INTELLIGENT MALWARE DETECTION SYSTEM

Sandeep B. Damodhare*

Prof. V. S. Gulhane**

ABSTRACT

Malicious programs spy on users' behavior and compromise their privacy. Unfortunately, existing techniques for detecting malware and analyzing unknown code samples are insufficient and have significant shortcomings. We observe that malicious information access and processing behavior is the fundamental trait of numerous malware categories breaching users' privacy (including key loggers, password thieves, network sniffers, stealth backdoors, spyware and root kits), which separates these malicious applications from benign software. Commercial anti-virus software is unable to provide protection against newly launched ("zero-day") malware. In this dissertation work, we propose a novel malware detection technique which is based on the analysis of byte-level file content. The proposed dissertation work will demonstrate the implementation of system for detection of various types of malware.

* ME Student, Dept of IT, SIPNA's College of Engineering & Technology, Amravati (MS) INDIA

** Associate Professor, Dept of CSE, SIPNA's College of Engineering & Technology, Amravati (MS) INDIA

INTRODUCTION

Malicious software (i.e., Malware) creeps into users' computers, collecting users' private information, wrecking havoc on the Internet and causing millions of dollars in damage. Malware detection and analysis is a challenging task, and current malware analysis and detection techniques often fall short and fail to detect many new, unknown malware samples. Current malware detection methods in general fall into two categories: signature-based detection and heuristics based detection. The former cannot detect new malware or new variants.

The latter are often based on some heuristics such as the monitoring of modifications to the registry and the insertion of hooks into certain library or system interfaces. Since these heuristics are not based on the fundamental characteristics of malware, they can incur high false positive and false negative rates. For example, many benign software access and modify registry entries. Hence, just because an application creates hooks in the registry does not mean that it is malicious (i.e., the application could be a useful system utility). Furthermore, to evade detection, malware may attempt to hook library or system call interfaces that the detector does not monitor. Even worse, since many rootkits hide in the kernel, most such heuristics-based detectors cannot detect them as they do not necessarily modify any visible registry entries or library or system call interfaces.

Malware may be easily transmitted among machines as (P2P) network shares. One possible stealthy way to infect a machine is by embedding the malicious payload into files that appear normal and that can be opened without incident. A later penetration by an attacker or an embedded Trojan may search for these files on disk to extract the embedded payload for execution or assembly with other malcode. Or an unsuspecting user may be tricked into launching the embedded malcode in some crafty way. In the latter case, malcode placed at the head of a PDF file can be directly executed to launch the malicious software. Social engineering can be employed to do so. One would presume that an AV scanner can check and detect such infected file shares if they are infected with known malcode for which a signature is available. The question is whether a commercial AV scanner can do so. Will the scanning and pattern-matching techniques capture such embeddings successfully? An intuitive answer would be "yes". Malware is software designed to infiltrate or damage a computer system without the owner's informed consent (e.g., viruses, backdoors, spyware, trojans, and worms) [1]. Numerous attacks made by the malware pose a major security threat to computer users. Hence, malware detection

is one of the computer security topics that are of great interest. Currently, the most important line of defense against malware is antivirus programs, such as Norton, MacAfee, and Kingsoft's Antivirus. These widely used malware detection software tools use signature-based method to recognize threats. Signature is a short string of bytes, which is unique for each known malware so that future examples of it, can be correctly classified with a small error rate. However, this classic signature-based method always fails to detect variants of known malware or previously unknown malware, because the malware writers always adopt techniques like obfuscation to bypass these signatures [2]. In order to remain effective, it is of paramount importance for the antivirus companies to be able to quickly analyze variants of known malware and previously unknown malware samples. Unfortunately, the number of file samples that need to be analyzed on a daily basis is constantly increasing [3]. According to the virus analysts at Kingsoft Antivirus Laboratory, the "gray list" that is needed to be analyzed per day usually contain more than 70 000 file samples. Clearly, there is a need for an automatic, efficient, and robust tool to classify the "gray list."

RELATED WORK

So far, several data mining and machine-learning approaches have been used in malware detection [4]. Since frequent item sets found by association mining represent the underlying profiles (of application programming interface (API) function calls) of malware and benign files, we developed an intelligent malware detection system (IMDS) adopting associative classification method based on the analysis of API calls. In order to overcome the disadvantages of the widely used signature-based malware detection method, data mining and machine-learning approaches are proposed for malware detection [5]. Naive Bayes method, SVM, and decision tree classifiers are used to detect new malicious executables in previous studies [6]. Associative classification, as a new classification approach integrating association rule mining and classification, becomes one of the significant tools for knowledge discovery and data mining [7].

Due to the fact that frequent item sets (sets of API calls) discovered by association mining can well represent the underlying semantics (profiles) of malware and benign file datasets, associative classification has been successfully used in the IMDS system developed in [8] and

[9] for malware detection. However, there is often a huge number of rules generated in a classification association rule mining practice [10]. It is often infeasible to build a classifier using all of the generated rules. Hence, how to reduce the number of the rules and select the effective ones for prediction is very important for improving the classifier's ACY and efficiency. Recently, many post-processing techniques, including rule pruning, rule ranking, and rule selection have been developed for associative classification to reduce the size of the classifier and make the classification process more effective and accurate [11]. It is interesting to know how these post-processing techniques would help the associative classifiers for malware detection.

A wide range of host-based solutions have been proposed by researchers and a number of commercial anti-virus (AV) software is also available in the market [5]. These techniques can broadly be classified into two types: (1) static, and (2) dynamic. Static techniques mostly operate on machine-level code and disassembled instructions. In comparison, dynamic techniques mostly monitor the behavior of a program with the help of an API call sequence generated at run-time. The application of dynamic techniques in AV products is of limited use because of the large processing overheads incurred during run-time monitoring of API calls; as a result, the performance of computer systems significantly degrades. In comparison, the processing overhead is not a serious concern for static techniques because the scanning activity can be scheduled offline in an idle time. Moreover, static techniques can also be deployed as an in-cloud network service that moves complexity from an end-point to the network cloud [8]. Almost all static malware detection techniques including commercial AV software — either signature-, or heuristic-, or anomaly-based — use specific content signatures such as byte sequences and strings. A major problem with the content signatures is that they can easily be defeated by packing and basic code obfuscation techniques [3]. In fact, the majority of malware that appears today is a simple repacked version of old malware [4]. As a result, it effectively evades the content signatures of old malware stored in the database of commercial AV products. To conclude, existing commercial AV products cannot even detect a simple repacked version of previously detected malware.

PROPOSED WORK

In the proposed dissertation work intelligent malware detection system will be implemented. The dissertation work will be carried out as follows.

1. Analysis of available malware detection systems.
2. Evaluation of how these systems complement each other to improve detection rates.
3. Implementation of malware detection system for detection of denial of service and backdoor.
4. Analysis of malware detection results.

SYSTEM REQUIREMENTS

Operating System: Open Source OS / Windows XP (SP2 or SP3)

Development Tool: .net/C/C++/VC++

REFERENCES

[1]. M. Antonie and O. Zaiane, "An associative classifier based on positive and negative rules," in *Proc. 9th ACM SIGMOD Workshop Res. Issues Data Mining Knowl. Discovery*, 2004, pp. 64–69.

[2]. D. Brumley, C. Hartwig, M. G. Kang, Z. Liang, J. Newsome, D. Song, and H. Yin. BitScope: Automatically dissecting malicious binaries. Technical Report CMU-CS-07-133, School of Computer Science, Carnegie Mellon University, March 2007.

[3]. D. Brumley, C. Hartwig, Z. Liang, J. Newsome, D. Song, and H. Yin. Botnet Analysis, chapter Automatically Identifying Trigger-based Behavior in Malware. 2007.

[4]. M. Costa, J. Crowcroft, M. Castro, A. Rowstron, L. Zhou, L. Zhang, and P. Barham. Vigilante: End-to-end containment of internet worms. In Proceedings of the 20th ACM Symposium on Operating Systems Principles (SOSP'05), October 2005.

[5]. J. R. Crandall and F. T. Chong. Minos: Control data attack prevention orthogonal to memory model. In Proceedings of the 37th International Symposium on Microarchitecture (MICRO'04), December 2004.

[6]. M. Egele, C. Kruegel, E. Kirda, H. Yin, and D. Song. Dynamic Spyware Analysis. In Proceedings of the 2007 Usenix Annual Conference (Usenix'07), June 2007.

[7]. P. Ferrie. Attacks on virtual machine emulators. Symantec Security Response, December 2006.

[8]. H. Cheng, X. Yan, J. Han, and P. S. Yu, "Direct discriminative pattern mining for effective classification," in *Proc. ICDE-2008*, pp. 169–178.

[9]. H. Cheng, X. Yan, J. Han, and C. Hsu, "Discriminative frequent pattern analysis for effective classification," in *Proc. ICDE-2007*, pp. 716–725.

[10]. M. Christodorescu, S. Jha, and C Kruegel, "Mining specifications of malicious behavior," in *Proc. ESEC/FSE-2007*, pp. 5–14.

[11]. F. Coenen and P. Leng, "An evaluation of approaches to classification rule selection," in *Proc. 4th IEEE Int. Conf. Data Mining 2004*, pp. 359–362.

[12]. X. Jiang and X. Zhu, "vEye: Behavioral footprinting for self-propagating worm detection and profiling," *Knowl. Inf. Syst.*, vol. 18, no. 2, pp. 231– 262, 2009.