ISROSET-IJSRCSE-07531.doc

by

Submission date: 22-Apr-2022 06:47AM (UTC-0600)
Submission ID: 1817239171
File name: ISROSET-IJSRCSE-07531.doc (353K)
Word count: 2476
Character count: 13277

International Journal of Scientific Research in Computer Science and Engineering
Vol.8, Issue.1, pp.01-05, February (2020)
Research Paper
E-ISSN: 2320-7639

Efficient and Simple Machine Learning-based Malware Identification Tool

J.Dhiviya Rose[1*], Isha Mittal[2], Ramya Mihir [3]

[1*] Assistant Professor, School of Computer Science, University of Petroleum and Energy Studies (UPES), Bidholi, Dehradun. INDIA 248007. ORCID ID: 0000-0003-2933-4226
[2,3] Student, School of Computer Science, University of Petroleum and Energy Studies (UPES), Bidholi, Dehradun. INDIA 248007.

e-mail: dhiviyanelson@gmail.com
*Corresponding Author: dhiviyanelson@gmail.com

Available online at: www.isroset.org

Received: .../Sept/2021, Accepted: 06/Dec/2021, Online: 31/Dec/2021

Abstract— When COVID-19 hit the world, it altered the working pattern of all the people around the world. Along with this, there has been an exponential growth in the number of malware and cyber-crime rates. Modern malware uses sophisticated techniqu such as polymorphism and metamorphism to thwart malware detection and analysis. Detecting malware based on its features and behavior is critical for the computer security community. Most anti-virus de nds on signature-based detection which is relatively easy to evade and is ineffective for zero-day exploit-based malware. With the rise of the Internet, there has been huge growth in the number of malware in the world. With this project, we provide a new approach to identify malware using static analysis, i.e. without executing. With the help of different machine learning models, we will identify malware if present in any file, to prevent any further attacks. The target audience and the people who will majorly get benefitted from this project are the students as well as the working professionals who are these days working in online mode due to the pandemic. This application will promote an easy use to identify the files that they receive over emails, SMS, or any other e-mode, to scan before opening any malware file and getting trapped. The target audience for this proposed system is mainly all the students, and professionals, who are more likely to be active on the internet.

Keywords— Malware, Internet Security, Machine Learning

I. INTRODUCTION

Malware includes viruses, ransomware, rootkits, trojans, and a malware attack that can even adversely affect a business and its operations. Appropriate security measures must be put in place by businesses to malware analysis tools as an incident response plan that will provide a proper procedure to ensure there are the recovery time and reduced costs. The analysis of malware nd its usage applies to a major role in the detection of the incident along with identifying the hosts and systems that have been affected in the area of security measures. With the help of the report generated from malware analysis, an organization can mitigate any vulnerabilities and prevent any additional compromises[1]. Thus, through this platform, we provide the organizations and companies an option of mitigating any future malware attacks planned by the attackers to stay protected[2]. This project is done on a google collaborator, where the file received by the corporates can be checked for any vulnerability and thus, prevent it by either not opening it, or by bypassing that vulnerability. Hence, the company can know whether it is completely legitimate or partially legitimate or is not at all safe to access that file, thereby aiding security, and integrity to the user.

The project scope is to create an easy platform for malware detection in a real-time environment where the students, working professionals, and even the new internet generation people can scan and check for malware in any file, to prevent themselves from being trapped in the cyber world. This project is also to create cyber awareness among the people so that the crimes that are increasing these days due to the internet, are reduced to a smaller extent[2]. Our main objective is to identify and classify malware using static analysis i.e. without executing with the help of Machine learning models[3]. So, we are providing software that will run and ask for the file to be checked for malware and notify the user either to open it or not. This application will promote an easy use to identify the files that they receive over emails, SMS, or any other e-mode, to scan before opening any malware file and being

trapped. Therefore, it is for mainly all the students, and professionals, who are more likely to be active on the internet.

The proposed system follows quite a simple methodology and the reference algorithm explaining for computing the malware using machine learning algorithms. The flowchart shows how the file will be checked for it. First, its packets will be checked from the network traffic, then using the ML Models and algorithms, it which if it is malicious, if found malicious, it will drop that packet, else if it is not known, then will check its behavior and decide, else it will be normal to open.

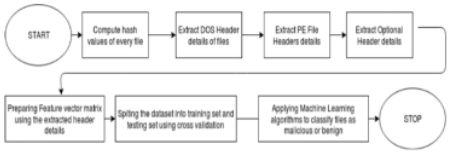

Fig. 1: Proposed system process flow

II. Related Work

The use of conventional mobile malware like botnets, ransomware, trojans has been reduced as most of the antiviruses are updated and remodeled to deals with this. In recent times mobile malware acts have become a common way of attack due to the increase in the use of banking threats and cryptocurrency threats[4]. The malware attacks can be classified as static or dynamic attacks and signature or anomaly-based attacks[5]. There has been a trace of research that has been added in proposing various algorithms for this malware identification. Initial the trace of the information is recorded and with the history, it concluded[6]. Later in the progression in the field of artificial intelligence and machine learning these trails are used in studying the pattern of usages by these attacks using the various classification moadels[7][8][9]. During this pandemic season, the number of cybercrimes has shown a tremendous increase and there has been exponential growth in the number of malware. The task became critical as most modern malware uses sophisticated techniques such as polymorphism and metamorphism to thwart malware detection and analysis[10]. The signature-based detection in anti-virus seems to be less efficient and is relatively easy to evade and is ineffective for zero-day exploit-based malware. Also, it has been noticed that the Internet has become a vital part of our lives with increased usage of services like online banking, online reservation, etc., and our dependence on the Internet is expected to grow[11][12]. With this project, we provide a new approach to identify malware using static analysis, i.e. without executing.

III. Methodology

The process flow of the system is shown in figure 1. The application will allow the user to select the file from the database of the user which later will be checked and scanned the file through different algorithms applied in it. If the file is safe to use, it will show that the file can be opened, else will warn the user not to open the file as it contains a malicious apk file with an active internet connection of the user. This project will help in reducing phishing attacks, where the criminal sends malicious files through emails and the people get trapped. This project will also help in avoiding cross-site scripting attacks, where the criminal can send a malicious JavaScript code within the file. It performs malware detection, where models will be trained on a large corpus of executables using a good set of discriminative predictors extracted through static analysis.

This project will use libraries like Python PE file, hashlib, pandas, DOS Header in the program. The Computation of hash values of every file to check for duplication based on the corpus of the file.If any duplication the files are **removed**. Extraction of header details of the binaries with the help of PE File module functions of python for the analysis purpose. The header details extracted include DOS header, PE File header and Optional header is generated. Followed with the preparation of feature vector matrix by selecting the best features for the training and testing purpose of the dataset is performed and the Cross-Validation split the dataset into training and testing sets. With that, the ML algorithms K-Nearest Neighbors,Decision Trees,Random Forest, Logistic Regression, and SVM (Support Vector Machines) are applied to classify files as malware. Figure 2 shows the processing flow of files in the proposed system.

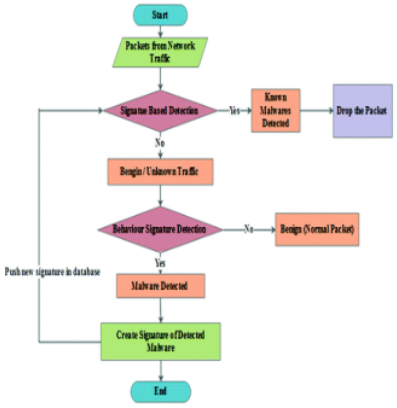

Fig. 2: File Processing Model

The file processing model describes the way of uploading the file or the application to be checked for malware. It is

monitored with the content in every file and if found suspicious, the detector will come into place and using the static code analysis method, will detect the malicious activity. Then will decide whether to give notification or not and help in either launching it and telling the user, else will try to remove it if possible, and already present in the library. If not found suspicious, the file is opened, and the process ends.

A simple and easy-to-use system is implemented where the user can detect malware using the APK dataset. Machine learning is used to build the model using the selected features as input. Comparative analysis is done and a classification report is generated. The data structures used in the implementation includes

- **Static Analysis**: Decompiling, Parsing, Features Generations
- **Dynamic Analysis**: Emulation, Log Extraction, Feature Generation
- **Feature Selection**
- **ML and Malware Detection**: Model Building, Testing, Classification Report

After the hash value for every file is computed the header details for the files are extracted. The is checked by applying machine learning algorithms that classify unseen setup files as legitimate or malicious. The basic assumptions includes that the user should know how to select a file before opening it. The user should understand the output of whether the file is malicious or not; if malicious, what is that file name. The project is dependent on the internet connectivity of the user. The project requires internet connection for multiple users. The software interface used in the implementation includes the programming language as python with the operating System as Windows/Linux/Macintosh and the APK dataset. The Google Collaborator is used as the database interface.The system design is defined by the data flow in figure 3 and the sequence of activities in figure 4.

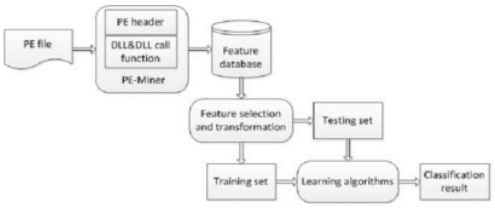

Fig. 3: Data Flow Diagram

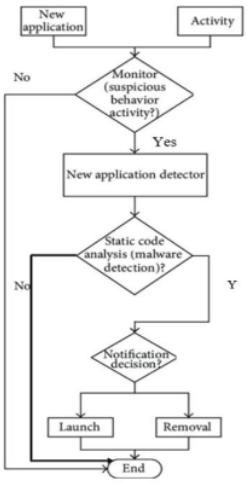

Fig. 4: Activity Diagram

The implementation Constraints used includes

- The model will only work on Windows Portable Executables (PEs). The number of available adversarial malware is limited so it will restrict the robustness of the network.
- The input test files will be correctly labeled, and there can be no misclassified PEs. Each input file will have a Windows API call. Processor and memory requirements are met beforehand.

IV. RESULTS AND DISCUSSION

The system was set to the performance requirements where the system should be based on the web and has to run from a web server. The system also should take initial load time depending on the internet connection strength of the user which also depends on the media size uploaded on the application. The system has proved its efficiency with the security requirements as The system must check for authorization and check for malware files followed with a warning. The software quality attributes used in testing include availability, correctness, usability, and maintainability. The system was checked with various test cases and snapshot are given in Figure 5.

- ✓ TEST CASE 1- Checking whether the file is uploaded or not
- ✓ TEST CASE 2- Checking whether the file has the data/ content in it or not
- ✓ TEST CASE 3- Checking whether the file is malicious or legitimate

Fig. 5: Implementation Screenshot

The limitation of the proposed system included

- ✓ File must be accessed from google drive,
- ✓ Currently, this program is only able to detect malicious content in the tabular form of data, hence no files other than CSV or XLSX are possible to detect.
- ✓ It requires authentication every time it compiles and runs, although it is an additional security feature but might be a limitation for access at different devices.
- ✓ The program is only executable on google collab due to dependency on google drive.

V. CONCLUSION

In this world of digitalization, where advancements have made people so comfortable with digital services, they don't need to travel out and stand in long queues to get small work done. But with these advancements, people are also prone to several offenses and crimes that might have been planned against them, which occur online, and would lead to asking for ransom or doing some illegal act on dark webs, etc. These offenses occur due to the slight mistake done on the part of the people due to the lack of knowledge of what files to open and whatnot. This platform created by us will help the people to know whether the document that they have received is malicious and should not be opened or are legitimate and are safe to be opened. Thus, this platform will help people to stay cyber safe and thereby reducing the cybercrime rate.

REFERENCES

[1] O. Aslan and R. Samet, "A Comprehensive Review on Malware Detection Approaches," *IEEE Access*, vol. 8, pp. 6249–6271, 2020, doi: 10.1109/ACCESS.2019.2963724.

[2] Y. Suleiman, S. Sezer, G. McWilliams, and I. Muttik, "New Android malware detection approach using Bayesian classification," *Proc. - Int. Conf. Adv. Inf. Netw. Appl. AINA*, pp. 121–128, 2013, doi: 10.1109/AINA.2013.88.

[3] A. Kumar *et al.*, "Malware detection using machine learning," *Commun. Comput. Inf. Sci.*, vol. 1232, pp. 61–71, 2020, doi: 10.1007/978-3-030-65384-2_5.

[4] T. Alsmadi and N. Alqudah, "A Survey on malware detection techniques," *2021 Int. Conf. Inf. Technol. ICIT 2021 - Proc.*, no. 2, pp. 371–376, 2021, doi: 10.1109/ICIT52682.2021.9491765.

[5] A. Amamra, C. Talhi, and J. M. Robert, "Smartphone malware detection: From a survey towards taxonomy," *Proc. 2012 7th Int. Conf. Malicious Unwanted Software, Malware 2012*, pp. 79–86, 2012, doi: 10.1109/MALWARE.2012.6461012.

[6] S. Tenneriello, "Panoramas," *Herman Melv. Context*, pp. 157–166, 2018, doi: 10.1017/9781316755204.017.

[7] T. Alsmadi and N. Alqudah, "A Survey on malware detection techniques," *2021 Int. Conf. Inf. Technol. ICIT 2021 - Proc.*, pp. 371–376, 2021, doi: 10.1109/ICIT52682.2021.9491765.

[8] H. El Merabet and A. Hajraoui, "A survey of malware detection techniques based on machine learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 1, pp. 366–373, 2019, doi: 10.14569/IJACSA.2019.0100148.

[9] Z. Wang, Q. Liu, and Y. Chi, "Review of android malware detection based on deep learning," *IEEE Access*, vol. 8, pp. 181102–181126, 2020, doi: 10.1109/ACCESS.2020.3028370.

[10] H. S. Anderson, B. Filar, and P. Roth, "Evading Machine Learning Malware Detection," *BlackHat DC*, p. 6, 2017, [Online]. Available: https://github.com/EndgameInc/gym-malware%0Ahttps://www.blackhat.com/docs/us-17/thursday/us-17-Anderson-Bot-Vs-Bot-Evading-Machine-Learning-Malware-Detection-wp.pdf.

[11] D. J. Wu, C. H. Mao, T. E. Wei, H. M. Lee, and K. P. Wu, "DroidMat: Android malware detection through manifest and API calls tracing," *Proc. 2012 7th Asia Jt. Conf. Inf. Secur. AsiaJCIS 2012*, pp. 62–69, 2012, doi: 10.1109/AsiaJCIS.2012.18.

[12] H. W. Hsiao, D. N. Chen, and T. Wu, "Detecting hiding malicious website using network traffic mining approach," *ICETC 2010 - 2010 2nd Int. Conf. Educ. Technol. Comput.*, vol. 5, 2010, doi: 10.1109/ICETC.2010.5530064.

ISROSET-IJSRCSE-07531.doc

ORIGINALITY REPORT

17%	16%	1%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	security.cse.iitk.ac.in Internet Source	7%
2	www.jigsawacademy.com Internet Source	4%
3	Submitted to Hoa Sen University Student Paper	3%
4	www.isroset.org Internet Source	1%
5	pdfs.semanticscholar.org Internet Source	1%
6	Submitted to Bahcesehir University Student Paper	1%
7	Submitted to Divine Word Univresity Student Paper	1%
8	linknovate.com Internet Source	<1%
9	Ertugrul Ayyildiz, Melike Erdogan, Alev Taskin. "Forecasting COVID-19 recovered cases with Artificial Neural Networks to enable designing	<1%

an effective blood supply chain", Computers in Biology and Medicine, 2021
Publication

10	Submitted to Koc University Student Paper	<1%
11	www.ijeat.org Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		