Final Project Abstract

Team Name: Cam Spotter
Usernames: hdeclerc, doyeka, jcontre2

Hypothesis

In 2019, the percentage of Android device users around the world was 86.6%. With such a high percentage rate, we sought out to investigate how secure Android devices are. We wanted to test the following hypothesis: The frequency of malware attacks on Android devices increases due to newly developed threats that bypass Android security protocols.

Data

We gathered data from Nokia and Google Android's yearly security reports for information concerning Potentially Harmful Applications (PHA malware) and their install rates and trends on Android devices. We also found it relevant to collect more data about Android malware detection which allowed us to create three regression models on Android application permission types. The other source of data we found investigates the Android application package files (APK) and correlations between security characteristics of the apps (i.e. we focused on the overprivileged applications and the types of permissions they have). A list of links to the data is as follows: http://darwin.rit.edu/reports, https://pages.nokia.com/T003B6-Threat-Intelligence-Report-2019.html, https://securelist.com/mobile-malware-evolution-2019/96280/, https://docs.greynoise.io/#greynoise-api, https://source.android.com/security/reports/Google_Android_Security_2018_Report_Final.pdf, https://figshare.com/articles/Android_malware_dataset_for_machine_learning_2/5854653

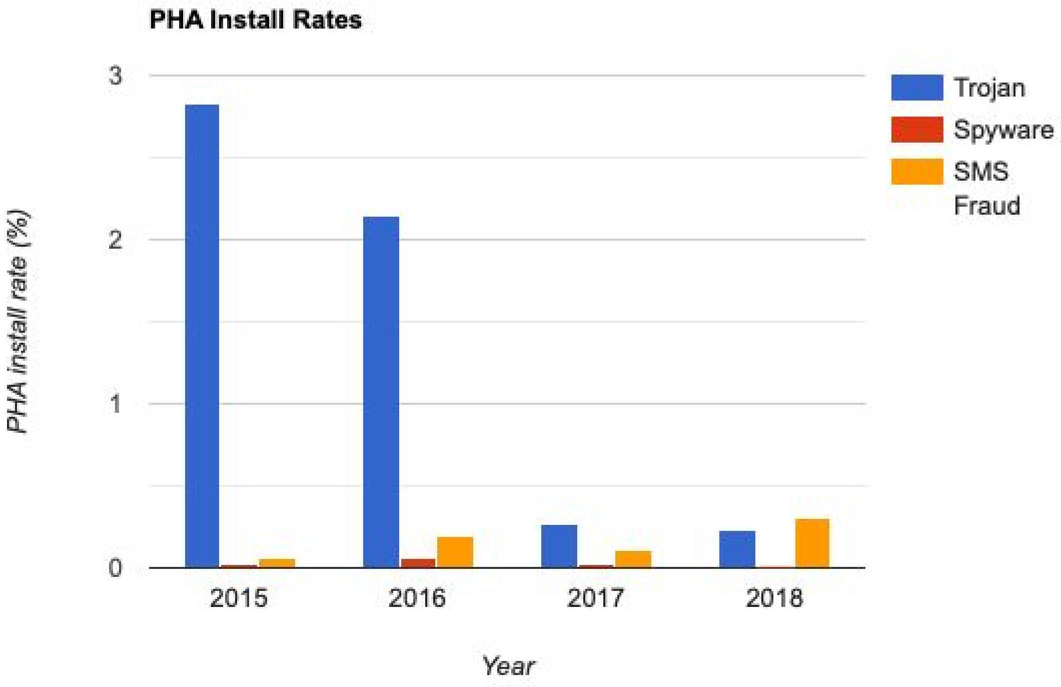
Capstone data (Computer Vision portion):

We collected and labeled image data by utilizing a modified API to scrape Google Images https://github.com/Joeclinton1/google-images-download for images of a variety of different phones. For our generated image data, we simply used bursts of iPhone photos of desired scenarios with a variety of different people. We also utilized OpenCV image augmentation to create an additional 9000 images in addition to the original 3000 images. We generated this data by mirroring, and slight rotations of the original image data.

Findings

Claim #1: An increase in potentially harmful applications (PHAs) results in a decrease in malware attacks on Android devices.

Support for Claim #1: Although we saw an increase in the number of PHA types and categories from the years 2015-2018, we saw a decreasing trend in the efficacy of PHA install rates across Android applications. We focused on the three most prevalent PHA's in the Android malware ecosystem: Trojan, Spyware and SMS Fraud. This decrease in malware attacks from 2015-2018 can be seen in Figure 1.
Figure 1:



Claim #2: There is no evidence to suggest apps with more permissions are more likely to be malicious
Support for Claim #2: We ran a multiple regression model to show the relationship between 114 variables and whether or not an app was malicious. We found that the coefficient of number of permissions was not significant with a value of .002 while allowing the ability to send messages was significant with a coefficient value of .441. Below are the coefficients and p-values of the 5 most significant variables as well as the # of permissions variable.

Multiple Regression		
Variable	**Coefficient**	**P-Value**
SEND_SMS (1=permissions granted)	**.441**	**.000**
DELETE_CACHE_FILES (1=permissions granted)	**.330**	**.000**
CONTROL_LOCATION_UPDATES (1=permissions granted)	**.276**	**.000**
SET_ACTIVITY_WATCHER (1=permissions granted)	**-.274**	**.002**
ACCESS_LOCATION_EXTRA_COMMANDS (1=permissions granted)	**.262**	**.000**
# OF PERMISSIONS	**.002**	**.030**

Capstone Claim #1: We were not able to achieve 70% accuracy classifying phones in images.

Support for Capstone Claim #1:

We wanted to test whether our network was able to recognize images of mobile phones. To do this, we built a convolutional neural network, of which you can find more details about in our handin. After our final round of testing, we obtained the accuracies noted in the table below. In conclusion, this was likely due to limited data and a lack of generalization of our model. We utilized a few different strategies which aided the model to improve significantly (dropout, batch_normalization, etc.) but not up to our goal of 70%.

Convolutional Neural Network Accuracy	
Network only chooses yes.	**73%**
Network only chooses no.	**27%**
Network guesses randomly.	**49%**
True prediction accuracy.	**64%**