マルウェアの類似度による機能推定

小篠 裕子 †　　勝手 壮馬 †　　森井 昌克 †　　中尾 康二 ‡

† 神戸大学大学院工学研究科
657-8501 神戸市灘区六甲台 1-1
{y_ozasa, skatsute}@stu.kobe-u.ac.jp ,
mmorii@kobe-u.ac.jp

‡ 独立行政法人情報通信研究機構
184-8795 東京都小金井市貫井北町 4-2-1
ko-nakao@nict.go.jp

あらまし 一般にマルウェアの大部分は，既知のマルウェアの亜種，もしくはパッキングと呼ばれるコード変換をされた既知のものと同一のマルウェアである．よって大部分のマルウェアは既知のマルウェアとほぼ同一の機能を有すると考えられる．解析対象のマルウェアが，既知のマルウェアの亜種であること，似た機能を有することが判明すれば，新たに解析をする必要はほとんどなく，機能の大部分を推定することが可能となる．本稿では，未知のマルウェアに対して，既に解析されたマルウェアとの類似度を与え，その類似度から未知のマルウェアが有する機能を推定するシステムを提案する．

Estimated Function of Malicious Code by Memory Dump Analysis

Yuko Ozasa†　　Souma Katsute†　　Masakatu Morii†　　Kouji Nakao‡

†Graduate School of Engineering,
Kobe University
1-1 Rokkodai Nada-ku Kobe-shi 657-8501 Japan
{y_ozasa, skatsute}@stu.kobe-u.ac.jp,
mmorii@kobe-u.ac.jp

‡National Institute of Information
and Communications Technology
4-2-1 Nukui-kitamachi Koganei-shi
Tokyo 184-8795, Japan
ko-nakao@nict.go.jp

Abstract There is a problem that the malware analysis does not catch up with the increasing number of the new malware. But the most of all the increased new malware is variant and the functions of the variant is similar to the known malware. In this paper, focus on the fact and propose the system estimates the functions of unknown malware by using the similarity measurement between the known and unknown malware.

1 Introduction

Proliferations of malware poses a major threat to modern information technology. According to a recent report by Symantec [1], every third scan for malware results in almost positive detection. Security of modern computer systems thus critically depends on the ability to keep anti-malware products up-to-date and abreast of current malware developments. This has proved to be a daunting task. Malware has evolved into a powerful instrument for illegal commercial activity, and a significant efforts is made by its authors to thwart detection by anti-malware products. As a result, the number of the new malware variants is increasing.

A great deal of malware that are currently spreading nowadays are produced by copycats

from an original malware strain or from its already spread variants. This proliferation of malware variants not only makes life far more hard from the computer user's point of view and affects the general security of networks but also significantly increases the malware analysist's amount of work.

The research of the malware analysis is grouped into two distinct area: those dealing with static code analysis[2] of unpacked, decrypted executables; and those dealing with dynamic code analysis[3] in a simulated environment. Both approaches have their merits and disadvantages. The scalability of the static code analysis is scaled extremely well, but automated unpacked and decryption is not always possible and will become harder in the future. The dynamic code analysis does not scale as well, but it is superior in characterizing new, unknown malware regardless of its packaging.

To decrease the malware analysist's amount of work, we propose the system which is able to estimate the functions of unknown malware. The approach of the proposed system is differ from the conventional malware analysis systems. We focus on the fact that the most of the malware is its variants and the variants have similar functions. The proposed system estimates the functions of unknown malware using the similarity between the unknown malware and known malware. The similarity is judged by comparing the binary codes of the target and known malware. The malware which has the high similarity is estimated to have the same function of the known malware, so the function of the malware which have high similarity can be estimated.

The most of the malware codes are obfuscated by great variety of packers, for examples, UPX, PEX, FSG, ASPack, Petite. To compare the binary codes between the unknown and known malware, the codes are needed to be unpacked. In order to obtain the unpacked code, we use the particular method of the memory dump [4].

The rest of the paper is organized as follows. Section 2 provides background information on malware and its variants and the conventional malware analysis. Section 3 describes the proposed system which estimates the function of the unknown malware using the similarity between the malware. In Section 4, the experimental result of the proposed system is shown. Section 5 conclude the paper.

2 Malware Analysis

2.1 Malware and its Variants

The code of malware, called malicious code refers to the boad class of software threats to computer systems and networks. It includes any code that modifies, destroys or steals data, allows unauthorized access, exploits or damages a system, or does something that the user doesn't intend to do. Perhaps the most sophisticated types of threats to computer systems are presented by malicious codes that exploit vulnerabilities in applications.

The ease with which it is possible to create new malware variants to infect machines forces human analysts to a great disadvantage. While copycats are able to automatically generate new malware variants in unprecedented numbers, the anti-virus communities focus only on the target malware they ignore the malware variants. It clearly has become infeasible to analyze manually what each of the thousands of new variants appearing each and every hour are capable in detail.

2.2 Malware Analysis

There are mainly two approaches in malware analysis, one is static analysis [2]and the other is dynamic analysis[3].

The static analysis is a white box approach in which the target malware samples is disassembled to enable an analyst to understand its detailed functionalities and code structure. In this analysis, the main problem is often how

to disassemble the executables because most of the malware codes are obfuscated by great variety of packers.

The dynamic analysis is, on the other hand, a black-box approach in which the target malware sample is executed in an environment that is designed to closely observe its internal activities in detail, i.e., server access and scan activity of the malware. A drawback of this analysis is that it only observes a single execution path, however, it still can extract a great deal of information about the behavior of the malware.

3 Estimation of The Function

The same spieces of variants have similar functions. When the similarity between the variants is evaluated, the functions of the variants can be estimated. In this paper, we propose the system estimates the functions of the unknown malware using the similarity measurement.

3.1 Similarity Between Malwares

In this section, let us consider about the measurement of the similarity between malware. The binary codes of the malware are compared, and the similarity between the malware is evaluated by the rate of the matched codes. To the extension of the similarity measurement, API invoked by the malware can be used as substitute for the binary code. The target of the measurement is unknown malware, and the samples using for the measurement are the malware which functions are known.

The comparison between the all binary codes of the target and samples are time consuming. So, the fixed byte sequences are randomly extracted from the target, which are called checking codes, used for the comparison. The comparison are repeated and the number of the matched code between the checking codes and the binary codes of samples are calculated at each time.

The average of the number of the matched codes are regarded as the similarity. In this paper, the number of the checking codes is 50 and their length is 8 byte each, and the comparison is repeated 5 times. These parameters are defined experimentally.

The procedure to calculate the similarity measurement is described as follows.

Step 1: The Extraction of The Checking Code

Extract the fixed byte sequences randomly from the targets for the checking codes. The number of the checking codes are 50 and their length is 8 bytes each. The checking code is randomly chosen except the follow cases:

1. At least the half of the checking code is involved in the other code.
2. The checking codes include the string of $0x00$ or $0xff$ which length is more than the half of them.

In these cases, the checking codes are chosen again randomly. This conditions are defined experimentally.

Step 2: The Comparison Between The Codes

Compare the target code and the sample codes and check whether the matched codes are exist or not. Each checking codes don't have a particular weight and the checking codes are treated equally. When some matched codes are existed, the matched codes are ignored except the code matched first. The similarity S at the one measurement is as follows:

$$S = \frac{N}{M} \cdot 100 \ [\%], \qquad (1)$$

where M is the number of the checking codes, and N is that of the matched codes. The unit of S is %.

Step 3: The Iteration of The Processes

Iterate Step 1 and Step 2 for a specified number of times. After the iteration, calculate the average between the similarities obtained in the each iterations. The average is the final similarity between the malware. To hold the variability caused by the randomness, the extraction in Step 1 is determined randomly at every iterations. The number of the iteration is 5 in this paper.

Step 4: The Output of The Similarity Measurement

Iterate Step 1 to 3, then output the result of the averaged similarity measurement.

After the procedure from Step 1 to 4, the similarity between the unknown and known malware is obtained. The unknown malware which has the high simirality to the known malware is estimated to have the same functions of the known malware. The malware which has almost 0 similarity is estimated to have few functions of known malware, and regarded as the new species of malware.

3.2 Estimation Of Function

The function of the malware is estimated by the similarity obtained by the measurement in Sect. 3.1. The malware which has the high similarity is estimated to have the same function of the known malware.

For estimatiing the functions of the malware, the list of the functions which the sample malware have is needed. The code of the function the malware has and API of the malware is treated as the functions list in this paper. The functions of each malware is weighted, and the sum of the weight is regarded as the score of the malware. If the score of a function exceeds a threshold, we regard the unknown target malware has it. The malware which have a low similarity, almost 0, are disable to estimated the functions. On the other hand, the malware which have a high similarity are estimated to be the same species of the known malware.

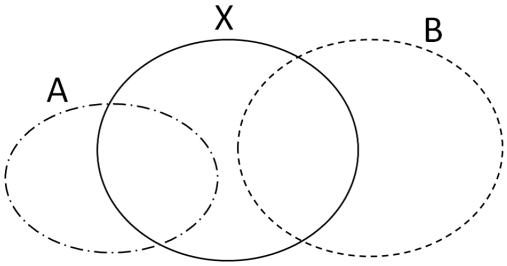

Figure 1: Inclusive Relation of The Malware Function

3.2.1 Interactive Similarity Measurement

The similarity measured in Sect. 3.1 is the one-way similarity. In the other hand, the estimation of the function needed the interactive similarity.

In the case when target malware is X and the sample malware is A, the orthodromic similarity S_{XA} is the similarity measured X toward A and the reversely-oriented similarity S_{AX} is the similarity measured A toward X.

3.2.2 The Weight of The Function

The weight regarded as the score of the functions are calculated by the results of the interactive similarity. Let us consider the case when the unknown malware is X and the sample malware which are the two spieces, A and B, the interactive similarity between the X and A and B have the following relation:

$$S_{XA} = S_{XB}, \quad (2)$$

$$S_{AX} > S_{BX}. \quad (3)$$

The relation between the X, A and B is shown in Fig.1. The hypothesis of Fig.1 is that the rate of the same functions of a malware is proportional to the similarity between the malware. From the Fig.1, the number of the common functions between X and A and B are

equal but the number of the functions in A except the common functions between X and A is smaller than the number of the functions in B except the common functions between X and B. So, the possibility that the number of the functions in A included in X is higher than the number of the functions in B included n X. In order to accurize the estimation in this case, the functions of A should be weighted higher than that of B. So, the weight for the estimation is obtained as follows:

$$W_A = \frac{S_{AX}^2 \cdot \alpha + S_{XA}^2 \cdot \beta}{(\alpha + \beta) \cdot 100}. \quad (4)$$

W_A is the weight of sample A, and α and β is the constants when $\alpha > \beta$.

3.2.3 Estimation of The Function

The functions of the unknown malware are estimated by the weight in Sect. 3.2.2. The weight of the each function is regarded as the score of the each function. For example, when the score of a sample is 30, each functions of a sample have the score, 30.

When the same functions are included in some samples, the sum of the score of the samples become the score of the function. The sum of the score become the score of the each function and used for the function estimation. If the sum of the scores exceeds a threshold, we regard the unknown target malware have the functions. In this paper, the threshold is determinedby the standard deviation of the sum.

4 Experimantal Results

To test the effectiveness of our proposed system estimates the function of the malware, we tested it on 58 malware of several species when $\alpha = 2$ and $\beta = 1$. The functions of the malware used for the experiment are known.

One of the 58 malware is chose for the target and other 57 malware are used as the sample malware. The functions of all 58 malware are estimated by the proposed system. The functions of the malware are known, so the true negative rate and the false positive rate of the proposed system can be obtained.

The true negative rate R_T is obtained by

$$R_T = \frac{F_{X \subset A}}{F_{EX}} \cdot 100. \quad (5)$$

$F_{X \subset A}$ is the number of the common functions of the target malware X and A, and F_{EX} is the number of the functions X has. The average of the true negative rate in this experiment is 60.8%. The minimum true negative rate is 4.9%, and the max false positive rate is 100%.

The number of the false positive N_F shows that of the functions of the target malware X except that of the functions A has. The avarage of the number of the false positive is 7, the max number of that is 17, and the minimum number of that is 0. The number of the 58 malware which number of the false positive is 0 is 8.

The threshold of the estimation is determined by the standard deviation of the sum in this paper. It is clear the false positive rate become higher and the false positive rate become lower when the threshold become lower. The threshold is needed to determined experimentally.

5 Conclusion

In this paper, we propose the system estimates the functions of unknown malware decrease the malware analysist's amount of work. The approach of the proposed system is differ from the conventional malware analysis systems. We focus on the fact that the most of the malware is its variants and the variants have similar functions. The proposed system estimates the functions of unknown malware using the similarity between the unknown malware and known malware. The similarity is evaluated by comparing the binary codes of the target and known malware. The malware which has the high similarity is estimated to have the same

function of the know malware, so the function of the malware which have high similarity can be estimated. The average of the true negative rate of the proposed system is 60.8%. The minimum true negative rate is 4.9%, and the max false positive rate is 100%. The number of the false positive N_F shows that of the functions of the target malware X except that of the functions A has. The avarage of the number of the false positive is 7, the max number of that is 17, and the minimum number of that is 0. The number of the 58 malware which number of the false positive is 0 is 8. The threshold of the estimation is determined by the standard deviation of the sum in this paper. It is clear the false positive rate become higher and the false positive rate become lower when the threshold become lower. The threshold is needed to determined experimentally.

6 Acknowledgment

I would like to thank Dr. Daisuke Inoue and Dr. Masashi Eto for their helpful comments and suggestions.

This work was partly supported by Network Security Incident Response Group of National Institute of Information and Communications Technology.

References

[1] Symantec Internet Security Threat Report http://www.symantec.com/business/theme.jsp?themeid= threatreport, April 2009.

[2] M.Chirstodorescu, S.Jha, "Static Analysis of Executable to Ditect Malicious Patterns," The 12-th USENIX Security Symposium, 2003.

[3] U.Bayer, A.Moser, C.Kruegel, and C.Kirda, "Dynamic Analysis of Malicious Code," Journal in Computer Virology, 2006.

[4] H.Okada, R.Isawa, M.Morii, and Koji Nakao, "An Automated Virus Code Analysis System," *SCIS2007*, 2007.