use obfuscation, such as packing, junk-insertion and other means of techniques to thwart traditional similarity comparison methods. In this paper, we introduce DepSim, a novel technique for finding dependency similarities between malicious binary programs. DepSim constructs dependency graphs of control flow and data flow of the program by taint analysis, and then conducts similarity analysis using a new graph isomorphism technique. In order to promote the accuracy and anti-interference capability, we reduce redundant loops and remove junk actions at the dependency graph pre-processing phase, which can also greatly improve the performance of our comparison algorithm. We implemented a prototype of DepSim and evaluated it to malware in the wild. Our prototype system successfully identified some semantic similarities between malware and revealed their inner similarity in program logic and behavior. The results demonstrate that our technique is accurate.

Keywords: Malware Analysis; Similarity Analysis; Dynamic Taint Analysis.

1 Introduction

Malware, software with malicious intent, such as viruses, worms, Trojans, and backdoors has emerged as a widely-spread thread to system security. According to a Microsoft report [3], in the first half of 2009, as many as 39,328,515 computers around the world have been infected by malware. It is difficult to detect and stop malware spreading reliably because new and polymorphic malware programs appear rapidly. For example, AgoBot [10] has more than 580 variants since it first came into view in 2002.

* Supported by the National Natural Science Foundation of China under Grant No. 60703076, 61073179; the National High-Tech Research and Development Plan of China under Grant No. 2007AA01Z451, 2009AA01Z435.
One of the main factors driving this explosion of variety is that malware authors usually reuse their old programs because of the cost of developing a new malware [18]. To address this dilemma and fast analyze malicious code, researchers would like to re-use stale behavior profile information from a prior malware sample by using similarity comparison to identify and analyze the malicious code variants. The problem of finding similarities in programs has been a central problem in malware analysis.

Previous efforts to automatically analyze and compare malware focused primarily on two types of comparisons, comparison of program structure and comparison of program behavior. Several program structure comparison methods have been proposed [1] [2] [7] [9] [13] [20]. For all these structural comparison approaches, successfully disassembling of malware is a pre-condition; malware writers usually use obfuscations to hinder traditional malware similarity comparison system, such as packing, encryption and instruction permutation, as these transformations can obviously change the content-based signatures of code. Without the precondition of successfully disassembling of malware, it is hard to deal with obfuscated malware with structural comparison. Previous behavioral comparison methods [8] [10] [11] [12] [15] mainly depend on behavior sequences, which could be changed easily by function-reordering, junk-insertion and other means of techniques. In addition, both two types of comparison methods use text distance or weighted text distance to measure the difference. These ways ignore the logical relations among system calls and focused primarily on sequence-based signatures that could be obfuscated easily and significantly. In summary, malware writers use various obfuscation techniques to transform their malware to make it hard to analyze. The means for traditional analysis are not sufficient to alleviate a threat posed by so many obfuscated malware variants.

To address the limitations of existing binary similarity comparison tools, we proposed and evaluated a novel technique, called DepSim, to find behavioral similarity. Unlike existing systems [11] [12], DepSim’s similarity comparison algorithm does not operate directly on system call sequences. We have implemented a proof-of-concept system of DepSim based on Wookon [29], our dynamic analysis system that monitors the execution of a malware in a controlled environment. DepSim first uses taint analysis and backtracking techniques to construct extended control dependency graphs and extended data dependency graphs. Then, at dependency graphs preprocessing phrase, we remove junk calls, reduce loops and take other normalization measures to convert them to a normalized form. At last, a customized graph isomorphism algorithm is used to find the best matches between dependency graphs.

In order to evaluate our system, we have implemented our prototype system and conducted experiments on collected wild malware. The experimental results demonstrate that our malware analysis techniques and semantic similarity comparison algorithms are more accurate than previous techniques used in binary similarity analysis, especially in dealing with obfuscated malware. Summarizing, this paper makes the following contributions:

- We present a novel, precise approach to describe malware behaviors and relations between them by extending CDG (Control Dependency Graph) and DDG (Data Dependency Graph) with Virtual Vertex and behavior profile.
- We propose a new semantic similarity comparison algorithm. The isomorphism algorithm depends on extended dependency graphs and is more accurate than previous techniques.