Mobile malware is a threat to smartphone users. Infection rates increased by 96% in the first half of 2016 and by 83% in the second half. In 2016, Nokia's collection of mobile malware samples increased from 600,000 to 12,000,000. [9] Malware can be used to steal money and data from Android devices.

Distinguishing between benign and malicious applications is important. In this project we used API calls and permissions as features for machine learning. We trained six classifiers and achieved up to 75% accuracy. Overall, using permissions is slightly more accurate than using API calls. However, the best accuracy came from using API calls with Neural Network Classifier.

In 2014, Forbes reported that 97% of malicious mobile applications were for the Android operating system [7].

Judy malware: estimated to have infected as many as 36.5 million Android phones as of May 2017 [8].

Goal: to identify features of Android applications that can be used to accurately distinguish between malware and benign Android apps.

Methodology: This will be accomplished through machine learning.

Terms and Definitions

API: Application programming interface, allows communication between software components. (e.g., send text)

Android Manifest: An xml file that provides information about the app to the system (e.g., permissions, activities, services)

Method Signature: Tuple of information that identifies a method in the following manner: Package.Class.Method(Parameters) Return Type. (e.g., Android.location.Address, setFeatureName(Ljava/lang/String;)V)

Feature Vector: Vector of features (API/Permission names) representing an application

Cross Validation: Procedure for testing classifiers. Data is divided into n folds. Classifier is trained and tested n times with a different fold as the test data each time.

Figure 1. Classifier Training Process. This process occurs once for every classifier. Feature extraction is done through Apktool [2]. Androguard [1] and JavaP [3]. Classifiers are from Scikit-Learn python library [4]. Our benign applications (499) came from Google Play [5]/PlayDrone [11] and our malware (552) came from AndroidSandbox [6].

Figure 2. Classifier Testing Process. This process occurs once for every application every time we test a classifier.

Figure 3. This data is the average accuracy of using 8-fold cross validation with each classifier on the set of applications.

Table 1. This data was obtained by training the 3 most accurate classifiers, RF (Random Forest), LR (Logistic Regression), and NN (Neural Network), and testing them on a set of 100 applications (50 benign and 50 malicious). Positive=malicious, negative=benign.