Lightweight Malware Detection based on Machine Learning Algorithms and the Android Manifest File

REU fellow: Monica Kumaran[1], Faculty mentor: Dr. Wenjia Li[2],
Affiliation: 1. College of Engineering, University of California at Berkeley, Berkeley, CA 2. NYIT School of Engineering and Computing Sciences, New York Institute of Technology, New York, NY 10023
Contact: monica.kumaran@berkeley.edu, wli20@nyit.edu

Abstract

Malware greatly harms victims by collecting and selling personal data to advertisers and phishers, selling mobile device identification numbers, or initiating expensive premium calls and texts without user permission. As the most popular mobile operating system, the open-source Android platform is particularly susceptible to malicious apps; in fact 98% of all mobile malware is targeted at the Android marketplace. In this study I compare the effectiveness of different machine learning algorithms classifying apps as malicious and benign based on requested permissions and inter-app intent communication. I also gauge the efficacy of using permissions versus intent filters, two elements from the Android manifest file. I find that a Cubic Simple Vector Machine (SVM) algorithm was the most accurate classifier at 91.7% accuracy, and that while permissions are a significantly more accurate feature set, combining them with intent filters does improve the classifier.

Android Marketplace

Android is the most popular smartphone platform today.

As of 2016 65 billion apps have been downloaded from Google Play store.

Tools Used

Apktool to decompile apps from apk files

Matlab for machine learning algorithms

Python 2.7 to automate data collection from app manifests

Methodology

I extracted 183 features from Android manifest files. Every app has an AndroidManifest.xml file which contains information the system needs to run the app.

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.software.application">
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
    <uses-permission android:name="android.permission.READ_SMS"/>
    <uses-permission android:name="android.permission.SEND_SMS"/>
    <uses-permission android:name="android.permission.RECEIVE_SMS"/>
    <application android:debuggable="true" android:icon="@drawable/icon" android:label="@string/
app_name">
        <receiver android:name=".Notificator">
            <intent-filter>
                <action android:name="android.intent.action.BOOT_COMPLETED"/>
                <category android:name="android.intent.category.HOME"/>
            </intent-filter>
        </receiver>
        <receiver android:name=".Checker">
            <intent-filter>
                <action android:name="com.software.CHECKER"/>
                <category android:name="android.intent.category.HOME"/>
            </intent-filter>
        </receiver>
        <receiver android:name=".SmsReceiver">
            <intent-filter>
                <action android:name="android.intent.action.DATA_SMS_RECEIVED"/>
                <data android:scheme="sms"/>
                <data android:port="8901"/>
            </intent-filter>
        </receiver>
        <activity android:configChanges="keyboardHidden|orientation" android:label="@string/
app_name" android:name=".Main">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <activity android:name=".OffertActivity"/>
        <activity android:name=".ShowLink"/>
        <service android:name=".C2DMReceiver"/>
        <receiver android:name="com.google.android.c2dm.C2DMBroadcastReceiver"
android:permission="com.google.android.c2dm.permission.SEND">
            <intent-filter>
                <action android:name="com.google.android.c2dm.intent.RECEIVE"/>
                <category android:name="com.software.application"/>
            </intent-filter>
            <intent-filter>
                <action android:name="com.google.android.c2dm.intent.REGISTRATION"/>
                <category android:name="com.software.application"/>
            </intent-filter>
        </receiver>
    </application>
    <permission android:name="com.software.application.permission.C2D_MESSAGE"
android:protectionLevel="signature"/>
    <uses-permission android:name="com.software.application.permission.C2D_MESSAGE"/>
    <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
</manifest>
```

Figure 1. A Sample Android manifest file

The extracted features fell into three categories.

Requested Permissions	Declared Permissions	Intent Filters
Android apps must request user consent to access functionalities such as location, camera, or contacts.	Android apps can create their own permissions, which can be used as an extra security layer against other apps accessing their data. I had a single variable marked true if an app created any of its own permissions.	Intents are used to request actions from other app components. Apps can send implicit intents to other apps requesting information. Intent filters tell the system which implicit intents the app can handle and what overall phone states the app wants to know.
154 features	1 feature	28 features

After extracting information from the original dataset, I trained different machine learning algorithms on a training set of 500 benign and 500 malicious apps. I used a ten fold validation scheme to determine accuracy.

Data

To compare the effectiveness of using intent filters versus permissions, I trained classifiers for a dataset of only intents, only permissions, and both combined.

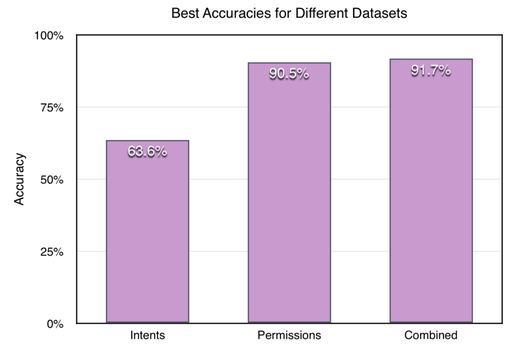

Figure 2. Graph of best accuracies achieved using the three datasets

I also compared different machine learning approaches to the combined dataset.

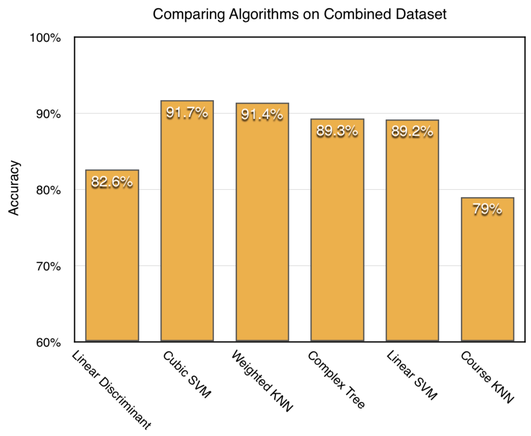

Figure 3. Graph of different algorithm performances on combined dataset

The Cubic Simple Vector Machine (SVM) created the most accurate classifier.

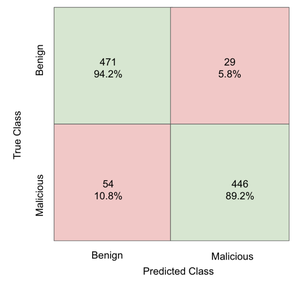

Figure 4. Confusion matrix of Cubic SVM

The True Positive Rate (TPR)/False Negative Rate (FNR) of benign apps is 94.2%/5.8%.

TPR/FNR of malicious apps is 89.2%/10.8%.

Discussion

- At 63.6% accuracy, analyzing only the intent filter set is not effective
- Permissions are a more reliable feature set
- Increasing overall accuracy by 1.2%, using intent filters along with other metrics is still useful
- Possible to use only the manifest file to create a viable classifier
- For the combined dataset the top four algorithms (Cubic SVM, Weighted K-nearest neighbors, complex tree, and linear discriminant) had similar performances despite different approaches
- Further refinement of Cubic SVM and others may be necessary

Further Work

- Pursue intent analysis further by analyzing source code
- Refine machine learning algorithm
- Expanding intent filter feature set

Acknowledgement

The project is funded by National Science Foundation Grant No. CNS-1559652 and New York Institute of Technology.

Special thanks to Dr. Li and Kendra Campbell.

References

"Topic: Android." Www.statista.com. Statista, n.d. Web. 02 Aug. 2016. <http://www.statista.com/topics/876/android/>.

Neal, Dave. "Android Is Target for 98 Percent of All Mobile Malware | TheINQUIRER." Http://www.theinquirer.net. N.p., 26 Feb. 2014. Web. 02 Aug. 2016. <http://www.theinquirer.net/inquirer/news/2331127/android-is-target-for-98-percent-of-all-mobile-malware>.