ADVANCED MALWARE DETECTION SYSTEM

Zakaryayev Z.N[1]., MSc of Autom., rob. and comp.-int. techn.
State Biotechnological University
Head of the electronics cycle
Military İnstitute named after Haydar Aliyev
Baku, Azerbaijan, zekeryayev.zaur1976@gmail.com
ORCID ID 0009-0007-2764-9749
Zakarya N.Z[2]., BSc of computer science
Azerbaijan Technical University
Baku, Azerbaijan

Abstract: In this article, based on the analysis of the Windows API execution sequences called by PE files, we develop an Intelligent Malware Detection System (IMDS) using classification based on objective-oriented association Analysis (OOA). The Intelligent Malware Detection System is an integrated system, consisting of three main modules: a PE parser, an OOA Rule Generator, and a rule-based classifier.
Keywords: alternative algorithm, malware, database, PE file, OOA mining

System architecture. The spread of malware poses a serious threat to the security of computer systems. Traditional signature-based antivirus systems cannot detect polymorphic and new, previously undetected malicious executable files. The IMDS system runs directly on Windows PE code. PE is designed as a common file format for all versions of the Windows operating system, and PE viruses are the majority of viruses that have been growing in recent years. Some well-known viruses such as CIH, CodeBlue, Nimda, Sircam, Killonce, Sobig and LoveGate all target PE files. The system consists of three main components: a PE parser, an OOA rule generator, and a malware detection module [1, p. 126]. API execution sequence for each benign/malicious executable file. Since a virus scanner is typically a speed-sensitive application, to improve system performance, we developed a PE analyzer to structure the execution sequences of PE file APIs instead of using a third-party disassembler. If a PE file is pre-compressed with a third-party binary compress, it must be decompressed before passing it to the PE analyzer. Through the API request database, the API execution sequence generated by the PE parser can be converted into a group of 32-bit global identifiers that represents the static execution sequence of the corresponding API functions. We then use API calls as signatures from PE files and store them in a signature database, which contains following fields: record ID, PE file name, and file type (“0” represents a benign file while “1” for the precious Mali file), called the API sequence, which is called the API identifier and the general number of the called API function [2, p. 46]. After this, the OOA data mining algorithm is used to create class association rules, which are written to the rules database. To definitively determine whether a PE file is malicious or not, we pass the selected API calls along with the generated rules to the malware detection module to perform classification based on association rules.

Classification based on OOA. Both classification and association analysis play important roles in data mining techniques. Classification is “the task of learning an objective function that maps each set of features to one of the predefined class

labels". There is no predefined purpose for mining association rules. Given a set of transactions in a database, all rules that satisfy the support and trust thresholds will be discovered [3, p.74]. In fact, the analysis of classification and association rules can be integrated into classification based on association rules. This method exploits the properties of frequent patterns to overcome scalability and overfitting issues in classification and achieves excellent accuracy.

OOA Fast FP-Growth Algorithm. Although the A Priori algorithm can be extended to OOA mining, it requires many iterations to generate all frequent itemsets before generating association rules. An alternative OOA mining algorithm called OOA FP-Growth is developed based on the FP-Growth algorithm. Overall, the OOA FP growth algorithm is much faster than the Apriori OOA algorithm for mining frequent itemsets. However, when the minimum support is small, OOA FP-Growth recursively generates a huge number of conditional FP trees, which consumes a lot of time and space. Our malware detection is based on finding frequent patterns from large data collections, so efficiency is an important issue for our system [4, p.36]. In our IMDS system, we extend the modified FP growth algorithm proposed in to perform OOA mining. This algorithm significantly reduces the processing time and memory overhead, and we call it the fast FP growth algorithm. Similar to the OOA FP growth algorithm, it also has two steps of the OOA Fast FP-Growth Algorithm: constructing an OOA Fast FP tree and generating frequent patterns from the tree. But the structure of OOA Fast FP tree is different from the structure of OOA FP tree as follows: The paths of OOA Fast FP tree are directed, and there is no path from the root to the leaves. Thus, fewer pointers and less memory space are required [2, p.136]. In an OOA FP tree, each node is the name of an element, but in an OOA Fast FP tree, each node is an element's ordinal number, which is determined by the element's support count.

Conclusion. This article describes a malware detection system based on object-oriented association (OOA) mining algorithms, a sequence of window APIs. Thus, the main idea is the following points. Development of an integrated intelligent malware detection system based on analysis of Windows API execution sequences. The system consists of three components: a PE parser, a rule generator, and a classifier, adapting existing association-based classification methods to improve the efficiency and effectiveness of the system.

References

1. Cheng H., Yan X., Han J. and Hsu C. Discriminative frequent pattern analysis for effective classification. In ICDE-07, 2007.
2. Fan M. and Li C. Mining frequent patterns in an fp-tree without conditional fp-tree generation. Journal of Computer Research and Development, 40:1216–1222, 2003.
3. Shen Y., Yang Q. and Zhang Z. Objective-oriented utility-based association mining. In Proceedings of IEEE International Conference on Data Mining, 2002.
4. Tan P., Steinbach M. and Kumar V. Introduction to data mining. Addison Wesley, 2005.