Detecting Android Malware Using Default and Customized Android Permissions

REU Fellow: Kendra Campbell[1], Faculty Mentor: Dr. Wenjia Li[2]
Affiliation: 1. School of Engineering and Applied Science, Hofstra University, Hempstead, NY 11549 2. School of Engineering and Computing Sciences, New York Institute of Technology, New York, NY 10023
Contact: *kcampbellmccalla1@pride.hofstra.edu*, *wli20@nyit.edu*

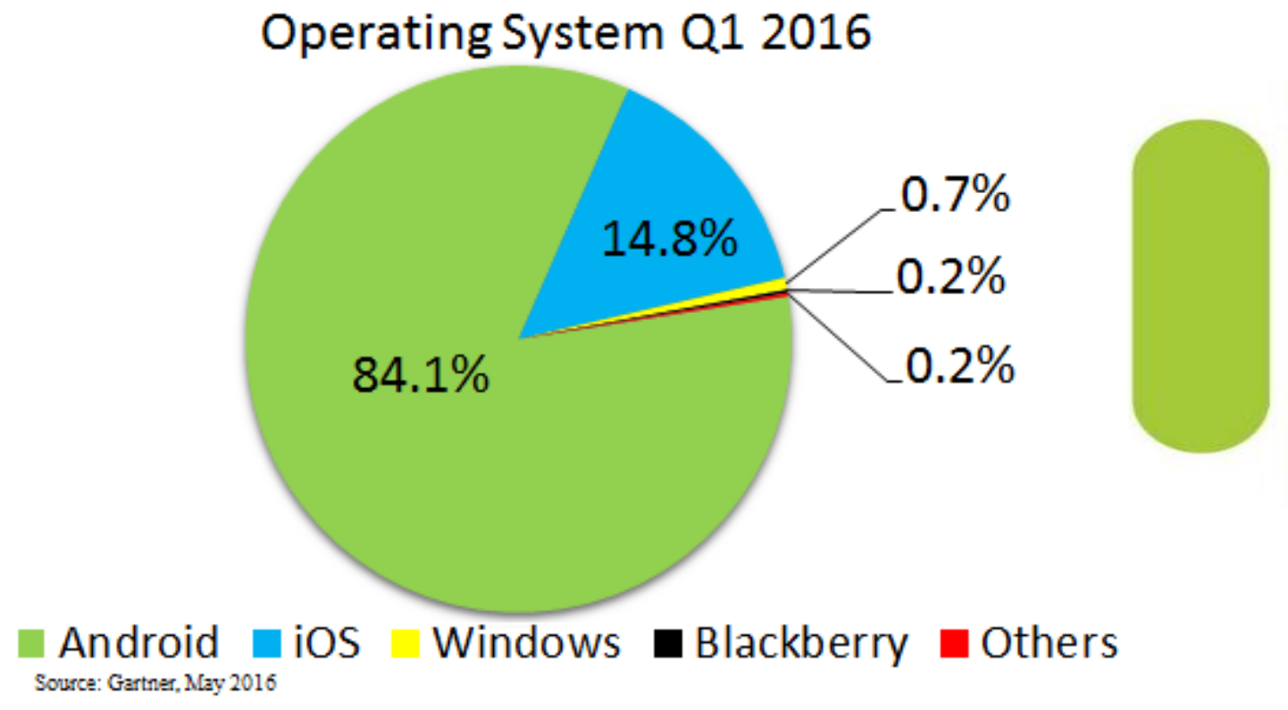
Abstract

With the growing popularity of Android systems, there has also been an increase in mobile malware specifically targeted at the Android operating system. Current research methods to detect mobile malware on android devices use a combination of static analysis, dynamic analysis, and machine learning. The goal of this research is to detect malware statically with the aid of a classification learner. After making modifications to the original project plan to achieve this goal, only the permissions used and defined in an application's manifest file were used to detect malware. Various features were sought after and extracted from each manifest file and were used to train a model using the Fine KNN algorithm. One model was found to have an overall accuracy of 94.5% and an overall error of 5.5%.

Introduction

The Android Operating System

Figure 1: Worldwide Smartphone Sales by Operating System Q1 2016

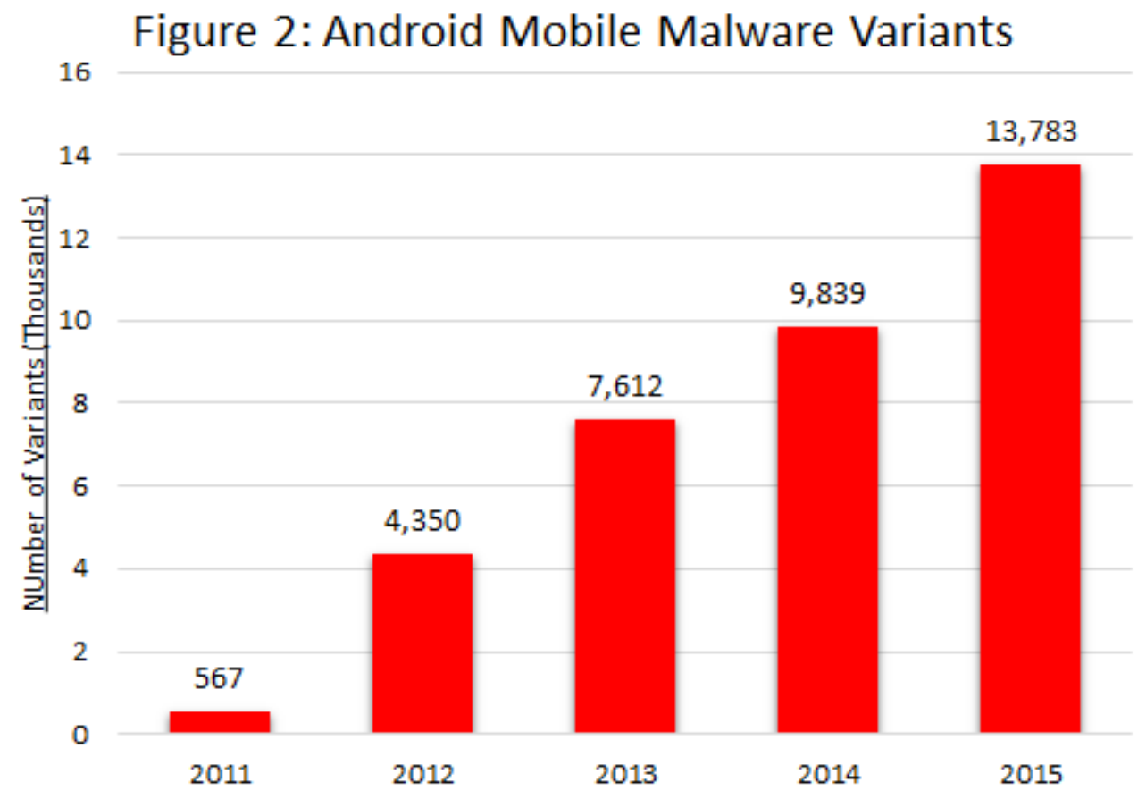


Source: Gartner, May 2016

Android is the most popular mobile device operating system today

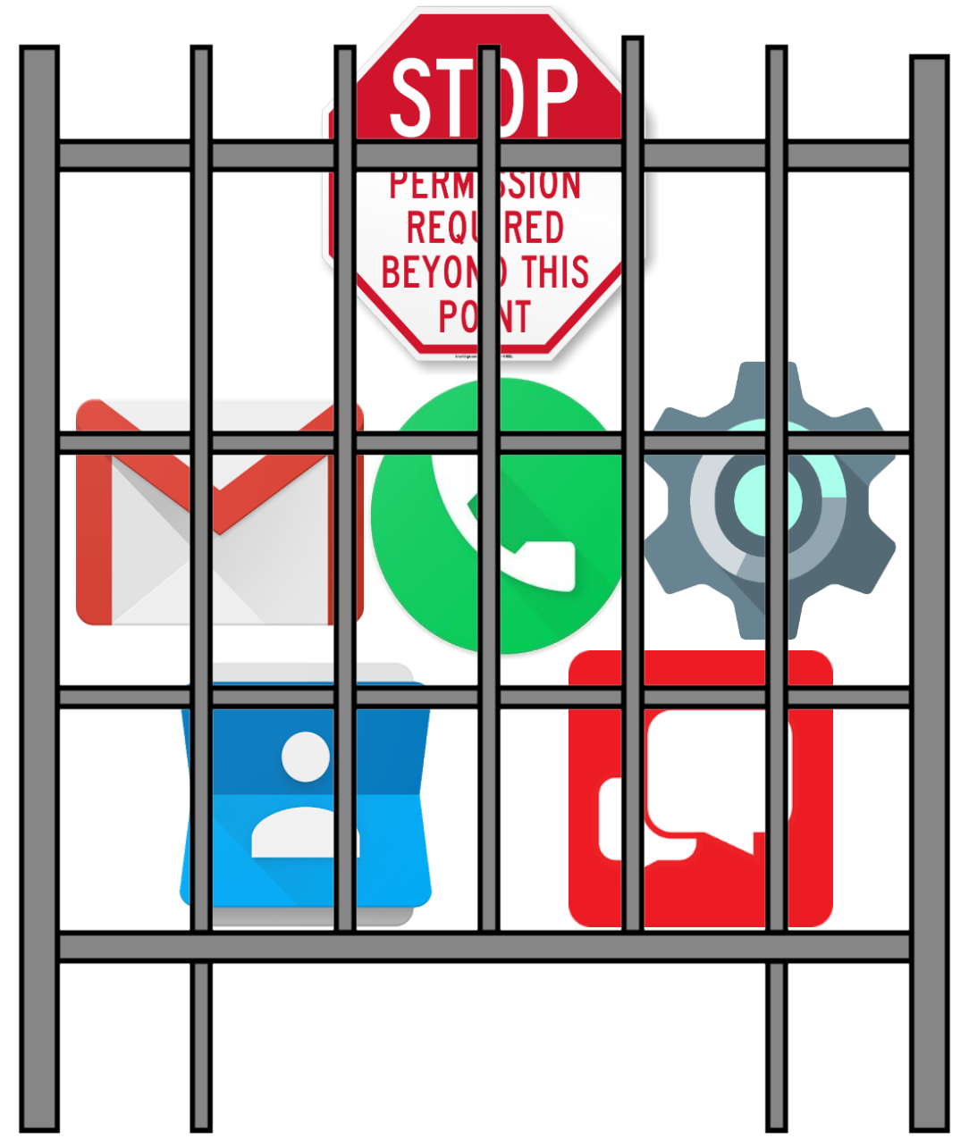
Android has also been plagued by malware in the recent years

Figure 2: Android Mobile Malware Variants



Source: Symantec, April 2016

Android Security



By default, no application has permission to perform operations that could harm other applications, the operating system, or the user

Applications access and share device resources and data by statically declaring the permissions they require in their manifest file

Android permissions are divided into several protection levels which tell the system the potential risk of the permission and the procedure to grant it

Research Objective

Use static analysis, machine learning, and information contained within an application's manifest file to detect malware

While also determining:

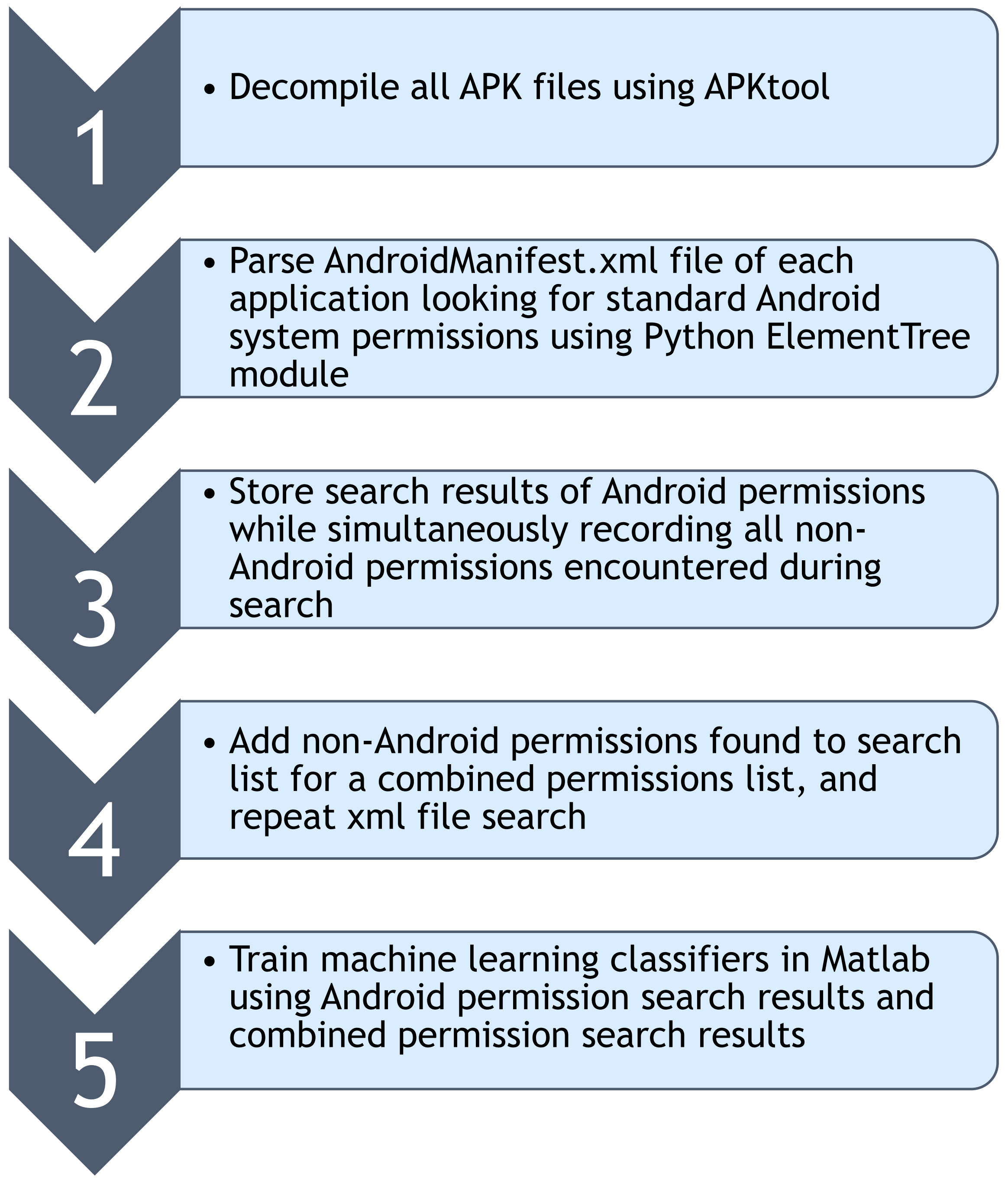
1. The most distinguishable Android provided system permissions between malicious and benign applications
2. The least distinguishable Android provided system permissions between malicious and benign apps
3. If malicious applications are more likely than benign apps to define their own custom permissions

Materials

The materials used to conduct this research were:

- APKtool 2.1.2, to decompile APK files
- Python 3.2.3, to read xml manifest files
- Matlab 8.6, to train machine learning algorithms
- 668 malicious and benign APK files (1,336 apps in total)
- Standard Android system permissions up to API level 24, to distinguish Android and non-Android permissions

Methods



1. Decompile all APK files using APKtool
2. Parse AndroidManifest.xml file of each application looking for standard Android system permissions using Python ElementTree module
3. Store search results of Android permissions while simultaneously recording all non-Android permissions encountered during search
4. Add non-Android permissions found to search list for a combined permissions list, and repeat xml file search
5. Train machine learning classifiers in Matlab using Android permission search results and combined permission search results

Results

Initial testing of the Android permission data set revealed the Fine KNN algorithm performed the best with the training data

Figure 3: Algorithm Overall Accuracy

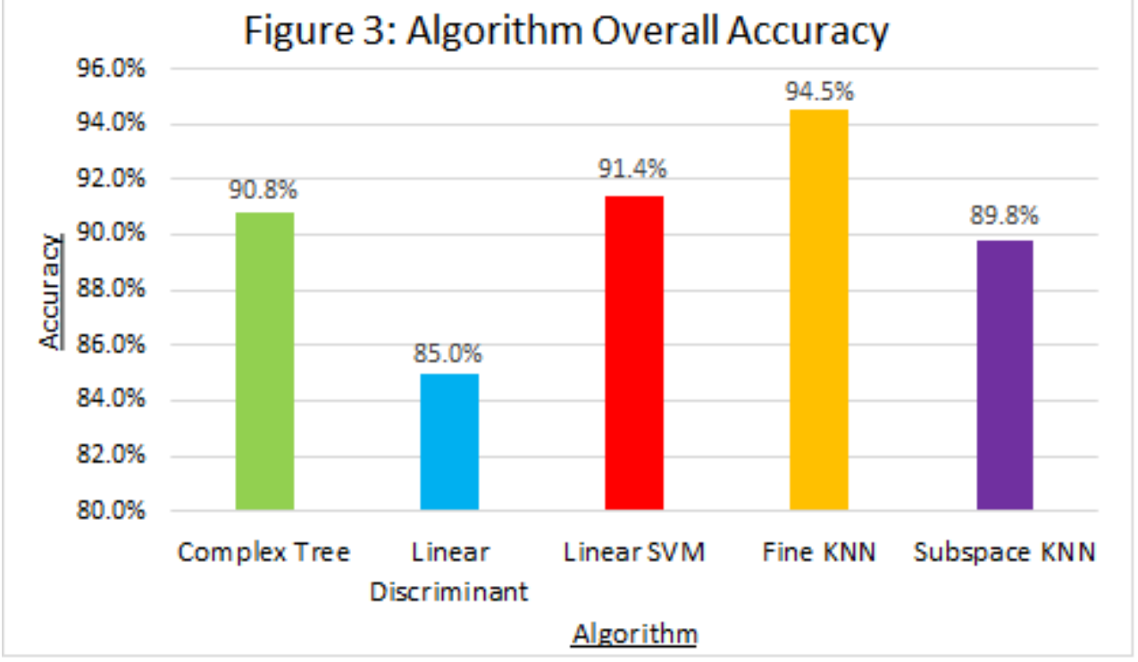
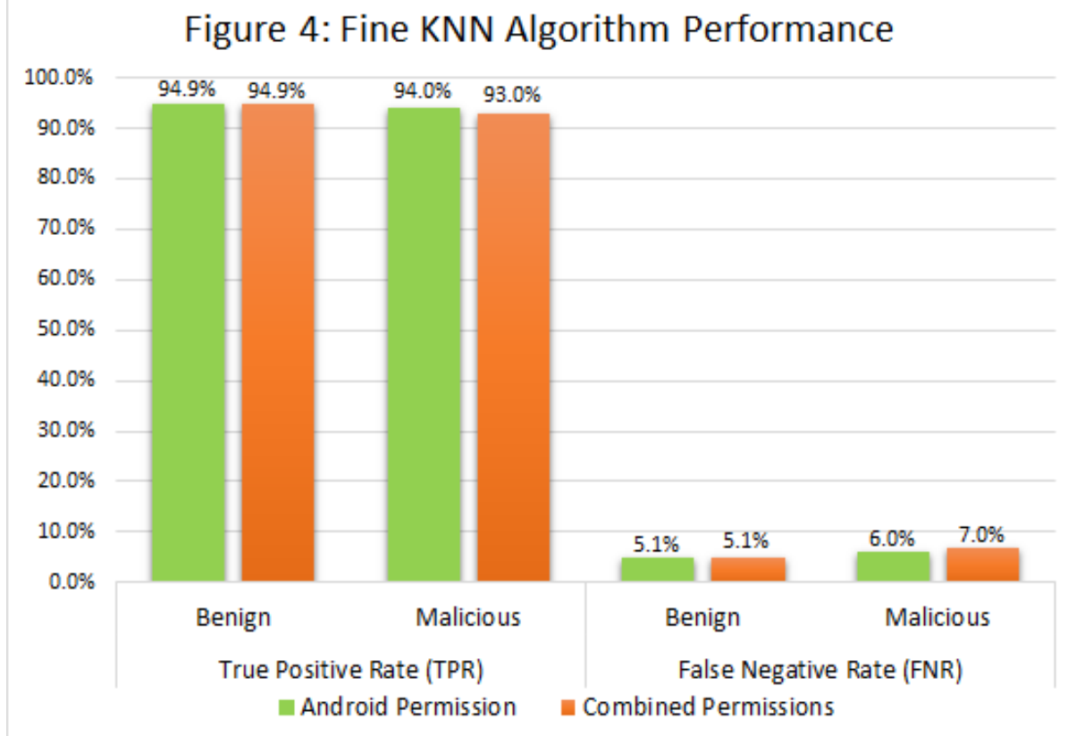


Figure 4: Fine KNN Algorithm Performance



The Android permission data set was found to have better accuracy than the combined permission data set and further analysis was performed using only the Android data

Ten permissions were used more by either malicious or benign apps by 12% - 64%. Five of them had a dangerous protection level and were used more by malicious apps

Figure 5: Most Distinguishable Android Permissions

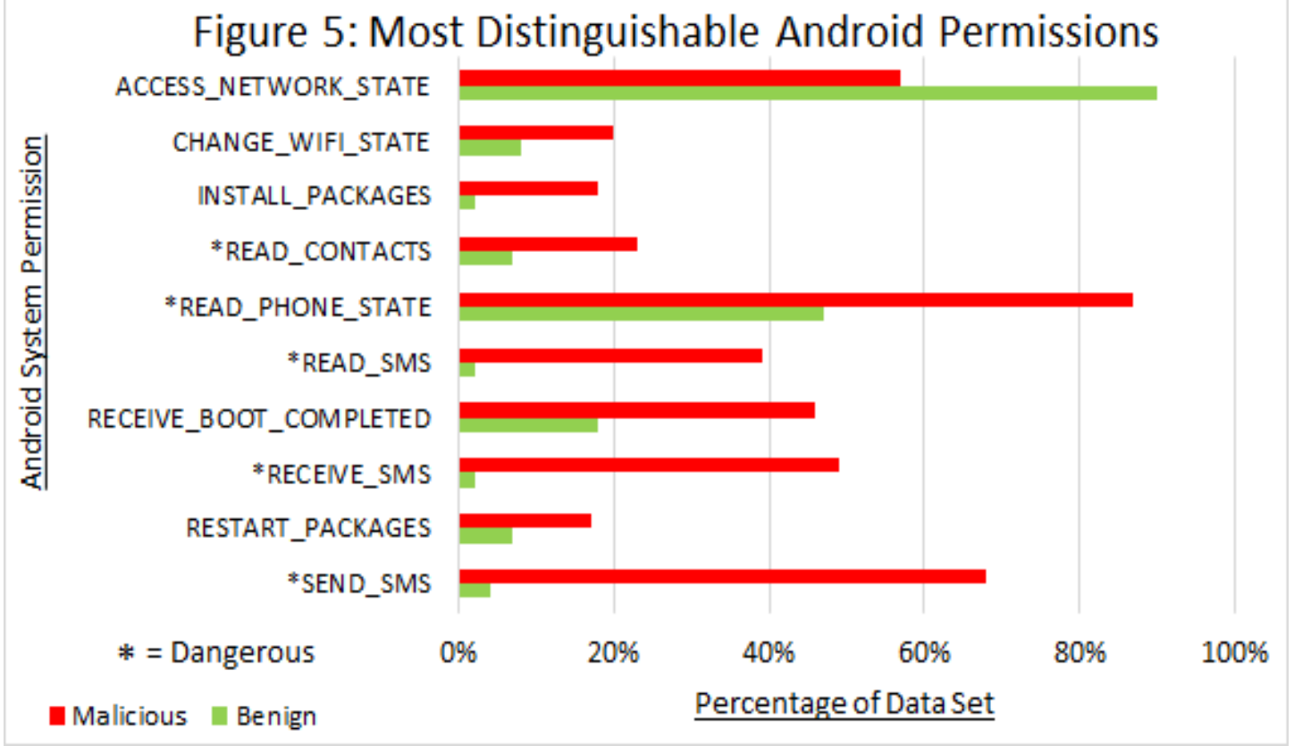
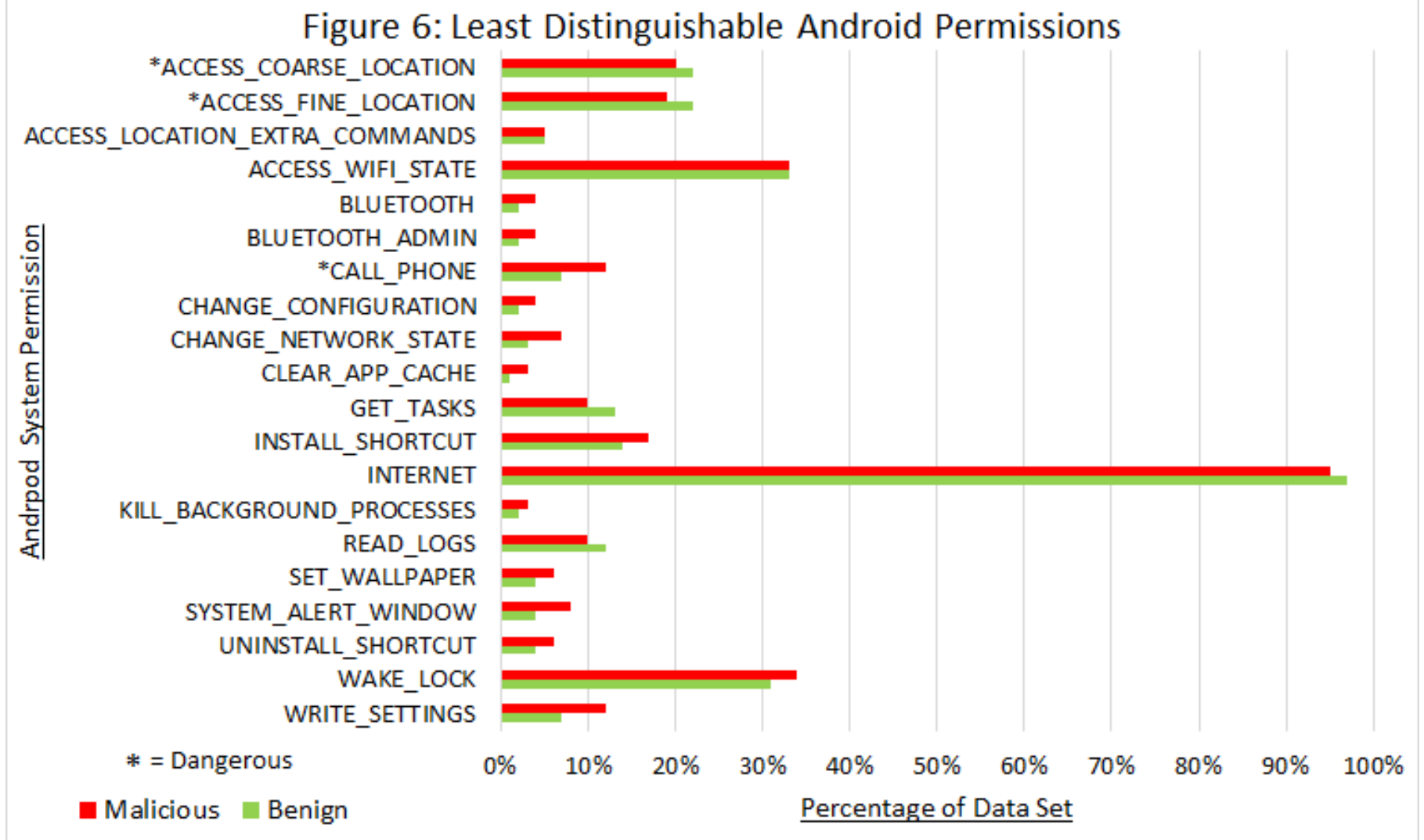


Figure 6: Least Distinguishable Android Permissions



For twenty permissions, the percentage difference of the number of benign and malicious apps that used them was between 0% - 5%. Of those twenty, three were dangerous

Figure 7: Benign Apps Permission Definition

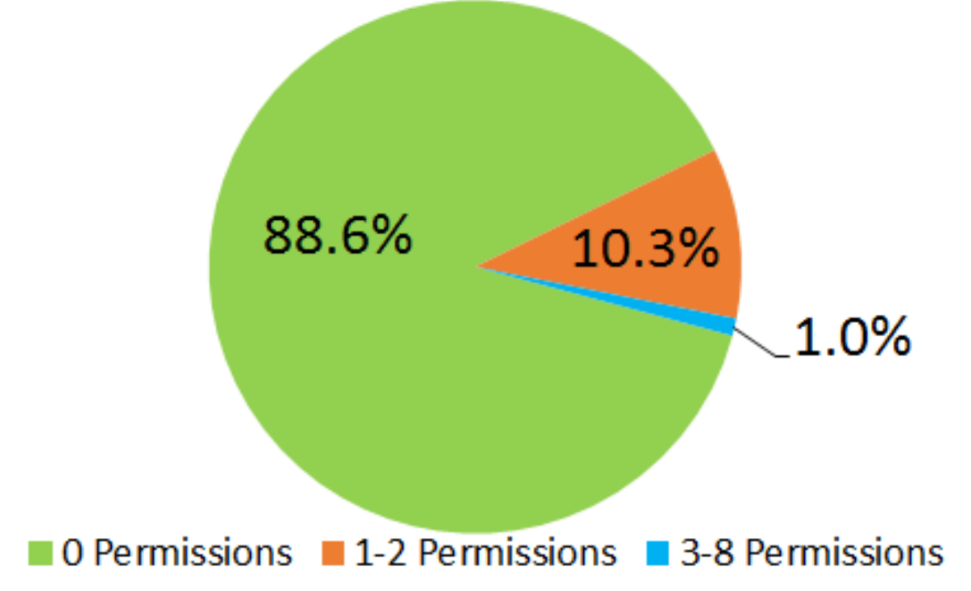
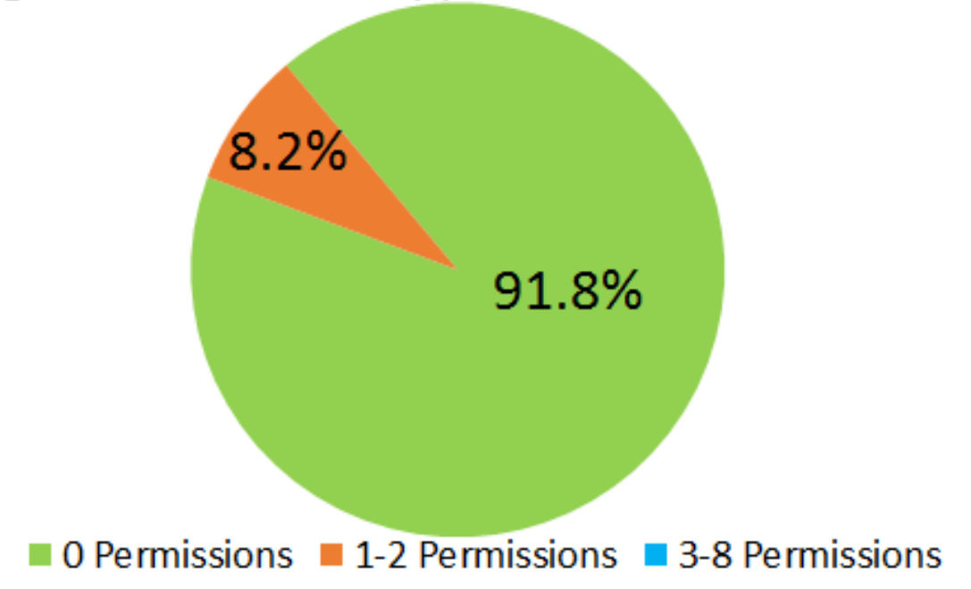


Figure 8: Malicious Apps Permission Definition



Malicious applications created their own custom permissions less frequently than benign applications. Benign applications were also found to create more permissions per app than malicious applications

Conclusion

- There was no major overall difference between classifier performance when trained with the Android permission data versus the combined permission data. Thus the smaller Android permission set, which is readily available to the public, is sufficient alone to detect malware
- This study further supports the statement that permissions alone cannot be used to detect all types of malware. It did find, however, that malicious applications are more likely to use the following dangerous permissions: READ_CONTACTS, READ_PHONE_STATE, READ_SMS, RECEIVE_SMS, SEND_SMS
- There are more Android permissions likely to be used by both malicious and benign apps than those used more frequently by only on type
- The percentage of benign and malicious applications that did not define any custom permissions, or defined one or two custom permissions, was relatively indistinguishable. An interesting find, however, was that no malicious app created more than two custom permissions, while some benign apps created up to eight
- More than 90% of malicious applications did not define any custom permissions. The number of malicious apps that did was still fewer than the benign. It could be that malicious apps are more concerned with gaining access to other features as opposed to safeguarding their own. The creation of custom permissions does not seem to aid malware in performing their devious operations

References

[1] Gartner, Inc, "Gartner Says Worldwide Smartphone Sales Grew 3.9 Percent in First Quarter of 2016", 2016.

[2] Symantec Corporation, "Internet Security Threat Report 2016", 2016.

[3]"System Permissions | Android Developers", *Developer.android.com*, 2016. [Online]. Available: https://developer.android.com/guide/topics/security/permissions.html. [Accessed: 27- Jul- 2016].

Acknowledgements

The project is funded by National Science Foundation Grant number CNS-1559652 and New York Institute of Technology.

I would like to thank:
Wenjia Li Ph.D.
Monica Kumaran
Chereese Sheen
Schwann Chan
Cameron Cai