New Malware Detection System Using Metric-Based Method and Hybrid Genetic Algorithm

Jinhyun Kim
School of Computer Science & Engineering
Seoul National University
1 Gwanak-ro, Gwanak-gu
Seoul, 151-744 Korea
jh@soar.snu.ac.kr

Byung-Ro Moon
School of Computer Science & Engineering
Seoul National University
1 Gwanak-ro, Gwanak-gu
Seoul, 151-744 Korea
moon@snu.ac.kr

ABSTRACT

Malicious software, or malware for short, is one of the most serious threats to computer systems. Malware disguise techniques are becoming more sophisticated, and signature-based malware detection systems can not cope with disguised malware timely. In this paper, we propose a new approach to detect disguised malware, focusing on the malware scripts. The proposed system consists of a metric-based detection algorithm and a hybrid genetic algorithm. The genetic algorithm tries further detection by extracting the main core of a program. Experimental tests on the proposed system show a remarkable performance improvement over existing anti-virus programs.

Categories and Subject Descriptors

D.4.6 [**Software**]: Security and Protection—*Invasive software*; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*

General Terms

Algorithms, Experimentation, Security

Keywords

Malware detection, metric-based method, hybrid genetic algorithm, malware disguise techniques

1. INTRODUCTION

Malware, or malicious software, is a computer program that performs any kind of malicious action to computer systems. The malicious actions include collecting users' private information and sending it to someone else, redistributing the malware, popping up advertisements, deleting arbitrary files, crashing the computer, and more [1].

Script is one of the common vehicles for malware. Malware scripts are dangerous since the source codes are opened to everyone. Users can easily modify it to create a variant, or even a new kind. Malware disguise techniques are becoming more complicated and signature-based malware detection is insufficient to detect disguised malware scripts[1, 2]. Therefore, we need to develop a new malware detection algorithm which is robust to disguising.

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.
ACM 978-1-4503-1178-6/12/07.

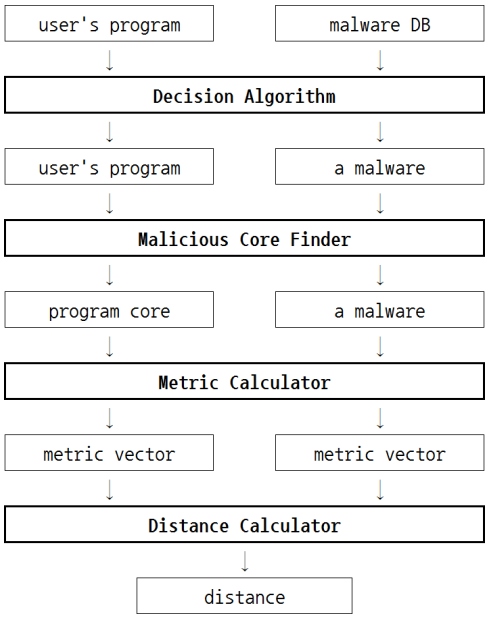

Figure 1: Overview of the proposed system

In this paper, we propose a malware detection algorithm which combines a metric-based method and a genetic algorithm (GA). Metric-based methods use a numerical vector to represent the characteristics of a program, and are known to work well in code plagiarism detection [3]. We adapt this method to detect malware scripts using token frequencies. We also used a hybrid genetic algorithm to find the malicious part of the program. We implemented the above system; the experimental result shows that each idea makes an improvement and the algorithm overall works well.

2. THE PROPOSED SYSTEM

The proposed system consists of four modules, each having slightly different input and output. First, *Decision Algorithm* determines whether the given program is malicious or not. And *Malicious Core Finder* uses a GA and extracts the malicious part of the program which is the most similar to given malware. Afterward, *Metric Calculator* converts programs to numerical vectors containing various metrics. And finally, *Distance Calculator* measures the distance between the vectors. The overview of the system and the relationship between the modules are illustrated in Figure 1.

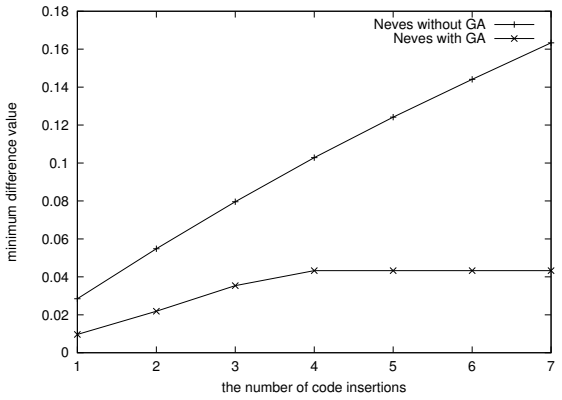

Figure 2: The difference value of generated malware scripts computed by systems without and with GA

3. GENETIC FRAMEWORK

We use a typical hybrid steady-state genetic algorithm.

Representation: We use binary representation. Each gene represents each statement of a file. If the i^{th} gene is set to 0, then the i^{th} statement is regarded as a junk code and is removed before evaluation.

Population: The size of the population is 20. One of the chromosomes is filled with $111\cdots111$ during the initialization process, which represents the entire program without any code removed. The other chromosomes are randomly initialized.

Selection: We use binary tournament selection. The probability that the better individual wins the tournament is set to be 80%.

Crossover and Mutation: We use uniform crossover. For mutation, each gene might be toggled with probability 1%.

Local Optimization Algorithm: We hybridize a local optimization algorithm with the GA. We toggle each of the gene to make a *local move*, and the one with maximum gain is selected. This process is repeated until there is no positive gain.

Replacement: We replace the worst solution of the population with the offspring, even when the offspring is worse than the prior solution.

Stopping Criterion: The GA stops when 90% of the solutions are the same. The GA also stops when it reaches the 10000^{th} generation.

4. EXPERIMENTAL RESULTS

We collected real malware scripts in VBScript language from VX Heavens website[1]. Fifty files have been prepared for the purpose of testing the proposed system, including ten benign codes, twenty known malware variants, and twenty generated variants. For fairness, this process has been conducted by another party who does not participate in the detection project.

Seven variants of the malware Neves were generated by increasingly inserting junk code. These malware variants were tested by the proposed system as well as the one without GA. We computed the difference value, which is defined to be a normalized distance between the variant and the original. We set 0.15 to the threshold for determining a file as a variant of the other, after some significant numbers of experiments.

Figure 2 shows the test results on the seven variants. In case of the system without GA, the difference value proportionally increased with the degree of code insertion. On the other hand, the one with GA showed little difference with respect to the degree of code insertion.The GA turned out to well cope with the junk codes.

We also compared the proposed system with known pieces of anti-virus programs. We used 43 pieces of known anti-virus programs via VirusTotal[2].

Table 1 shows the overall results. The proposed system detected benign files with 100 percent accuracy, which means zero percent false alarm. For existing variants, the proposed system showed 80 percent of detection, while existing anti-virus programs showed 62.8 percent of detection. When it comes to newly generated malware, the existing anti-viruses detected only 34.5 percent of them; on the other hand, the proposed system successfully detected for all of them.

	Proposed system	Anti-viruses
Benign codes	100%	98.36%
Known variants	80%	62.79%
Generated malware scripts	100%	34.54%
Overall	92%	58.58%

Table 1: Detection rate of our system and known anti-virus programs

5. CONCLUSION

In this paper, we proposed a new malware detection system using a hybrid GA and a metric-based method. The GA part was used to remove the junk codes from a program efficiently. Experimental results showed that the proposed ideas helped deal with malware disguising. The proposed system detected most of the files in the dataset, while existing anti-viruses missed a significant portion of them.

6. ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2011-0018006), Brain Korea 21 Project in 2012, and Engineering Research Center of Excellence Program of Korea Ministry of Education, Science and Technology (MEST) / National Research Foundation of Korea (NRF) (Grant 2012-0000463). The ICT at Seoul National University provided research facilities for this study.

7. REFERENCES

[1] J. Aycock. *Computer Viruses and Malware*. Springer, 2006.

[2] N. Idika and A. P. Mathur. A survey of malware detection techniques. Technical Report 286, Purdue University, 2007.

[3] T. Lancaster and F. Culwin. A Comparison of Source Code Plagiarism Detection Engines. *Computer Science Education*, 14:101–112, June 2004.

[1] `http://vx.netlux.org/`

[2] `http://www.virustotal.com/`