COVID-19 Android Malware Detection Using Permissions and App Icons

By: David Tan

Faculty Mentor: Alfredo Perez

Department of Computer Science, Columbus State University

Abstract

Since the start of the COVID-19 pandemic in late 2019, there has been a dramatic increase in the percentage of malicious applications targeting mobile devices worldwide which indicates that malicious developers are capitalizing on the COVID-19 pandemic. Disguised as benign applications, these malware applications are taking advantage of the asymmetry between irrelevant/redundant permissions and informative permissions of benign Android applications in order to access the sensitive and private information on these devices. In this paper, we propose an Android permissions-based COVID-19 malware detection system that determines if an application is malicious based on the usage of suspicious permissions. The system used a multi-level based methodology. First, we extracted and identified the Android permissions and icons from the public dataset of 2,017 applications of the research paper "Beyond the virus: A First Look at Coronavirus-themed Android Malware" by employing the python tool Androguard. We then utilize a machine learning model to categorize the applications as either malware or non-malware based on the usage of suspicious permissions (Wang et al., 2021). Through extensive experiments, the proposed model has the potential to demonstrate a high level of symmetry between irrelevant permissions and malware applications. In conclusion, the proposed system has the potential to provide a low-cost alternative for Android malware detection for malicious applications including repacked applications.

Keywords

Machine learning, cybersecurity, COVID-19 pandemic, android malware, malware detection

Introduction

As the use of systems, applications, and software is on the rise globally, user privacy and data protection are of the utmost concern to individuals, firms, and businesses operating online. Furthermore, there has been a significant increase in the number of Android malware applications since the start of the COVID-19 pandemic. Android's status as one of most commonly used smartphone operating systems and its daily increase in applications in the smartphone application market has made the scope of this issue of increasingly emerging Android malware applications a pressing issue in the industry (Kim, Kang, Cho, & Choi, 2022).

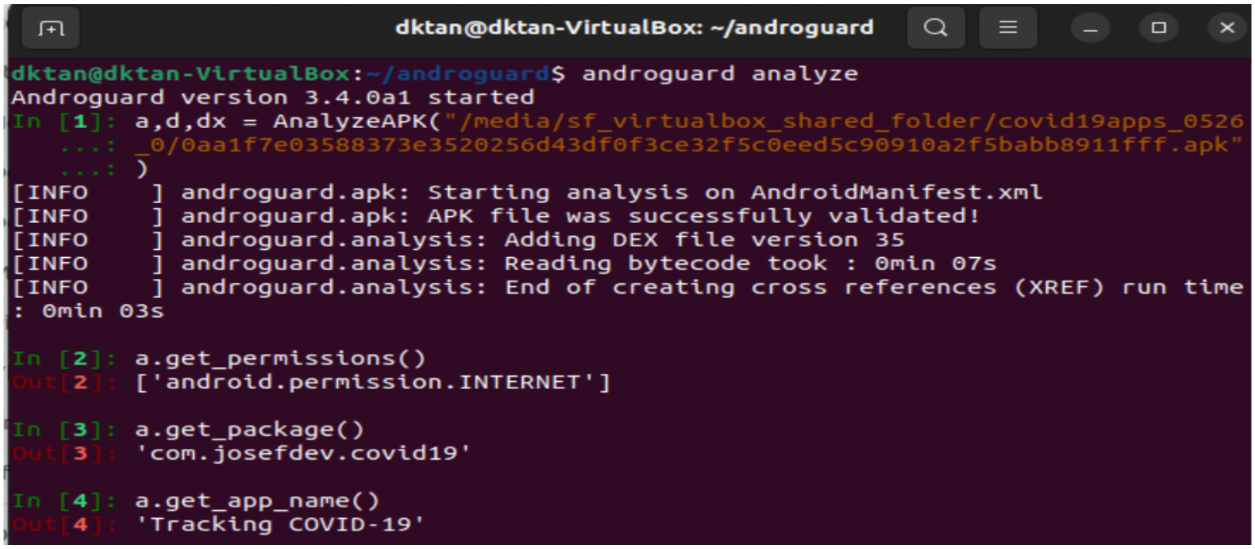
Furthermore, the most popular features offered by Android platforms, which are the supply of feature rich applications for a wide range of users and the possibility to facilitate users and developers with an open-source policy for application availability and tolerance to application verification at the time of release, have driven the increase in interest of many cybercriminals worldwide (Mahesh, & Hemalatha, 2022). These malicious Android applications, disguised as benign applications, are designed to either steal users' private information or make profit by phishing and extortion in the form of malware like worms, exploits, Trojans, and viruses. Some of these applications are released by malicious developers in many variants in order to avoid detection and to target a larger audience (Shen et al., 2022). To address the increasing concerns of the intrusion of malicious applications into the smartphone application market and the lack of understanding of the research community of the scope of the coronavirus themed mobile malware, many researchers have developed and used various approaches to create effective and efficient Android malware detection tools using different methodologies like the one presented in this paper (Junyang et al., 2020; Almomani, Alkhayer, & El-Shafai, 2022; Hosseini, Nezhad, & Seilani, 2021; Kim et al., 2022).

The mechanisms of our malware detection using APKs consisted of static analysis, which is performed in a non-runtime environment concerning application binaries and analyzing the meta and auxiliary information (Almomani et al., 2022). Static analysis was selected over dynamic analysis, which is another technique commonly used in malware detection involving APKs, because static analysis is considered to be faster, less costly, and more helpful in developing an initial view of the APKs (Mahesh, & Hemalatha, 2022). The ultimate aim of using static analysis is to identify and segregate malicious or repackaged applications before their installation and executions by flagging the application as malicious based on their permissions (Mahesh, & Hemalatha, 2022). Currently, Android uses the permission-based model to protect user's private information by restricting applications from accessing this information, because Android application permissions are the most fundamental evaluation mechanisms on the Android platform (Singh et al., 2021). However, the limitation in the current permission-based model used by Android makes permission scanning a necessary step for malware detection without being granted explicit permission (Singh et al., 2021). Previous studies have also used machine learning and deep learning models to identify evasion-aware malicious applications (Almomani et al., 2022; Hosseini et al., 2021; Kim et al., 2022; Mahesh & Hemalatha, 2022; Akbar et al., 2022). In this paper, we focus primarily on permission-based malware detection by describing our proposed model for Android permission-based malware detection.

Methodology

The purpose of the proposed scheme of our permission-based model is to produce malware detection of APKs for non-malware and malware classification for a faster classification. To address the increasing concerns of the intrusion of malicious applications into the smartphone application market and the lack of understanding of the research community of the scope of the coronavirus themed mobile malware, we have developed and created an effective and efficient approach to Android malware detection (Hosseini et al., 2021). The mechanisms of our malware detection, which determines if an application is malicious based on the usage of suspicious permissions, using APKs, consisted of static analysis in order to produce a faster and less costly initial view of the APKs. The system used a multi-level based methodology.

First, we decompile the APKs files of Android applications using AndroLyze, a AndroGuard tool, for the permission extraction process (see Fig. 1).



```
dktan@dktan-VirtualBox: ~/androguard
dktan@dktan-VirtualBox:~/androguard$ androguard analyze
Androguard version 3.4.0a1 started
In [1]: a,d,dx = AnalyzeAPK("/media/sf_virtualbox_shared_folder/covid19apps_0526
   ...: _0/0aa1f7e03588373e3520256d43df0f3ce32f5c0eed5c90910a2f5babb8911fff.apk"
   ...: )
[INFO    ] androguard.apk: Starting analysis on AndroidManifest.xml
[INFO    ] androguard.apk: APK file was successfully validated!
[INFO    ] androguard.analysis: Adding DEX file version 35
[INFO    ] androguard.analysis: Reading bytecode took : 0min 07s
[INFO    ] androguard.analysis: End of creating cross references (XREF) run time
: 0min 03s

In [2]: a.get_permissions()
Out[2]: ['android.permission.INTERNET']

In [3]: a.get_package()
Out[3]: 'com.josefdev.covid19'

In [4]: a.get_app_name()
Out[4]: 'Tracking COVID-19'
```

Fig. 1. depicts the codes used to run the AndroGuard tool AndroLyze in order to extract the permissions, app name, and package name of the Android applications.

Furthermore, the APKs files of Android applications were from the dataset of 2,017 applications of the study “Beyond the virus: A First Look at Coronavirus-themed Android Malware” (Wang et al., 2021). The public dataset contained significant features of each application such as AV rank, apk size, and application release date. After generating the permission feature set, then the machine learning model for detecting malware was employed. In other words, the proposed scheme involved filtering, finalizing, and extracting the permissions dataset and developing the proposed machine learning algorithm to classify the Android malicious applications (Almomani, Ahmed, & El-Shafai, 2022).. The proposed scheme for the model is depicted in Fig. 2.

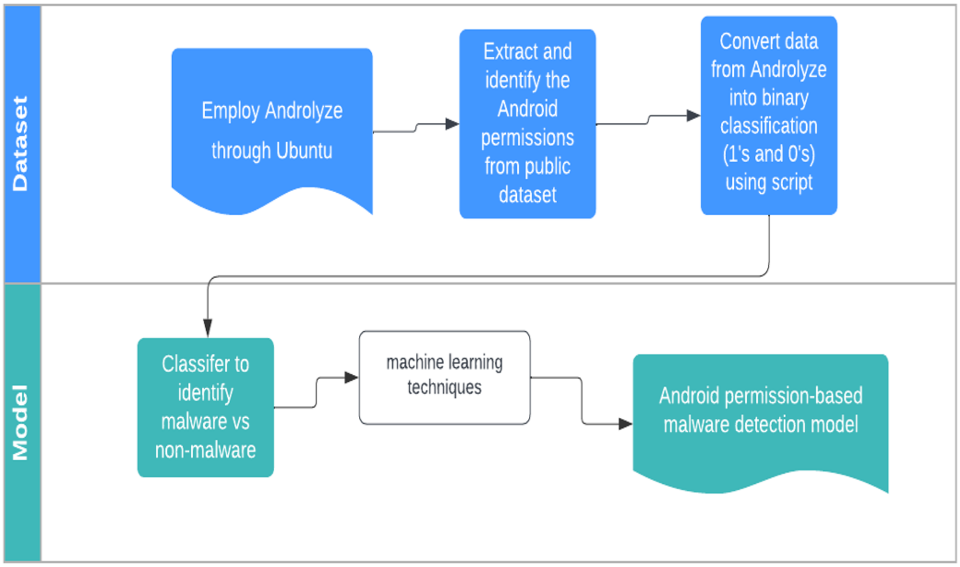

Dataset
Employ Androlyze through Ubuntu
Extract and identify the Android permissions from public dataset
Convert data from Androlyze into binary classification (1's and 0's) using script
Model
Classifer to identify malware vs non-malware
machine learning techniques
Android permission-based malware detection model

Fig. 2. depicts the proposed scheme for the machine learning model proposed in this study.

A script was created and implemented in order to convert the information obtained from Androguard concerning apk files and their permission(s) into a binary classification of 0's and 1's where 1's confirmed if the apk file possesses that specific permission(s) and where 0's confirmed if the apk file did not possess that specific permission(s). Our proposed permission-based model employed a supervised machine learning classifier using a Random Forest algorithm to correctly predict malicious applications with minimum false positives (Almomani et al., 2022; Albahar et al., 2022). Furthermore, training and testing strategy techniques would be used to create uniformity in the model (Almomani et al., 2022).

Results

Due to the time constraint associated with our study, our only concrete results involve Fig. 3 and Fig. 4. Fig. 3 shows our results from filtering, finalizing, and extracting the permissions dataset, and it serves to demonstrate that our results were similar to the results of the study from the paper “Beyond the virus: A First Look at Coronavirus-themed Android Malware,” which is the origin of the dataset. Fig. 4 confirms that our proposed machine learning model using Random Forest classifier is feasible and can be used in future research.

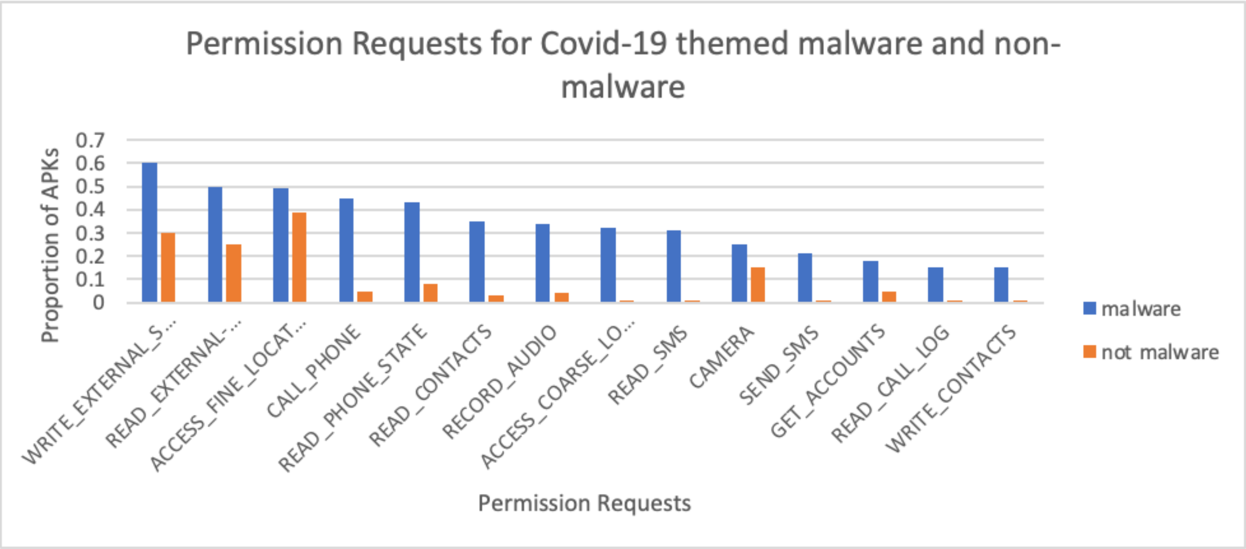


Fig. 3. depicts our results of permission requests for COVID-19 themed malware and non-malware that are inline with the results of the paper “Beyond the virus: A First Look at Coronavirus-themed Android Malware.”

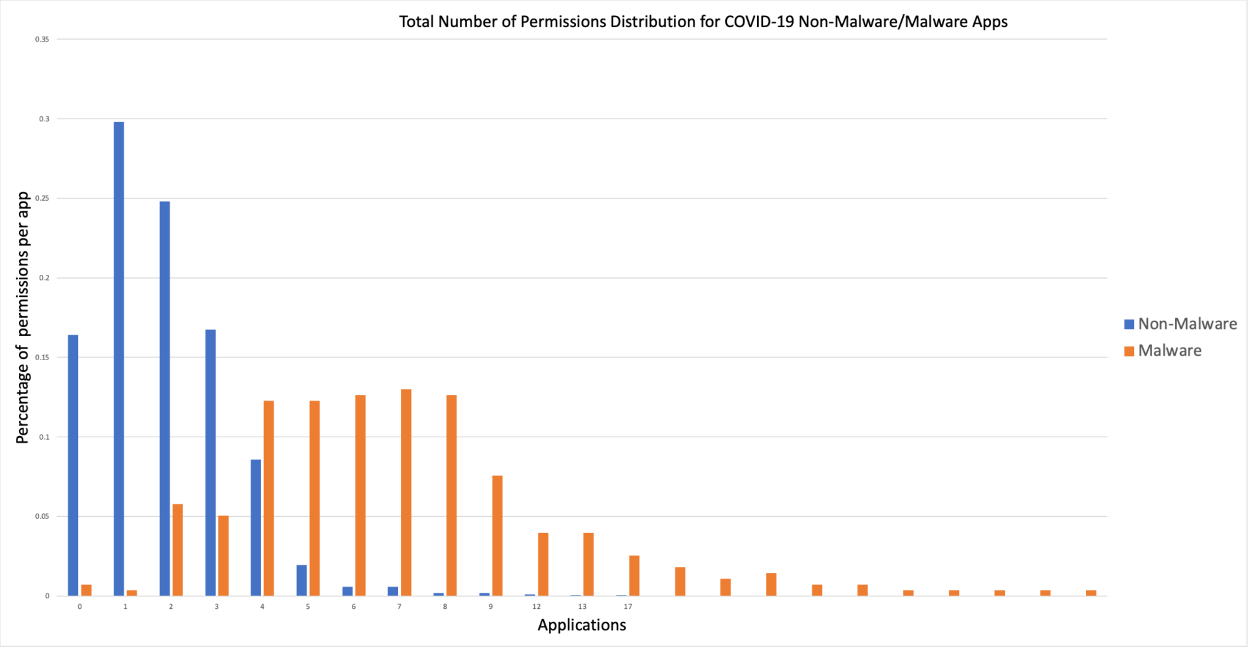


Fig. 4. shows our results of the total number of permissions distribution for the COVID-19 non-malware/malware applications that confirms that our proposed machine learning model is feasible.

Conclusion

Malware, which is any software designed to cause disruption in order to gain unauthorized access to the information of a user, has been used by cybercriminals and malicious developers to conduct identity theft and exploitation (Sahin, Akleylek, & Kilic, 2022). The leaked data obtained by these malware have been used to design phishing attacks and gain access to larger amounts of personal data. Events such as the recent announcement of Chinese state-sponsored cyber vulnerability exploitations and Russian state-sponsored cyber criminal activity have brought to light the widespread need for improved responses to security breaches Xu, Li, Deng, & Xu, 2022). Unfortunately, these cyber crimes can affect the integrity and reputation of many victim corporations along with causing catastrophic financial losses for both the organizations and its consumers. Therefore, it has been of great interest to researchers in the field of cybersecurity and individuals, firms, and businesses operating online to combat this cyber threat. In this paper, we used static analysis to identify and segregate malicious or repackaged applications before their installation and executions by flagging the application as malicious based on their permissions. We have developed and created an effective and efficient approach to Android malware detection which can be employed in future research.

Future Work

Based on limitations in our research such as a lack of time, we would recommend the implementation of the proposed permission-based model of malware detection and the development of an icon-based model of malware detection for future work. The proposed model can be used as a low cost alternative for Android malware detection for malicious applications including repacked applications.

Acknowledgements

I would like to thank Dr. Perez for guiding and helping me through this summer research. I would also like to thank Columbus State University and its faculty members and students along with my fellow REU participants. I would like to thank the National Science Foundation (Computer and Information Science and Engineering) for awarding Columbus State University with this REU.

References

Akbar, F., Hussain, M., Mumtaz, R., Riaz, Q., Wahab, A. W. A., & Jung, K.-H. (2022). Permissions-Based Detection of Android Malware Using Machine Learning. Symmetry (20738994), 14(4), N.PAG.

Albahar, M. A., ElSayed, M. S., & Jurcut, A. (2022). A Modified ResNeXt for Android Malware

Identification and Classification. Computational Intelligence & Neuroscience, 1–20.

Almomani, I., Ahmed, M., & El-Shafai, W. (2022). Android malware analysis in a nutshell. PLoS ONE, 17(7), 1–28.

Almomani, I., Alkhayer, A., & El-Shafai, W. (2022). An Automated Vision-Based Deep Learning Model for Efficient Detection of Android Malware Attacks. IEEE Access, Access, IEEE, 10, 2700–2720

Hosseini, S., Nezhad, A. E., & Seilani, H. (2021). Android malware classification using convolutional neural network and LSTM. Journal of Computer Virology and Hacking Techniques, 17(4), 307–318.

Junyang Qiu, Jun Zhang, Wei Luo, Lei Pan, Nepal, S., & Yang Xiang. (2020). A Survey of Android Malware Detection with Deep Neural Models. ACM Computing Surveys, 53(6), 1–36.

Kim, H., Kang, M., Cho, S., & Choi, S. (2022). Efficient Deep Learning Network With Multi-Streams for Android Malware Family Classification. IEEE Access, Access, IEEE, 10, 5518–5532

Mahesh, P. C. S., & Hemalatha, S. (2022). An Efficient Android Malware Detection Using Adaptive Red Fox Optimization Based CNN. Wireless Personal Communications: An International Journal, 1–22.

Sahin, D. O., Akleylek, S., & Kilic, E. (2022). LinRegDroid: Detection of Android Malware Using Multiple Linear Regression Models-Based Classifiers. IEEE Access, Access, IEEE, 10, 14246–14259

Singh, J., Thakur, D., Gera, T., Shah, B., Abuhmed, T., & Ali, F. (2021). Classification and Analysis of Android Malware Images Using Feature Fusion Technique. IEEE Access,

Access, IEEE, 9, 90102–90117

Shen, L., Feng, J., Chen, Z., Sun, Z., Liang, D., Li, H., & Wang, Y. (2022). Self-attention based convolutional-LSTM for android malware detection using network traffics grayscale image. Applied Intelligence: The International Journal of Research on Intelligent Systems for Real Life Complex Problems, 1–23.

Wang L, He R, Wang H, Xia P, Li Y, Wu L, Zhou Y, Luo X, Sui Y, Guo Y, Xu G. (2021). Beyond the virus: a first look at coronavirus-themed Android malware. *Empir Softw Eng. 26*(4):82. doi: 10.1007/s10664-021-09974-4.

Xu, J., Li, Y., Deng, R. H., & Xu, K. (2022). SDAC: A Slow-Aging Solution for Android Malware Detection Using Semantic Distance Based API Clustering. IEEE Transactions on Dependable and Secure Computing, Dependable and Secure Computing, IEEE Transactions on, IEEE Trans. Dependable and Secure Comput, 19(2), 1149–1163.