Subgraph-based Metamorphic Malwares Analysis

Jonghoon Kwon, Jehyun Lee, Teabum Kim, Heejo Lee
Div. of Computer and Communication Engineering, Korea University
Seoul, South Korea
[e-mail: signalnine, arondit, ktb88, heejo@korea.ac.kr]

Abstract

Malware authors commonly used code obfuscation to evade detection mechanisms. When this technique is applied to malwares, they can change their instruction sequence and also even their signature. These malwares which have same functionality and different appearance are able to evade signature-based AV products. Thus, AV venders paid large amount of cost to analyze and classify malware for generating new signature. In this paper, we propose a new approach for analyzing metamorphic malwares. The proposed mechanism first converts malware's API call sequences to CodeGraph through dynamic analysis. After that, we extract all subgraphs and analyze how similar two malware's behaviors are through subgraph similarity. To validate proposed mechanism, we use 46 real-world malwares include 20 variants. In evaluation, all metamorphic malwares are classified correctly, and similar module behaviors among different malwares are also discovered.

***Keywords*:** Malware, Metamorphic Malwaer, Obfuscation, Behavior Analysis

1. Introduction

Malware is a software implemented to damage computers or networks without any user's intention. Such malware like trojan, virus, worm and bot leads most kinds of cyber crimes, such as DDoS attacks, spam, click fraud and information theft. Even, there is numerical effort to detect malware, malware developers are constantly archieve their intention by thwarting the efforts.

The major difficulty of malware detection is the rapid increase of malware variants. As a Sysmatec report[1], the number of new malware signatures has shown extremely growth by more than doubling on a year-to-year between 2006 and 2008. Moreover, new signature creation in 2009 is figured about 3M. The main reason for the increase is that the malware variants can be easily poduced using code obfuscation[2][3]. The code obfuscation provides chance to evade existing AV scanners with inexpensive cost. Thus, AV venders need to pay large cost for new signature.

In this paper, we propose a new behavior-based analysis method for metamorphic malware. Our approaches first converts malware's API call sequences to CodeGraph[4] through dynamic analysis and extract all feasible subgraphs to analysis. Lately, we compare a behavior of malwares using the subgraphs. In evaluation, we can classify metamorphic malwares succesfully, and also find several intereting features about similar modules used by different malwares.

This research was supported by the IT R&D program of MKE/KEIT, [KI001863, The Development of Active Detection and Response Technology against Botent] and Mid-career Researcher Program through NRF grant funded by the MEST[2010-0027793]. In additionaly, this research was sponsored in part by the R&BD Support Center of Seoul Development Institute and the South Korean government (Project title: WR080951, Establishment of Bell Labs in Seoul / Research of Broadband Convergent Networks and their Enabling Technologies)

2. Proposed Mechanism

Our previous research which named CodeGraph focused on semantic characteristics of malware. This approach have shown we can analyze the characteristics of a portable executable binaly as a directed graph, and succesfully simplify the graph through classifying API calls commonly used in malwares to 128 groups.

Even the advantages, CodeGraph are not able to contain malware's whole instruction sequence and module level information. To overcome such difficulties, we newly adopt dynamic analysis and subgraph analysis. We divide and extract all feasible subgraph sets from CodeGraph. These subgraphs will cover all posible module level activities and can be used to analyze not only metamorphic malwares, also different malwares which are producted using same modules. The numbers of all feasible subgraphs of malware *M* with node *n* is presented as follows.

$$|G(M_n)_i| = \sum_{k=1}^{n-2} (k) \quad (1)$$

These subgraphs are used to find exact matched subgraph $SG(M_{src}, M_{tgt})_i$ through inter-comparison. After that, $SG(M_{src}, M_{tgt})_i$ are belong to part of similarity measurement *Sim*.

$$Sim(M_{src}, M_{tgt}) = \frac{\sqrt{\sum [SG(M_{src}, M_{tgt})_i]^2}}{n(M_{tgt})} \quad (2)$$

Also, $SG(M_{src}, M_{tgt})_i$ can be used behavior graph of modules which are part of same modules of two different malwares. Such information will be useful to trace and analyze malware constructed with multiple modules.

3. Experimental Results and Analysis

To validate proposed mechanism, we collect 45 real-world malwares include 19 metamorphic malwares, and gather total 1,035 analysis file through inter-analysis. Our experiments is based on Intel Core2Duo 2.66Ghz, 4GB main memory. Particularly, we use VMware and WindowsXP without any security patches as a guestOS. The similarity test results of metamorphic malwares are shown in Table 1.

Table 1. Metamorphic malware classification

Malware	Variants	Sim
Win32.Worm.Allaple.Gen	5	1
Win32.Worm.Vb.NVA	9	1
Trojan-Downloader.Win32.Multdl	2	1
Trojan.Downloader.Win32.Delf	3	1

And also we can find malwares which have similar behavior patterns but classified different malwares by AV venders. *Worm.Win32.Auto-run.lkj,Win32.Worm.VB.NVA* have shown *Sim* 0.99. Even thought, they are addressed different malware, they have to be classified as variant malwares in terms of behavior. Fig.1 (a) presents exact matched subgraph for these malwares

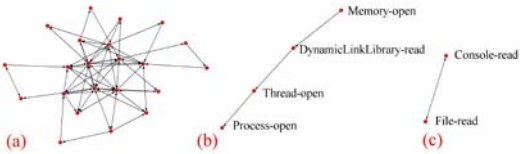

Fig. 1 Exact matched subgraph sample

Another interesting result is that some malwares show matched subgraphs, even they are ranked low similarity. Fig.1(b) and (c) illustrate subgraphs which occurred in *Win32.Worm.VB.NVA* and *Win32.Worm.Allaple.Gen*, and it is prospected as process folk and information collecting.

4. Conclusion

In this paper, we propose a new approach to analyze and classify metamorphic malware. The subgraph analysis is able to provide not only metamorphic malware classification, but also behavior analysis of modules used by different malwares. Such information can be useful to AV venders and make malware writer harder to develop metamorphic malwares.

Future work is extracting semantic signatures from various kinds of malware. Additionally, we will try to analyze more interesting behavioral features of metamorphic malware.

References

[1] Symantec Co., "Symantec global internet security threat report", Apr. 2010.

[2] M. D. Preda, M. Christodorescu, S. Jha, and S. K. Debray. "A semantics-based approach to malware detection". *ACM Trans. Prog. Lang. Syst.*, 30(5), 2008

[3] C. Nachenberg. Computer virus-antivirus coevolution. Commun. *ACM*, 40(1):46–51, 1997.

[4] J. Lee, K. Jeong, and H. Lee, "Detecting Metamorhpic Malware using Code Graphs", *ACM Int'l Symp. on Applied Computing (ACM SAC)*, Mar. 22. 2010.