International Journal of Scientific Research in Computer Science and Engineering
Research Paper
Vol.10, Issue.2, pp.64-68, April (2022)
E-ISSN: 2320-7639

Efficient and Simple Machine Learning-based Malware and Trojan 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

[*]Corresponding Author: dhiviyarj@ddn.upes.ac.in

Available online at: www.isroset.org

Received: 12/Mar/2022, Accepted: 15/Apr/2022, Online: 30/Apr/2022

Abstract— When COVID-19 hit the world, it altered the working pattern of all the people around the world. Along with this, it is seen that there has been an exponential growth in the cases of malware, trojans and cyber-crime rates. New and recent malwares uses advanced techniques like polymorphism and metamorphism to help in assisting the malware detection and analysis procedure. Identifying malware in view of its features and conduct is analytic and serious for the computer security. Most of the anti-viruses that are present rely upon the signature-based noticing which is moderately easy to dodge and evade and is insufficient and also ineffective for zero-day exploit-based malware. With the ascent of the Internet, there has been enormous development in the quantity of malware on the planet. 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 comprises of viruses, ransomware, trojans, rootkits, and malware assault that could unfavorably influence a business and its tasks and operational functions. Appropriate security and safety measures should be set up by the organizations and businesses for malware examination as an incident response plan that will give a legitimate system with proper procedure to guarantee and ensure that there is a recuperation/ recovery time and diminished and fewer costs. The investigation of malware and its utilization applies for a significant role in the spotting and noticing of the incident alongside recognizing the hosts and frameworks that have been impacted in the area of safety measures. With the assistance of the report that is produced from malware investigation, an association can relieve and reduce any weaknesses and vulnerabilities and block any additional compromises that have to be taken, thereby reducing the count and chances of finding vulnerabilities in the system[1]. Thus, through this platform, we provide the organizations, businesses and companies an option and an opportunity of mitigating any future malware attacks planned by the attackers in order to stay protected[2].

This project is done on a google collaborator platform, where the file received by the corporates can be checked for any vulnerability and viruses and thus, prevent it by either not opening it, or by bypassing that vulnerability, thereby staying protected. Additionally, it has been noticed that the Internet has become an essential part of our lives with increased usage of services like web-based banking, online reservation, and many more such services, and our sustenance on the Internet is expected to grow in the coming future, so with this ascent of the Internet, there has been seen a huge development in the quality and the number of malware across the globe. Hence, through this, the company can know whether it is completely legitimate or partially legitimate or is not at all safe to access that file, thereby aiding the security, and integrity of the user.

The project scope is to create an easy-accessible 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 getting trapped in the cyber world[2]. 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. 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 to the user 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 getting trapped. So, 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.

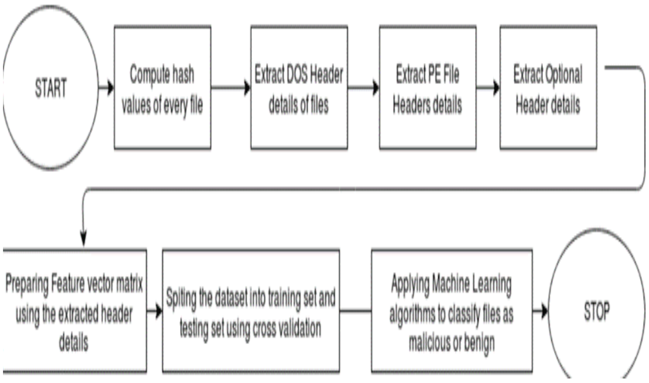

START
Compute hash values of every file
Extract DOS Header details of files
Extract PE File Headers details
Extract Optional Header details
Preparing Feature vector matrix using the extracted header details
Spiting the dataset into training set and testing set using cross validation
Applying Machine Learning algorithms to classify files as malicious or benign
STOP

Fig. 1: Proposed system process flow

II. RELATED WORK

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 and awareness of what files to open and whatnot. New modern malwares are being developed in such a manner that seems very lucrative and attractive and also legitimate to the people, which insists them to open that file, thereby end up landing trapped. The main reason behind getting locked up on a device due to malware is that the people are unaware of the cybercrimes, they lack knowledge and also provide permission without being aware.

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's concluded[6]. Later in the progression done in the field of artificial intelligence and machine learning, these trials are used in studying the pattern of usages by these attacks using the various classification models[7][8][9]. During this pandemic season, the count of cybercrimes has shown a huge increment and there has been outstanding growth in the figure of malware. The undertaking of these counts have become crucial as most current malware utilizes complex strategies, for example, polymorphism and metamorphism to ruin malware detection and examination[10]. The signature-based detection in anti-virus seems to be less efficient and simple to dodge and is inadequate for zero-day exploit-based malware. Additionally, it has been seen that the Internet has turned into an essential piece of our lives with expanded use of services like web-based banking, online reservation, and so forth, and our sustenance on the Internet is supposed to develop[11][12]. Also, with the recent developments, human life is shifted from real to virtual environments, and so the cyber-criminals, due to the COVID-19 pandemic.

The criminals have gained an advantage over this shift and it has become easier for them to commit a crime and launch a cyber-attack with the help of the recently developed new malware. New methods of developing malware are quite different and advanced as compared to the earlier traditional methods of developing as well as combating the malware variants. The new methods use Deep Learning methods instead of Machine Learning models[13]. There have been seen that there are already existing malware detecting tools and techniques like the Intrusion Detecting System (IDS), Firewalls, and Virus Scans. But since, with the advancements made in the technologies, the cybercrime and count of malware is also increasing with new variants seen in it, these tools and techniques have to either get updated to reduce the effect of that malware, or new tools and techniques must be created to thwart them[14]. In recent times, it is also seen that android malware is also getting increased, and this increase is mainly seen in the lockdown. Android malware is an on-trend to the prevalent Android Operating System, as most of the users have been using smartphones and it is easier to install malware on Android devices, without the permission of the user. Static Analysis is one of the ways that is applied for Android Malware detection and it helps to easily and quickly detect the malware before its installation takes place[15].

Additionally, as we know that malicious programs genuinely cause a lot of problems to the person, which is considered to be done by the malware inputs into the devices, the malware discovery framework can be utilized using several information mining and AI techniques. This prediction of obscuring a malware can be done through

ANN (Artificial Neural Network) algorithm and through behaviors analysis decision tree[16]. Along with this, it is noticed that social networking has become a topmost application used by most of the users to share and communicate information. The cyber criminals find this way as the easiest way of inputting the malware, and trapping the individuals by taking access to their personal information, which is the most secure one. Hence it is important to identify the open issues and provide more secure solution that would help to solve the problem and also get known with the various OSN threats like misusing the identity, phishing attacks etc.[17].With this project, we provide a new approach to identifying malware using static analysis, i.e. without executing, on the computer system as well as on the android devices.

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 functions of detecting the malware, where models will be prepared on an enormous corpus of executables utilizing a decent arrangement of discriminative indicators that are extracted through static analysis procedure.

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.

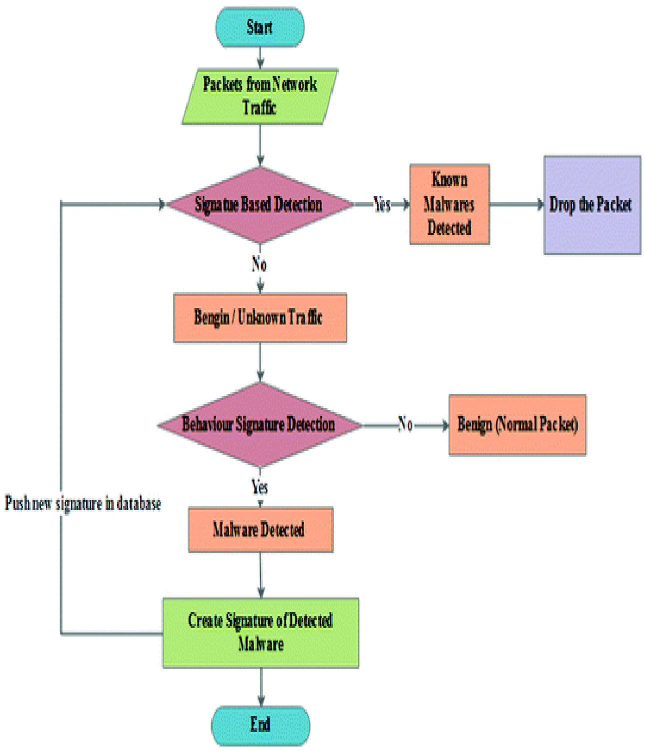

Start
Packets from Network Traffic
Signature Based Detection
Yes
Known Malwares Detected
Drop the Packet
No
Bengin / Unknown Traffic
Behaviour Signature Detection
No
Benign (Normal Packet)
Yes
Push new signature in database
Malware Detected
Create Signature of Detected Malware
End

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 platform 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.

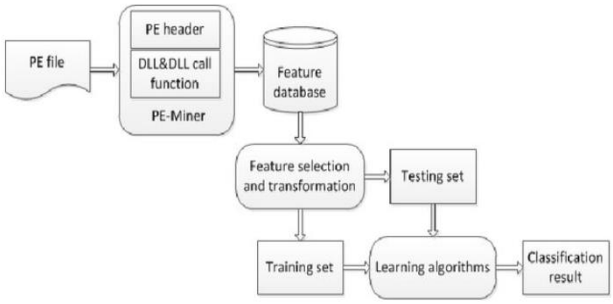


Fig. 3: Data Flow Diagram

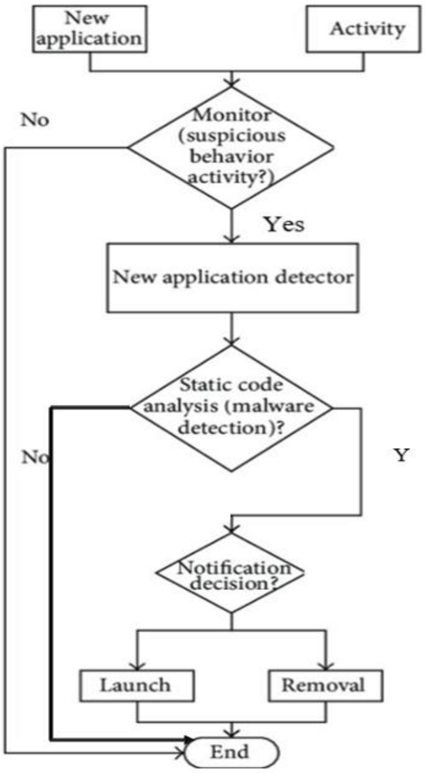


Fig. 4: Activity Diagram

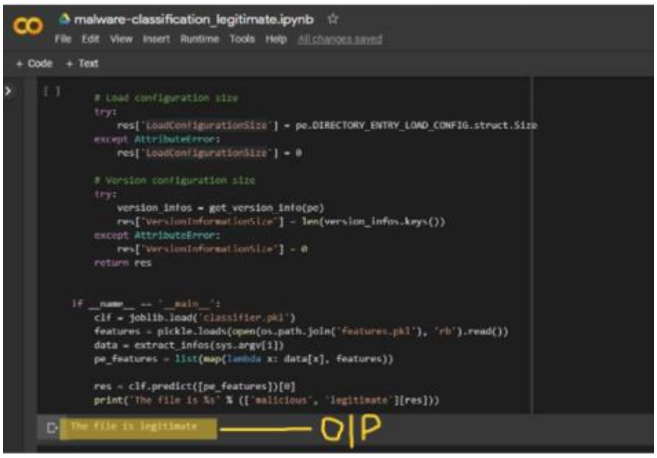
The implementation Constraints used includes

- 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 machine should be built on the web and needs 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



```
    # Load configuration size
    try:
        res['LoadConfigurationSize'] = pe.DIRECTORY_ENTRY_LOAD_CONFIG.struct.Size
    except AttributeError:
        res['LoadConfigurationSize'] = 0

    # Version configuration size
    try:
        version_infos = get_version_info(pe)
        res['VersionInformationSize'] = len(version_infos.keys())
    except AttributeError:
        res['VersionInformationSize'] = 0
    return res


if __name__ == '__main__':
    clf = joblib.load('classifier.pkl')
    features = pickle.loads(open(os.path.join('features.pkl'), 'rb').read())
    data = extract_infos(sys.argv[1])
    pe_features = list(map(lambda x: data[x], features))

    res = clf.predict([pe_features])[0]
    print('The file is %s' % (['malicious', 'legitimate'][res]))
```

```
The file is 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

Getting upgraded every single moment on the internet, also requires us to get upgraded and stay safe on internet as we are in our real life, therefore, in order to stay up-to-date, we need to stay aware of things going around us on internet, and look for ways that could keep us safe. With this upgradation, the cybercrimes are also getting upgraded, which are getting hard to be identified by the normal people, as they are being created using the mixture of old already existing malware, resulting in new variants arrival.

Therefore, this platform created by us uses new methods and new ways that rely on static analysis rather than execution that will help the people to get aware and also 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.

[13] Omer Aslan, Abdullah Asim Yilmaz, "A New Malware Classification Framework Based on Deep Learning Algorithms," IEEE Access, **vol. 9, pp. 87936-87951, 2021,** doi: 10.1109/ACCESS.2021.3089586.

[14] Sudhir Kumar Pandey, B.M. Mehtre, "Performance of malware detection tools: A comparison," 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, 2014, pp. 1811-1817, doi: 10.1109/ICACCCT.2014.7019422.

[15] Y. Pan, X. Ge, C. Fang and Y. Fan, "A Systematic Literature Review of Android Malware Detection Using Static Analysis," in IEEE Access, **vol. 8, pp. 116363-116379, 2020,** doi: 10.1109/ACCESS.2020.3002842.

[16] Sweta Khatana, Anurag Jain, "Malware Detection Using the Behavioral Analysis of the Web based Applications and User," International Journal of Computer Sciences and Engineering, **Vol.7, Issue.5, pp.1026-1031, 2019.**

[17] Jamuna Rani S., Vagdevi S., "Online Intrusion and Security Measures in Social Networking Environment – A Survey", International Journal of Computer Sciences and Engineering, **Vol.8, Issue.12, pp.39-45, 2020.**