A Survey on Data Mining Methods for Malware Detection

Ms. Shital Balkrishna Kuber

ME (II), Computer,
Vidya Pratishthan's College Of Engg,
Baramati, Maharashtra, India
kuber182@gmail.com

Abstract—Malware is any type of malicious software that has the capability to enter into system without authorization of the users. Thus malware detection is the important issue in the computer security.

Signature based detection is more popular method to detect the malware attack but main drawback of this method is that it is not used to detect the Zero-day attack. We need to update the database regularly and human experts are needed to create the new signature. The drawbacks of Signature based malware detection is minimized by using heuristic method. Heuristic method is used to detect zero-day attacks. There are various methods used to detect the malware like n-gram method, Finite state automaton method, Control Flow Graph method, N-gram analysis at byte level etc. These methods having their various advantages and disadvantages. This study enlightens the various methods used to detect malwares.

Keywords— malware, n-gram, opcode, Signature-based detection, Anomaly-based detection, Specification-based detection. Disassemble process.

INTRODUCTION

Malware is communal term for any malicious software which enters system without authorization of the users. Modern communication infrastructures are highly susceptible to many types of malwares attacks. Due to malicious attacks cause several damage to private users, governmental organizations and commercial companies. The explosion in high-speed internet connections facilitates malware to propagate and infect computer system very rapidly. Once the malware system finds its way into the system, it scans the system and find out the vulnerabilities of operating system. Then perform inadvertent actions on the system finally slowing down the overall performance of the system. In every year the malwares are increasing in an alarming rate. Therefore malware detection becomes a vital issue in today's computer systems.

Malware detection techniques

Malware detection technique is the technique used to detect or identify the malware. Generally, malware detection technique can be categorized into three types Signature-based Anomaly-based and Specification-based detection.

Signature-based malware detection

It maintains the database of signature and detects malware by comparing pattern against the database. It shall require some amounts of system resources to detect the malware also this technique can detect the known malware accurately. The disadvantage of this technique is it not effective against the Zero-day attack so it cannot detect the new, unknown malware as no signature available for such type of malware. Data mining and machine learning techniques are used to overcome this limitation of signature based detection.

Most of the antivirus tools are based on signature based detection techniques. These signatures are created by examining the disassembled code of malware binary files. Various disassemblers and debuggers like IDA Pro, ollydbg, WinDb32 are available which help in disassembling the executables. Disassembled code is analyzed and features are extracted. These features are used in constructing the signature of particular malware family.

Heuristic-based malware detection

It is also called as anomaly based detection. Here mainly the goal is to analyze the behavior of known or unknown malwares. Behavioral parameters include various factors such as source/ destination address of malwares, different types of attachments and other measurable statistical features. It usually occurs in two phase:

1. Training (learning) phase
2. Detection (monitoring) phase.

During the training the behavior of the system is observed in the absence of attack and machine learning technique is used to create a profile of such normal behavior. In detection phase, this profile is compared against the current behavior, and deviations are flagged as potential attacks. A key advantage of anomaly based detection is its ability to detect zero-day attacks. Zero-day attacks are attacks that previously unknown to the malware detector.

Specification based detection

Specification-based detection is a derivative of anomaly based detection that tries to defeat the typical high false alarm rate associated with the anomaly-based detection. Specification-based detection relies on program specifications that describe the intended behavior of security-critical programs. It monitors executions program involve and detecting deviation of their behavior from the specification, rather than detecting the occurrence of specific attack patterns.

This technique is similar to anomaly detection where they detect the attacks as deviate from normal. The difference is that instead of relying on machine learning techniques, it will be based on manually developed specifications that capture legitimate system behavior. It can be used to monitor network components or network services that are relevant to security, Domain Name Service, Network File Sharing and routers**.**

LITERATURE SURVEY

D. Bilar [1] investigated opcode frequency distributions as a means to identify and differentiate malware. They discuss a malware detection mechanism through statistical analysis of opcode distribution. His results shows that most recurrently occurring opcodes are not a good indicator of malware like move, push, call etc. While, less recurrently occurring opcodes are a good indicator of malware like add, sub, ja, adc etc.

Advantages

1. His Technique gives a preliminary assessment of its usefulness for malware detection.
2. This Technique gives better accuracy for differentiation of modern (polymorphic and metamorphic) malware.

Disadvantage

In this technique, the dynamic approach is not taken into the consideration.

D. Bilar [2] analyzes the call graph structure of 120 malware and 200 non malicious executable files. He treat each executable file as a graph of graphs. This follows the intuition that in any procedural language, the source code is structured into functions; these functions can be viewed as a flowchart, i.e. a directed graph. These functions call each other, thus creating a larger graph where each node is a function and the edges are calls-to relations between the functions. This larger graph is called as the callgraph. The structure of callgraph is recovered by disassembling the executable into individual instructions. He distinguishes between short and far branch instructions: Short branches do not save a return address while far branches do. Intuitively, short branches are generally used to pass control around within one function of the program, while far branches are used to call other functions. He statically generates the CFG of benign and malicious code.

He compared the basic block count for benign and malicious code. Bilar concluded that malware tends to have lower basic block count. The CFG of malicious file have less interaction, fewer branches and limited functionality. On the other hand, the benign files tend to have more block count with complex interaction.

Advantage

He proposed the new approach i.e. CFG construction for detecting the malware.

Disadvantage

There is a space overhead for storing the information of CFG.

R. Sekar et al. [3] implemented a Finite State Automaton (FSA) approach. FSA-learning is computationally costly, or that the space usage of the FSA may be excessive. They present a new approach in this paper that overcomes these difficulties. Their approach builds a compact FSA in a fully automatic and skilled manner, and without requiring access to source code for programs. They compared the FSA approach with n-gram analysis method.

Advantages

1. They found that the false positive rate of the FSA algorithm is low than the n-gram approach.
2. The space and runtime overhead of FSA learning is minimal.
3. FSA approach can detect a wide range of malware attacks.
4. The training periods needed for FSA based approach are shorter.
5. FSA-technique can capture both small term and lengthy term temporal relationships among system calls, and thus perform more precise detection.
6. The FSA uses only a constant time per system call during the learning as well as detection period. This factor leads to low overheads for intrusion detection.

Disadvantage
FSA algorithm does not preserve the order of system calls made from libraries.

Wei-jen Li et al. [4] describe N-gram (N=1) analysis, at byte level, to compose models derived from learning the file types that the system intends to handle. Li et al. perform an N-gram analysis at byte level (N=1) on PDF files with embedded malware.

Advantages
1. This technique proved an effective technique for detecting malicious PDF files.
2. This technique detects the malware embedded at the beginning or end of a file.

Disadvantages
1. This technique is failed to detect malware embedded in the middle of the file.
2. Li et al. focused on n-gram analysis only with n=1.but does not perform analysis on n=2, n=3 and so on.

Santos et al. [5] demonstrated that n-gram signatures based approach to detect unknown malware. They found that for n=2, the detection rate is low, for n=4, the detection rate is maximum. In this paper they use a new methodology for malware detection based on the use of n-grams for file signatures creation. They tackle the issue of dealing with false positives using a parameter named 'd' to control how strict the method behaves to classify the instance as malware or benign software in order to avoid false positives.

Santos et. al [6] proposed the use of a single-class learning method for unknown malware detection based on opcode sequences. This method is based on examining the frequencies of the appearance of opcode sequences to build a machine-learning classifier using only one set of labeled instances within a specific class of either malware or genuine software. They performed an empirical study that shows that this method can reduce the effort of labeling software while maintaining high accuracy.

Advantage
Single-class learning needs several instances that belong to a specific class to be labeled.
Therefore, Single-class learning method can reduce the cost of unknown malware detection.

Disadvantage
This method cannot be able to detect packed executable.

Shabtai et al. [7] used static analysis to study the effectiveness of malware detection. For this purpose they used different n-gram size (N=1 to 6) with various classifiers. Shabtai's findings showed that N=2 performed best. To detect the unknown malicious code they used opcode n-gram patterns as feature extraction technique, feature selection method and learning algorithm. They use OpCode n-gram patterns, generated by disassembling the executable files of both benign and malware files.

Advantages
1. They found that malware detection rate is high for n=2 while previous study Santos et. al found that for n=4 the detection rate is high. This is new investigation regarding the n-gram size.
2. The class imbalance problem is taken into consideration.

Disadvantages
1. Generally in textual domain TFIDF is more successful representation for the retrieval and categorization purposes but they found that TFIDF representation introduces additional computational challenges in the maintenance of the collection.
2. TFIDF representation has no added value over the TF representation.

CONCLUSION

Malware has a long history of evolutionary development as the war between the anti-malware researchers and the malware writers has progressed. This study presents the different malware detection methods in data mining. This Survey focus on various approaches used in current antivirus system like signature based malware detection and heuristic based malware detection. This study also enlighten the different methods like n-gram analysis at byte level, single-class learning method, Finite State Automaton (FSA) method used to detect the malwares.

REFERENCES:

[1] D. Bilar, "Opcodes as predictor for malware, Int. J. Electron. Security Digital Forensics, vol. 1, no. 2, pp. 156 - 168, 2007. D. Bilar, "Callgraph properties of executables and generative mechanisms, AI Commun., Special Issue on Network Anal. in Natural Sci.and Eng., vol. 20, no. 4, pp. 231-243, 2007.

[2] D. Bilar, “Callgraph properties of executables and generative mechanisms,” AI Commun., Special Issue on Network Anal. in Natural Sci.and Eng., vol. 20, no. 4, pp. 231-243, 2007.
[3] R. Sekar, M. Bendre, D. Bollineni, and Bollineni, R. Needham and M.Abadi, Eds., “A fast automaton-based method for detecting anomalous program behaviors,” in Proc. 2001 IEEE Symp. Security and Privacy, IEEE Comput. Soc., Los Alamitos, CA, USA, 2001, pp. 144-155.
[4] Wei-Jen Li, W. L. K. Wang, S. Stolfo, and B. Herzog, “Fileprints: Identifying file types by n-gram analysis,” in Proc. 6th IEEE Inform. Assurance Workshop, Jun. 2005, pp. 64-71.
[5] I. Santos, F. Brezo, J. Nieves, Y. K. Penya, B. Sanz, C. Laorden, and P. G. Bringas “Opcode-sequence-based malware detection,” in Proc.2nd Int. Symp. Eng. Secure Software and Syst. (ESSoS), Pisa, Italy, Feb.3-4, 2010, vol. LNCS 5965, pp. 35-43.
[6] I. Santos, F. Brezo, B. Sanz, C. Laorden, and Y. P. G. Bringas, “Using opcode sequences in single-class learning to detect unknown malware,” IET Inform. Security, vol. 5, no. 4, pp. 220227, 2011.
[7] A. Shabtai, R. Moskovitch, C. Feher, S. Dolev, and Y. Elovici, “Detecting unknown malicious code by applying classification techniques on opcode patterns, Security Informatics, vol. 1, pp. 1-22, 2012.
[8] Vinod P., V. Laxmi, M. S. Gaur , “Survey on Malware Detection Methods.”
[9] R. Moskovitch, C. Feher, N. Tzachar, E. Berger,M.Gitelman, S.Dolev,and Y. Elovici, “Unknown malcode detection using opcode representation,” in Proc. 1st Eur Conf. Intell. and Security Informatics (EuroISI08), 2008, pp. 204-215