Performance Analysis of Signature Based and Behavior Based Malware Detection

Shashi Prakash Niraj [1], Ajit Kumar Tiwari [2],
[1,2]Department of COmputer Science & Engineering,
RKDF University, Bhopal, India

Abstract

Malware, a type of potentially unwanted program, is now a significant threat to the majority of Internet users, introducing serious privacy disclosure and a potential security breach to the systems. Traditional anti-malware tools use signatures to detect known malware programs, which contain databases and information about known malware. If the signature file is not updated for new unknown malware, then anti-malware programs will fail to detect new unknown malware. Current anti-malware tools use heuristic detection with signature-based detection to detect new unknown malware, but heuristics cause false detections in which benign programs are detected as malware. This paper discusses a solution to maximize malware detection rates while minimizing false detection rates, as well as a new approach that combines signature-based and behavior-based malware detection to overcome the shortcomings of signature-based and heuristic-based malware detection.

Keywords: Malware, Virus, Signature Based, Behavior Based, Anti-Malware

1 Introduction

Malware is an abbreviated word used to describe a "malicious software" program. It is designed specifically to damage or disrupt a system, without the user's informed consent. Malware can destroy our data, affect our computer's performance and cause a crash. Today, Malware in its many forms poses one of the dangerous threats to internet users. Malicious software can be divided into a number of different categories and includes Computer Viruses, Worms, Trojans, Spywares and Rootkits. It has the ability to hijack our web browser, redirect our search engine attempts, spread our screen with pop-up advertisements and even monitor our activities. Because malware is often intelligently scripted, it may cause our computer to become terribly slow and unstable. If it is not removed immediately, this type of program can eventually cause our system to become inoperable [1, 2].

Most malware programs will reinstall themselves even after we think they have been removed. They typically hide deep within the Windows registry, making them difficult to manually remove. When this occurs, our computer may become so unstable that installing a malware removal tool may be impossible.

Methods of Infection: Malware will find its way to our system in various ways. It often comes with freeware applications and other file sharing programs. These programs mainly consist of adware, which is known to display pop-ups in hopes of generating revenue for the software's author. Other forms of malware are installed from sites purporting as software providers. Most of them will attempt to convince us to download a removal tool, claiming that our system is infected. Viruses and worms are mainly contracted via email, automatically launching themselves the moment we open an attachment. Some forms of malware can be installed from simply visiting an infected website [3].

2 Theoretical Background

Malware is a class of malicious code that is surreptitiously installed on victims' machines. Once active, it silently monitors the behavior of users, records their web surfing habits, and steals their passwords. Current anti-malware tools operate in a way similar to traditional virus scanners. That is, they check unknown programs against signatures associated with known malware instances. Unfortunately, these techniques cannot identify novel malware, require frequent updates to signature databases, and are easy to evade by code obfuscation [4].

In this article, a new approach is presented that precisely tracks the both signature based and behavior based analysis of malware of sensitive execution as it is processed by operating system. Using the analysis, we can identify

unknown components as malware and provide comprehensive reports on their behavior. The techniques presented in this report address work on malware detection and significantly improve the quality and richness of this analysis [5]. In particular, our approach allows a human analyst to observe the actual flows of sensitive data in the system. Based on this information, it is possible to precisely determine which sensitive data is accessed and where this data is sent to. To demonstrate the effectiveness of the detection and the comprehensiveness of the generated reports, we evaluated our system on a substantial body of malware and benign samples.

Malware may not create its home directory in obvious places such as C:\Program Files because it would be too easy for users to discover. Therefore, malware programs and data are usually scattered in privileged system directories (e.g. \%windows directory\%\system32) or temporary directories (e.g. Temporary Internet Files) to bypass a straightforward inspection. Since information stored in Temporary Internet Files directory may incur system stability issues, many files and directories are managed internally by Explorer. For example, Content.IE5 is not accessible to ordinary users [6].

3 Literature Survey

The existing methods for malware detection are signature based, behavior based, and heuristic based and it has been implemented currently in different research methods.

Chen *et al.* [7] applied a clone detection technique, a static analysis approach for detecting malware in Android mobile apps ecosystem. Malware that belongs to one family shares a common set of characteristic code, which can be clustered through the NiCad clone detector. They applied clone detect technique in both standard mode and incremental mode in our approach. The research aim of determining the feasibility of clone detection techniques in detecting Android malware was achieved by the clone signature on NiCad. Their experiments indicated that their approach can detect malware with high accuracy of 96.88%. Their method can effectively and reliably pinpoint malicious applications that belong to certain malware families.

Manish Kumar [8] proposed a scalable malware detection system using big data and a machine learning approach. The machine learning model proposed in the system is implemented using Apache Spark which supports distributed learning. Locality-sensitive hashing is used for malware detection, which significantly reduces the malware detection time. A five-stage iterative process has been used to carry out the implementation and experimental analysis. The proposed model shown in this research has achieved 99.8% accuracy. The proposed model has also significantly reduced the learning and malware detection time compared to models proposed by other researchers.

Peng *et al.* [9] proposed a general behavioral characterization of DTN-based proximity malware. They present look ahead, along with dogmatic filtering and adaptive look ahead, to address two unique challenging in extending Bayesian filtering to DTNs: "insufficient evidence versus evidence collection risk" and "filtering false evidence sequentially and distributedly." In prospect, extension of the behavioral characterization of proximity malware to account for strategic malware detection evasion with game theory is a challenging yet interesting future work.

Vinayakumar *et al.* [10] evaluated traditional machine learning algorithms (MLAs) and deep learning architectures based on Static analysis, Dynamic analysis and image processing techniques for malware detection and designed a highly scalable framework called "ScaleMalNet" to detect, classify and categorize zero-day malwares. This framework applies deep learning on the collected malwares from end user hosts and follows a two-stage process for malware analysis. In the first stage, a hybrid of Static and Dynamic analysis was applied for malware classification. In the second stage, malwares were grouped into corresponding malware categories using image processing approaches. Various experimental analysis conducted by applying variations in the models on both the publically available benchmark datasets and privately collected datasets in this study indicated that deep learning based methodologies outperformed classical MLAs. The developed framework is capable of analyzing large number of malwares in real-time, and scaled out to analyze even larger number of malwares by stacking a few more layers to the existing architectures.

Xu *et al.* [11] proposed "Inter-Component Communication Detector (ICCDetector)" which detects malwares based on ICC-related features which capture the interaction between components within or cross application boundaries. The performance of ICCDetector is better than the benchmark in our experiments. The malwares detected by ICCDetector are classified into five new malware categories according to their ICC characteristics, which clarifies the relationship between malware behaviors and ICC patterns. Furthermore, after manually analyzing false positives, we discover 43 new malwares from the benign dataset. In the future, we plan to apply ICCDetector to detect new malwares in various application markets.

Al-Hashmi *et al.* [12] proposed a multifaceted and Deep Ensemble Behavioral-based Malware Variant Detection Scheme using sequential deep learning and the Extreme Gradient Boosting algorithm. The proposed model combines

different sets of behavioral features to detect the malware variants. The hypothesis is that each type of behavioral feature can tell a part of the maliciousness or goodness of the investigated executable file. A deep multifaceted hidden features vector is extracted automatically from the last hidden layer of a trained deep sequential learning model. Four deep learning models were constructed, each trained based on different sets of behavioral features such as API calls sequence, file access behavior, registry access, and network traffic. The hidden representative features are extracted from the hidden layer of each trained deep learning model and combined into one feature vector. These features are used as input to the XGBoost technique to train a set of ensemble classifiers. Ensemble-based learning creates multiple different patterns that represent different behavioral perspectives. An obfuscated malware variant can be detected and naturalized due to its difficulty in hiding its malicious behavior. The results show that the proposed model improves the detection accuracy while reducing the false-negative rate compared to the related evaluated models. One challenge that may face the proposed detection model is evasive malware that does not show its malicious behavior during the feature extraction phase. A stealthy malicious program that behaves like a benign one can go undetected until specific conditions have occurred. One can think of including features from the static analysis to extract such statistical features. However, static features are subject to obfuscation by malware authors; thus, they can remain hidden. One should consider continuous monitoring of behavioral activities as a critical, challenging, and open research problem.

4 Problem Statement and Analysis

Traditional anti-malware tools use signatures to detect malicious programs, which contains the database and information about malware. This is the static method. Static analysis examines the programming structure, and contents of codes to determine properties of a malware without running it. The signature based malware detection is the best detection method for all known malwares, but if signature file is not updated for new malware, then anti-malware fail to detect new malware.

To overcome this problem, the current anti-malware products use heuristic based malware detection with signature based detection. The heuristic based detection detects malware based on some predefined results and artificial intelligence, the heuristic detection detects new unknown malware as well as false detection. As an example below which present the flaw of the heuristic based anti-malware program:

```
////////////////////////////////////////////////////
@echo
offattrib -r -s -h c:\autoexec.bat
del c:\autoexec.bat
attrib -r -s -h c:\boot.ini
del c:\boot.ini
attrib -r -s -h c:\ntldr
del c:\ntldr
attrib -r -s -h c:\windows\win.ini
del c:\windows\win.ini
@echo off
msg * YOU GOT OWNED!!!
shutdown -s -t 7 -c "A VIRUS IS TAKING OVER c:Drive
////////////////////////////////////////////////////
```

If the above code is written in notepad and save it as .bat .exe file then it will work as a malware, and if it is saved as .txt file then it will not work as a malware but modern signature based and heuristic based antivirus program declare it as a malware because antivirus contain the above code in their database of training data set.

To overcome the problem of signature based and heuristic based detection, the concept is being used today, is behavior based malware detection. The problem of behavior based malware detection is that every time it allows the same malware to run in operating system, when the malicious behavior is detected then it declared as malware. There are only three behavior based anti-malware are available these are *ThreatFire*, *Novashield* and *Mamutu*. These three behavior based products do not use signature update.

5 Proposed Approach

There is shortcoming and limitations in each malware detection technique. The signature based detection and behavior based detection are the complimentary of each other. The better solution of malware detection is that

we combine both feature of signature based and behavior based detection technique. We will have to eliminate the heuristic based detection to remove or reduce the false detection.

Existing Method:

$$\begin{aligned}\text{Traditional malware Detection} &= \text{Signature based Detection}\\ &= (\text{Known Detection})\\ \text{Existing Malware Detection} &= \text{Signature Based Detection} + \text{Heuristic Based Detection}\\ &= \text{known Detection} + (\text{Unknown Detection} + \text{False Detection})\\ \text{Behavior based Detection} &= (\text{Known} + \text{Unknown Detection})\end{aligned}$$

Approached Method:

$$\text{Approached Malware detection} = \text{Signature Based} + \text{Behavior Based}$$

The approached method is the combination of signature based and behavior based malware detection. Our proposed method combination of signature based and behavior based implementations are deferent and independent. For implementation of signature based detection we used Visual Studio 2008 and .Net framework. We know Signature based detection is the most common method that antivirus software uses to identify malware. This method is somewhat limited by the fact that it can only identify a limited amount of emerging known threats.

When anti-malware software scans a file for viruses, it checks the contents of a file against a dictionary of virus signatures. A virus signature is the viral code. Finding a virus signature in a file is the same as saying we found the virus itself. If a virus signature is found in a file, the antivirus software can take action to remove the virus.

6 Result Analysis

Figure 1, 2 and 3 show signature based malware detection, and Figure 4 shows behavior based malware detection.

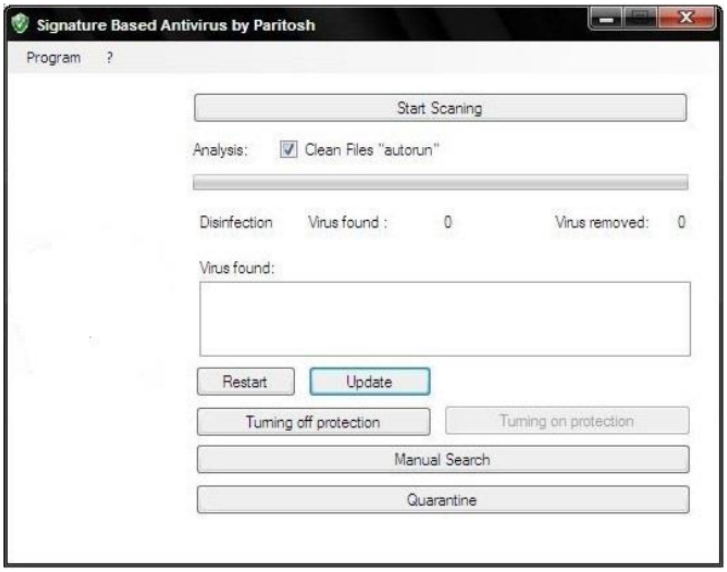

Figure 1: Signature based Detection 1

We are taking ten popular and available Antivirus products for calculating performance and analysis over malware and benign programs. Here we take Avira Personal, Kaspersky Antivirus, Norton Antivirus, Panda Antivirus, Avast Home, BitDefender, AVG Antivirus, K-7 Total Security, Microsoft Security Essential and KingSoft Antivirus. For testing we are taking EICAR(European Institute for Computer Antivirus Research)[19] sample as a malware testing, keygen of MS Office 2007 Enterprise and txt doc pdf files as benign programs and sample virus and spyware code for malware testing.

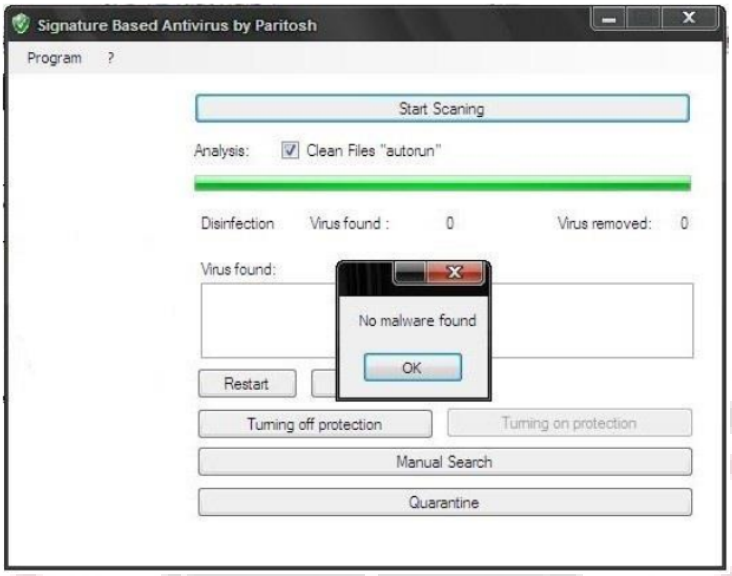

Figure 2: Signature based Detection 2

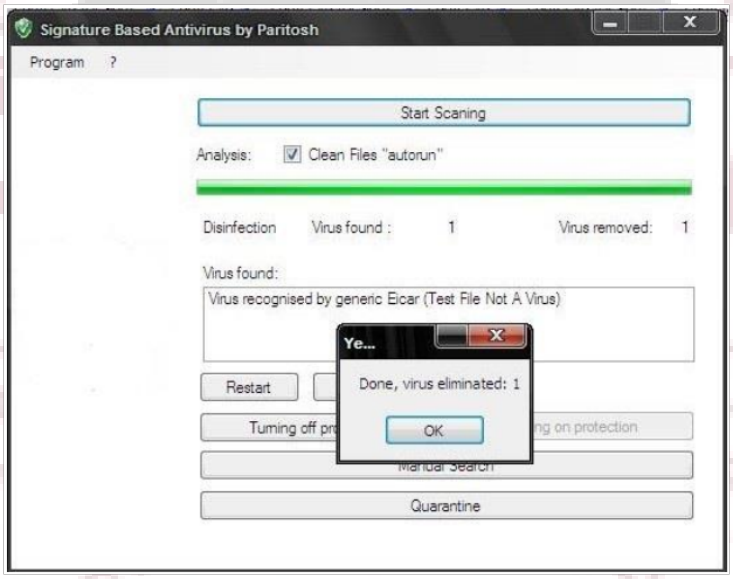

Figure 3: Signature based Detection 3

We know that virus code written in txt, doc or pdf file will not work as malware until we save it as .exe(executable), .bat(batch) or .com(command) file. The keygen of MS Office 2007 Enterprise is not a malware, it is a benign file. Every signature based and heuristic based antivirus software has high malware detection rate as well as false detection rate. We here, are taking 60 benign file which include keygens of software products like MS Office 2007, Windows Vista etc, some system software and malware code in txt files. The Figure 5 presents the false detection rate of ten popular anti-malware products:

The graph above presents Avast home, BitDefender, K-7 Total Security, KingSoft have high false detection rate whereas Avira Personal and Panda Antivirus have low false detection rate. As the result based on testing of 60 malicious files and 60 benign files, our proposed malware detection method, we find the malware detection rate around 95.24%, the false detection rate we find around 3.33% and finally the overall degree of accuracy we find 95.93%. The more accurate malware detection rate, false detection rate and the overall degree of accuracy depends on maximum number of testing files.

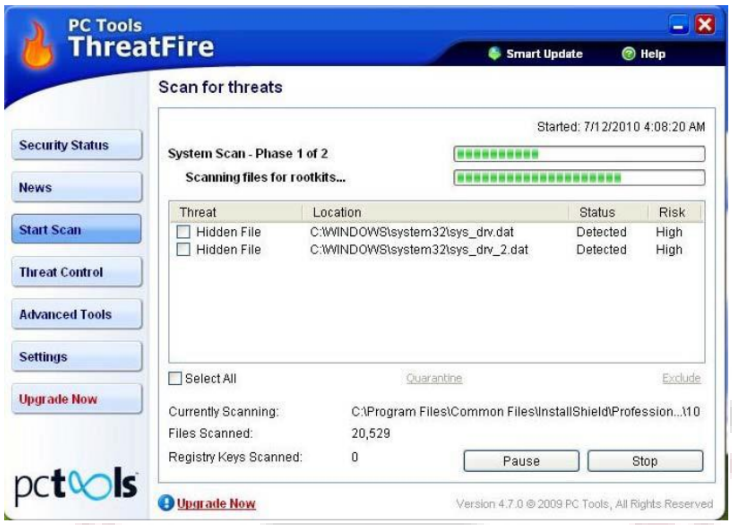

Figure 4: Behavior based Detection

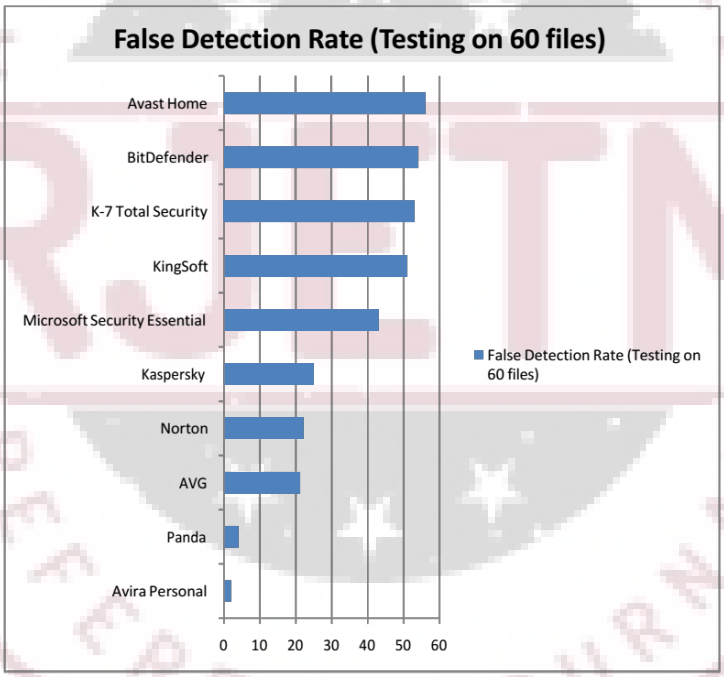

Figure 5: False Detection Rate

7 Conclusion

The paper proposes and designs a combination of signature based and behavior based malware detection system excluding heuristic based detection. Signature based malware detection is the best detection against known malwares, but it is fail to detect new unknown malware. Heuristic based malware detection with signature based malware detection is something better for new unknown malwares but heuristic based detection causes false detection as well because heuristic detection is not true detection it is based on previous result and artificial intelligence. Behavior based malware detection is more better for new unknown malwares, but it allows malwares to run or execute in operating system, every time when same unknown malware attacks it detects behavior of malwares so it is little time consuming. It is better to detect malwares before it runs in operating system.

References

[1] A. Kleymenov and A. Thabet, *Mastering Malware Analysis: A malware analyst's practical guide to combating malicious software, APT, cybercrime, and IoT attacks*, 2nd ed. Packt Publishing, 2022.

[2] D. Barker, *Malware Analysis Techniques: Tricks for the triage of adversarial software.* Packt Publishing, 2021.

[3] A. Mohanta and A. Saldanha, *Malware Analysis And Detection Engineering: A Comprehensive Approach To Detect And Analyze Modern Malware*, 1st ed. Apress, 2020.

[4] A. Bettany and M. Halsey, *Windows Virus and Malware Troubleshooting*, 1st ed. Apress, 2017.

[5] G. M. and S. C. Sethuraman, "A comprehensive survey on deep learning based malware detection techniques," *Computer Science Review*, vol. 47, p. 100529, 2023. [Online]. Available: https://doi.org/10.1016/j.cosrev.2022.100529

[6] C. C. Elisan, *Advanced malware analysis*, 1st ed., ser. Networking & Comm - OMG. McGraw-Hill Education, 2015.

[7] J. Chen, M. H. Alalfi, T. R. Dean, and Y. Zou, "Detecting android malware using clone detection," *Journal of Computer Science and Technology*, vol. 30, no. 5, pp. 942–956, Sep 2015. [Online]. Available: https://doi.org/10.1007/s11390-015-1573-7

[8] M. Kumar, "Scalable malware detection system using big data and distributed machine learning approach," *Soft Computing*, vol. 26, no. 8, pp. 3987–4003, Apr 2022. [Online]. Available: https://doi.org/10.1007/s00500-021-06492-9

[9] W. Peng, F. Li, X. Zou, and J. Wu, "Behavioral malware detection in delay tolerant networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 53–63, 2014. [Online]. Available: https://doi.org/10.1109/TPDS.2013.27

[10] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, and S. Venkatraman, "Robust intelligent malware detection using deep learning," *IEEE Access*, vol. 7, pp. 46 717–46 738, 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2906934

[11] K. Xu, Y. Li, and R. H. Deng, "Iccdetector: Icc-based malware detection on android," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1252–1264, 2016. [Online]. Available: https://doi.org/10.1109/TIFS.2016.2523912

[12] A. A. Al-Hashmi, F. A. Ghaleb, A. Al-Marghilani, A. E. Yahya, S. A. Ebad, M. S. M.S., and A. A. Darem, "Deep-ensemble and multifaceted behavioral malware variant detection model," *IEEE Access*, vol. 10, pp. 42 762–42 777, 2022. [Online]. Available: https://doi.org/10.1109/ACCESS.2022.3168794