Malware Similarity Calculation using Improved Smith-Waterman Algorithm

In-Gyeom Cho*, Eul-Gyu Im*
*Dept. of Computer and Software
Hanyang University, Korea,
E-mail : {dlsrua1004, imeg}@hanyang.ac.kr

1. Introduction

Recently variants of malware are increasing rapidly. The variants do the same malicious behaviors but have different internal structures. Classifying these variants according to their behaviors is useful. In this paper, the similarity calculation method for classification is proposed using the Smith-Waterman algorithm, and some improvements are proposed for the existing algorithm.

2. Related Works

The dynamic analysis of malware may use API information. The information of API calls could be processed and used to calculate similarities among malware. For instance, API calls information was defined as 'frequent itemsets' [1] or 'behaviors of the malware' [2], and malware samples were classified. In addition, the malware classification using API calls` sequential properties [3] and detecting variants of malwares by the measuring of similarity in control flow graph related with API calls [4] were proposed. Instead of API calls, malware classification by using the malware`s instruction frequency [5] were proposed also.

3. Proposed methods

3.1. Malware similarity calculation using API sequence analysis

Smith-Waterman algorithm [6] is one of most widely used *sequence alignment* algorithms. In the Bioinformatics field, this algorithm has been used to analyze sequences of DNAs or proteins to identify similar patterns. Sequences of DNAs contain noisy data, and this characteristic is similar to data extracted from malware. Since the sequences of API invocations of malware can be represented similarly as the sequences of DNAs, Smith-Waterman algorithm can be applied to analyze similarities of API invocation sequences of malware.

Smith-Waterman algorithm takes two sequences as inputs. Then the algorithm compares units, or 'tokens', of two sequences. In this research, an API call represent a token, and each API call is represented with 4-byte code, where 2 bytes are used for API's name and the other 2 bytes for API's category. Individual scores for tokens are calculated by comparing these sequences. These scores means how the two sequences are similar. The score is 2 if two tokens are matched, -1 if they are not, and these scores are fundamentally defined in the Smith-Waterman algorithm.

However, the sum of these individual scores cannot be directly used as a similarity score of two sequences. When the Smith-Waterman algorithm calculate the scores, only positive scores can be accumulated. This means that the negative scores cannot affect for the scores, and the longer sequences have the bigger scores. So a different similarity calculation method is needed. We define the similarity as follow:

$$Similarity = \frac{L_p * 2}{(L_a + L_b)/2 * 2}$$

L_p, L_a, L_b mean the maximum length of patterns found and the length of the two sequences, A and B. The range of similarity is 0 to 1, and classification for malware is possible using the similarity.

3.2. Additional improvement

All the APIs belong to a category. Although two API's name are not same, if they belong to the same category, they can be regarded as similar APIs. According to the scoring method in Section 3.1, these two APIs in the same category will get the score -1. This means that these two APIs cannot affect the overall similarity score. So instead of the scoring method in Section 3.1, we added weights w to a new scoring method. We set this weight value to 1.4, after experimenting with some test data. The weights were added to the score of two APIs in the same category. In other words, when two APIs that have the same name and category, or names are not same but in the same category, the weight is added. As a result, the new similarity was defined as follows:

$$Similarity_{new} = \frac{L_{match} * (2 + w) + L_{mismatch} * (-1 + w)}{(L_a + L_b)/2 * (2 + w)}$$

L_{match}, $L_{mismatch}$ mean the number of matched tokens and that of unmatched tokens in the same category, in the longest pattern. The scores for each token were multiplied, then the similarity was calculated.

4. Experiments

The proposed method was tested. The test was carried out for 15 malware samples, i.e. 3 families and 5 malware samples for each family. The test was for the method of Section 3.1 and the proposed method in Section 3.2, and the results are shown as graphs in Figure 1.

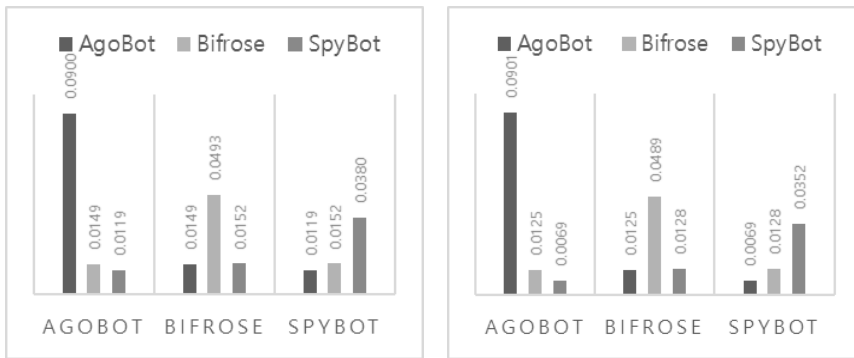

Figure 1. Similarity with no weight (left) and with weight = 1.4 (right)

The graph shows similarity averages of three families. The left one is calculated with the method in Section 3.1, and the right one is with the proposed method. By any methods, each samples have the biggest similarity for their families. But the proposed method`s result shows the larger difference between the similarities of samples in the same family and those in other families. This means that the proposed method has better accuracy for malware classification.

5. Conclusion

This paper proposed the similarity calculation method among malware using Smith-Watermen algorithm. The weight notation was used for improvement, and the value of the weight was set after experiments. As a result of the test, the similarity could be applied for malware classification, and the accuracy was improved using the weight.

The Smith-Waterman algorithm has *O(mn)* time complexity. Therefore, if the sequences compared are too long, the algorithm`s performance should be very low. For this performance problem, some follow-up studies will be carried out.

6. Acknowledgements

This research project was supported by Ministry of Culture, Sports and Tourism(MCST) and from Korea Copyright Commission in 2014.

7. References

[1] QIAO, Yong, et al. Analyzing Malware by Abstracting the Frequent Itemsets in API Call Sequences. In: *Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on*. IEEE, 2013. p. 265-270.

[2] LANGIN, Chet; RAHIMI, Shahram. Soft computing in intrusion detection: the state of the art. *Journal of Ambient Intelligence and Humanized Computing*, 2010, 1.2: 133-145

[3] HAN, Kyoung-Soo; KIM, In-Kyoung; IM, Eul Gyu. Malware Classification Methods Using API Sequence Characteristics. In: *Proceedings of the International Conference on IT Convergence and Security 2011*. Springer Netherlands, 2012. p. 613-626.

[4] HAN, Kyoung-Soo; KIM, In-Kyoung; IM, Eul Gyu. Detection methods for malware variant using api call related graphs. In: *Proceedings of the International Conference on IT Convergence and Security 2011*. Springer Netherlands, 2012. p. 607-611.

[5] HAN, Kyoung Soo; KIM, Sung-Ryul; IM, Eul Gyu. Instruction Frequency-based Malware Classification Method 1. *International Information Institute(Tokyo). Information*, 2012, 15.7.

[6] SMITH, Temple F.; WATERMAN, Michael S. Identification of common molecular subsequences. Journal of molecular biology, 1981, 147.1: 195-197.