Android App Malign Detection Using Machine Learning

[1]Y.Susheela, [2]Dr.Dinesh Kumar, [3]K.Sri Harsha
[1]Assistant Professor, [2]Associate Professor, [3]Student [1,2,3] Dept. of Computer Science Engineering,
[1,2,3]Vaageswari College of Engineering, Karimnagar, Telangana.
E-Mail: chiduralasusheela@gmail.com, sahni.dinesh@live.com, cmnarsingoju@gmail.com

ABSTRACT

The rapid growth of mobile devices has been increased in today world and made it remarkable for advance technologies, various operating systems have been developed from those, android is most popular and because of numerous applications it provides but cannot provide privacy and data integrity due to malicious Android application downloaded from third party markets. Mobile malware is the highest threat and our goal is to detect the malicious android application using genetic algorithm with the combination of support vector machine to build the training classifier which integratesandroid app permissions ahead itself to the developers and end –users. This will assist the developers in the safe use of APIS when developing applications. APIs are extracted from the packed app file we trained a classifier to identify whether an app is potentially malicious or not. Malware detection in android can be performed in two ways behavior-based detection and signature-based methods. The signature-based detection method is simple and detect already known malware. The behavior-based method uses the techniques from machine learning and data science such as decision trees and deep learning and three types of analysis techniques are identified as static, dynamic and hybrid analysis methods .static analysis used to perform by analyzing the byte code into source code. Dynamic analysis is used for detecting malware by analyzing the application. We compared static, dynamic and hybrid analysing the basis of data set feature extraction techniques, feature selection techniques, detection methods we identify the malicious android applications which assist the application developers.

Keywords: Bag of visual words, static analysis, dynamic analysis and hybrid analysis, malicious, support vector machine(SVM).

1. Introduction

In this technological era, usage of mobile applications rapidly increasing and android is one of the most popular operating system available due to convenience and efficiency in various applications. Android has become the most popular one because of the numerous mobile apps it provides. Unfortunately, smartphones running Android have been increasingly targeted by attackers and infected with malicious apps: according to the mobile threat report released by F-Secure over 95% of malicious apps were distributed on the Android platform. Google play is the official app store for android-based devices where many number are been published on it was in millions. Of these, the applications are classified into different category as regular apps, low-quality apps, high quality-apps which in turn consists of malicious applications which makes the targets for cyber criminals. And ML is the most prominent techniques used for detection of malicious android applications which helps in protection. And the data security is not maintained properly which lead to users steal data during the app installation. Android is built on top of linux Kernel. The Android architecture and its built-in security as well as they are threats and security issues for android applications. In addition, third-party markets, which are the primary providers of Android apps, contain a lot of cracked or modified apps with no indication as to whether they have been inspected for security threats or not.

Fig 1.a Malware Detection

The proliferation of dangerous apps (malware) on the Android platform is accelerated by the absence of security screening for Android apps. Therefore, it is essential to create a method that is effective and efficient for detecting malware on Android systems. We suggest a machine learning algorithm and SVM-based approach for malware detection on the Android platform. The android app permissions will be used in this approach to train the SVM classifier to determine if a specific app is malicious or benign by using them as features in the feature vector.

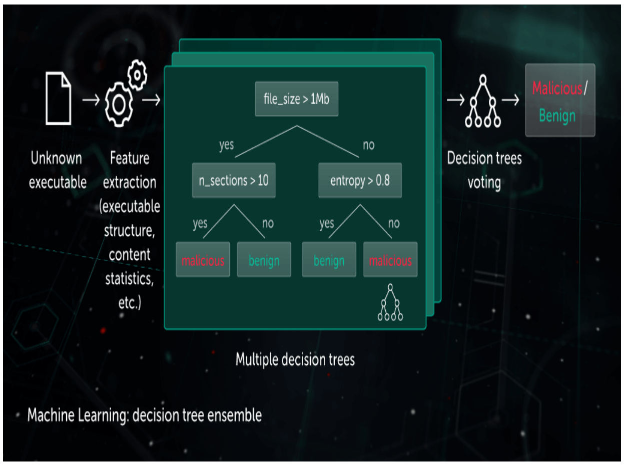


Fig1.b Prediction of Malware/Benign

The data security is not maintained properly which lead to users steal of data during the app installation. Android is built on top of linux kernel. The android architecture and its built-in security as well as they are threats vectors for android. It consists of the hardware layer, native c/c++ libraries and android run time, java application and Programming interface framework, and system applications are stacked top of each one. Each layer is responsible for particular task. In order to successfully differentiate between malware files and clean files while trying to limit the amount of false positives, we present a flexible framework in which one may make use of various machine learning methods. By first working with cascade one-sided perceptron and then with cascade kernelized one-sided perceptrons, we discuss the concepts underlying our system in this research. The concepts underlying this framework were put through a scaling-up procedure that enables us to work with very big datasets of malware and clean files after being successfully tested on medium-sized datasets of malware and clean files. Numerous mobile applications have been inspired by and made possible by the explosive growth of mobile devices and the extraordinary advancements in 4G/5G mobile networking technologies. Different mobile operating systems have been created to support these mobile devices. Because it offers so many mobile applications (apps), Android is one of those operating systems that has gained the most popularity.

The Android platform itself offers a number of security solutions that harden malware installation in order to address the extensive security threats, such as the Android permission system and Google "Bouncer." Each Android app must explicitly request the corresponding permission from the user during the installation process in order to carry out specific tasks on Android devices, such as sending an SMS message. However, a lot of users frequently grant arbitrary permissions to unidentified Android apps without even considering the kinds of permissions they are asking for, significantly weakening the security offered by the Android permission system. Because of this, it is practically impossible for the Android permission system to restrict the spread of malicious apps.

On the other side, Google "Bouncer" is a service that was added to Google Play, the official Android market, in 2012. Using its reputation engine and cloud infrastructure, it attempts to automatically check apps (both new and previously uploaded ones) and developer accounts in Google Play. Bouncer still has numerous restrictions, even though it gives a second layer of security for Android. First off, because Bouncer can only scan Android apps for a short period of time, dangerous apps can easily avoid it by remaining harmless throughout the scanphase. Second, when the initial installation is scanned by Bouncer, no harmful code has to be Thus, it is critical to develop an effective and efficient approach to detect malware for Android platforms. We propose a malware detection scheme for Android platform using Machine learning algorithm and SVM-based approach. In this scheme, we will take the android app permissions, and use them as features in the feature vector and will train the SVM classifier to detect whether a particular app is malware or benign.

2. LiteratureSurvey

Once the developers start identifying or developing the application need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before developing the system the above consideration are taken into account for developing the proposed system. In today's scenario, there is no proper security provided for the applications of android which can steal the users data. According to the recent survey, there will be a rapid increase in the use of mobile application population of over 1.6 billion. There isno securityprovided to protect the users data. Thus, secured data is the key solution to reduce the malwareattacks. The solution for the problems is being raised. The securing users data can be a solution to user's time and efficiency so that the data can be maintained in the secure manner. In this, the data is secured through analyzing and processing APK files, the output is obtained. It allows the user to feel more secure and user friendly for everyone. This is used to provide confidentiality to different developers and useful for developers to identify the malware ahead itself which will be cost efficient and for the users in the times. This makes the people and the society more conventional in the present and for the futurepurpose.

III. ProposedSystem

The sensitivity of mobile platforms and their potential for abuse, several security issues not too dissimilar to those already affecting traditional IT network counterparts are beginning to surface. The most significant issue is the emergence of traditional malware such as viruses, worms, Trojan horses, and rootkits. Malicious software in this context behaves similarly to the same threats on traditional IT networks. Mobile malware is the highest threat to the security of IoT data, users personal information, identity, and corporate/financial information.

We considered static, dynamic, and hybrid detection analysis. In this performance analysis, we compared static, dynamic, and hybrid analyses on the basis of data set, feature extraction techniques, feature selection techniques, detection methods, and the accuracy achieved by these methods. Therefore, we identify suspicious API calls, system calls, and the permissions that are extracted and selected as features to detect mobile malware.

IV.RelatedWork

Previous reviews were discussed various ML-based Android malware detection techniques and ways to improve android security.

Systematic studies survey conducted using existing literature up to 2017 to identify malware techniques. static analysis techniques used for android applications. The tools can be used to perform Android code analysis using static analysis techniques. The most widely used approach to detect privacy and security issues. The review identified four methods as characteristic-based, opcode-based, program graph-based and symbolic execution-based. After that, it evaluated the capabilities of static analysis based android malware detection. under static analysis and dynamic there are several approaches that can be used to identify android malware according to it they are five types of malware detection methods they are static detection, dynamic detection, hybrid detection and permission detection and emulation detection based. They also summarized the reviewed work with the model accuracy of malware detection.

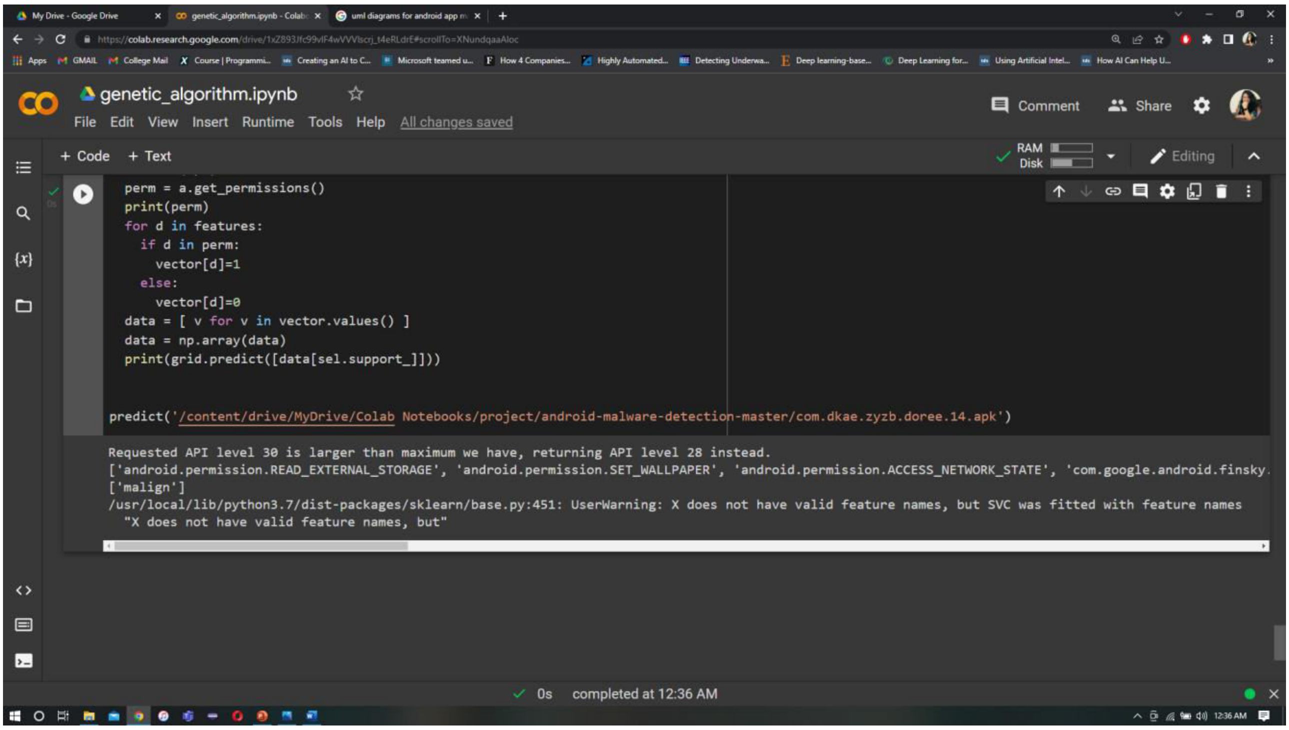
The malware and APK analysis methods were not discussed in detail and so it is better to analyse the accuracies of the methods are identified The novel ML/DL and other models which can be used to detect Android malware were also not in the focus of this review

In , a systematic review on DL-based methods for Android malware defence was discussed. Malware detection, malware family detection, repackaged/fake app detection, adversarial learning attacks and protections, and maliciousbehaviour analysis were identified as the malware defines objectives this review together with the usage of DL models.

Apart from Android malware detection techniques, source code vulnerability analysis is also important to address security concerns in Android. The survey analysed several studies on ML-based and data mining approaches which can be used to identify software vulnerabilities until 2017. Though this survey provides a good analysis, they considered most of the research work in general software security. Therefore, the vulnerability analysis in Android code was not discussed. However, findings such as ML models' usage for vulnerability analysis are still beneficial for specific programing languages' related analysis

V. Results

Theresultsoftheprojectwill be shown as bengin and malgin of the particular APK file. And checks using algorithm which is been implemented in our project using machine learning techniques -Using SVM classifier and so we can detect the particular android application is Malicious or not.

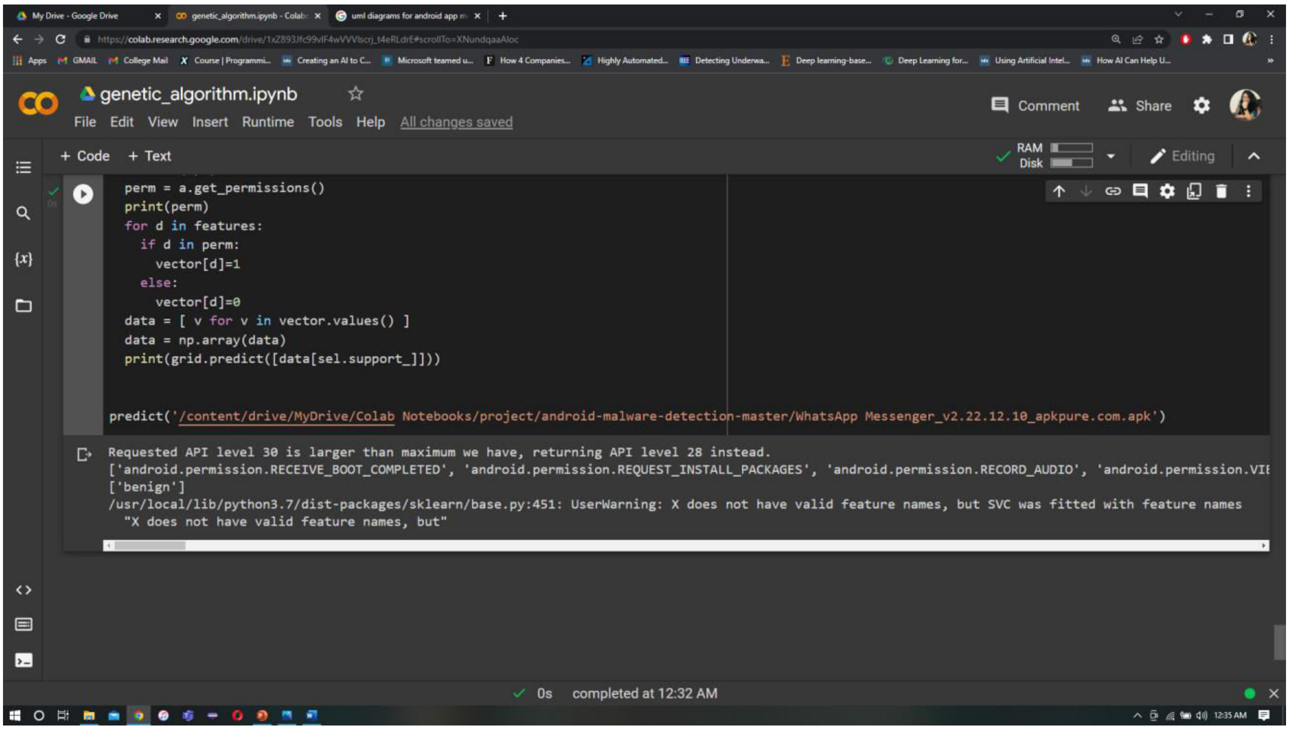


```
  perm = a.get_permissions()
  print(perm)
  for d in features:
    if d in perm:
      vector[d]=1
    else:
      vector[d]=0
  data = [ v for v in vector.values() ]
  data = np.array(data)
  print(grid.predict([data[sel.support_]]))


predict('/content/drive/MyDrive/Colab Notebooks/project/android-malware-detection-master/com.dkae.zyzb.doree.14.apk')
```

```
Requested API level 30 is larger than maximum we have, returning API level 28 instead.
['android.permission.READ_EXTERNAL_STORAGE', 'android.permission.SET_WALLPAPER', 'android.permission.ACCESS_NETWORK_STATE', 'com.google.android.finsky.
['malign']
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have valid feature names, but SVC was fitted with feature names
  "X does not have valid feature names, but"
```

Fig a. Malign



```
perm = a.get_permissions()
print(perm)
for d in features:
  if d in perm:
    vector[d]=1
  else:
    vector[d]=0
data = [ v for v in vector.values() ]
data = np.array(data)
print(grid.predict([data[sel.support_]]))


predict('/content/drive/MyDrive/Colab Notebooks/project/android-malware-detection-master/WhatsApp Messenger_v2.22.12.10_apkpure.com.apk')
```

```
Requested API level 30 is larger than maximum we have, returning API level 28 instead.
['android.permission.RECEIVE_BOOT_COMPLETED', 'android.permission.REQUEST_INSTALL_PACKAGES', 'android.permission.RECORD_AUDIO', 'android.permission.VIE
['benign']
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have valid feature names, but SVC was fitted with feature names
  "X does not have valid feature names, but"
```

Fig.b. Benign

Conclusion

Detecting Android malware in a quick and accurate manner is essential for Android OS users. To solve this problem, many studies have introduced machine learning for the detection of malicious applications, and feature selection has also been employed to speed up the process. In this work, we proposed an SVM-based malware detection scheme with the combination of genetic algorithm for Android platform and used permissions as features to build an SVM classifier, which can automatically distinguish malicious Android apps (malware) from legitimate ones. Experiment results show that the proposed scheme can identify malware APK files in an accurate manner

future scope

As for the future work the model can be extended by adding multiple functions that can check everything. The proposed model only includes the permissions to give the result but in future themodel can be also extended with different functionality check to give accurate and fast result. Which will be more beneficial in the developing phase for promoting the applications that will keep away the user from being attacked by the malware and keeps the user applications private.

REFERENCES

1. Al-Rfou et al. (2016) Al-Rfou R, Alain G, Almahairi A, Angermueller C, Bahdanau D. Theano: A Python framework for fast computation of mathematical expressions. arXiv. 2016. . http://arxiv.org/abs/1605.02688.Mr. Basavaraju S.R., Automatic Smart Parking System using Internet of Things (IOT), 2015.Mr.BasavarajuSR(ASPSusingIoT).pdf
2. Topgül, O.; Tatlı, E. The Past and Future of Mobile Malwares. In The 7th International Conference on Information Security and Cryptology; Springer: Berlin, German,2014; https://www.erdalozkaya.com/malware-past-present-and-future/
3. Aktas& Sen (2018) Aktas K, Sen S. UpDroid: updated Android Malware and its familial classification. In: Gruschka N, editor. Secure IT Systems. NordSec 2018. Lecture Notes in Computer Science. Vol. 11252. Springer;Cham:2018.https://www.mdpi.com/2227- 7390/9/21/2813/htm

4. Allix et al. (2018) Allix K, Bissyandé TF, Klein J, Le Traon Y. AndroZoo: collecting millions of android apps for the research community. MSR ’16 proceedings of the 13th international conference on mining software repositories, Austin, Texas; 2018. pp. 468471. https://ieeexplore.ieee.org/document/7832927
5. Liu, K.; Xu, S.; Xu, G.; Zhang, M.; Sun, D.; Liu, H. A review of android malware detection approaches based on machine learning. IEEE Access 2020, 8, 124579–124607. https://www.researchgate.net/publication/342614130_A_Review_of_Android_Malware_Detection_Approaches_Based_on_Machine_Learning