Android Attacks

Ofir Magen And Yuval Ben Yaakov

January 19, 2023

Abstract

Various malicious application has many common factors such internal storage permissions or sending texts messages. so a good machine learning model can recognize those patterns and classify with a high accuracy if a given app is malicious or benigin. The problem starts when the user creates and adversarial example which takes the benigin/malicious app and change the feature space in a way the app will remain benigin/malicious but the model will classify as the opposite. In this case we will need a robust model that will classify the adversarial example as wanted.

this model use combination of 2 existing models: MLP and FD-VAE and each sample is represented by 379 features which are API Calls, Permissions and intent actions.

our goal is to attack the model by using with white box by giving the malicious application features to manipulate the model

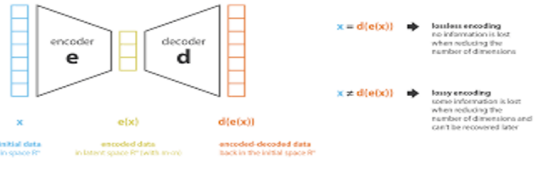
1 Introduction

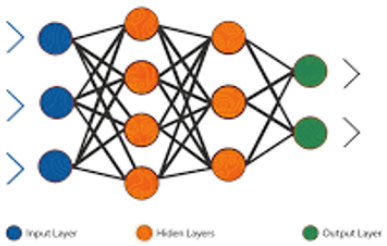
These days, Android is the most popular operating system. From smart televisions to mobile phones etc.
This operating system is an open source code. and this is the reason for Android operating system being familiar. In addition, this is the main reason for Android being easy to attack. in result, there has been needs to protect the system. the protection reflected by machine learning and algorithms to identify malicious apps.
One of the most familiar models is Robust detection model. This model is a combination of two existing models: MLP and FD-VAE and each sample is represented by 379 features which are API Calls, Permissions and intent actions. There is no such model that is immune to attack. that's the reason why this model is open to attack. There are few known attacks. For example, features attack based on features build in Manifest.XML. The Manifest.XML file is includes app's permessions.
Unfutenatelly, There are some defence models that can predict malicious apps or benigin apps but there is no defence mechanism that can prevent one hundred percent malicious attacks.

2 Related works

3 Methodology

Robuse detection model is a combination of two existing models: MLP and FD-VAE and each sample is represented by 379 features which are API Calls, Permissions and intent actions.
FD-VAE : Feature Driven Variational Encoder it consists of 2 parts encoder and a decoder , the encoder role is to reduce the dimension of the given feature vector to a low dimension representation, the decoder role is to take the low dimension representation and convert it back to the original feature space, then the model checks the reconstruction error and if it passes a predetermined threshold then the model will classify the sample as malicious, the guidelines to this model is that benigin samples will have similar encoding.
MLP: Multi Layer Perceptron is a deep learning model that has one or more hidden layers. In this model MLP is used to predict if the sample is bengin or malicious/adversary.

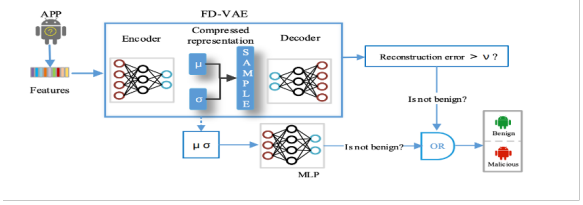




3.1 Feature Extraction

As mentioned on Introduction section, Robuse detection model represent every sample as a 379 feature vector with values of 0,1. The features are 126 intent actions, 106 sensitive api calls, 147 permissions. Apktool is used on every application in the dataset, and the permissions are extracted from the AndroidManifest.xml file, the sensitive api calls and the intent actions are pulled from smali code.

Smali code – human readable generated code using Apktool from the .dex class which is the a machine code instruction of the apk file



3.2 adversarial Example Attacks Ideas

After being familiar with the Robuse detection model, it's the time to attack.
First idea gone up is a white box attack and uses the idea that the model has static analyze (379 pre-chosen features) so we can take a benigin apk and add it a permission/intent action/api calls that are not compiled into features, that turn the apk malicious.
Second idea is another white box attack that requires knowing the data the model has trained on and creating an adversarial attack that it yet seen (or anyone has yet seen which is more powerful, because we don't need to know the training dataset).
Our black box idea was using similar attack to mal-gan which can can learn to generate an adversarial example attack the model has yet seen (because the model will probbably classify correctly adversarial examples he has seen).
Trying to add noise to the smali code or the Androidmanifest.xml for a black box attack will probbably will have no effect because the model has predetermined 379 features and will ignore the noise.
Trying to do any gradient based attacks will probably wont work because there are 2 models that can decline the sample (FD-VAE and the MLP) so we might outsmart the loss of the MLP but the FD-VAE might decline the sample.

4 Results

we was try to run the model but we got multiple errors..

5 Conclusion

There was a few interesting ideas to attack the Robuse detection model, and even though the model is called robust it is also seems that there are Adversarial example attacks that will work, especially if the model did now have enough data to train on to classify the adversarial examples as malicious.
There are some few ways to improve the defensive operations. Even though it's important to know that all models are exposed to attack. This is the main reason for Android users being so volunarible to attacks on daily basis routines and operations like downloading new applications, opening PDF files, opening images, camera opening premessions and even more. those operations are so simple and basic for using devices that support Android operation systems. Users MUST be aware to the operations they make while using devices supporting android operation systems and watch out.
However, Android operation system is developed and improve her defensive operation mechanisms on daily basis. Users must update the new operation system versions that comes out and we already know that google play maps the apps on device as well to reach a highly defensive mechanism.
In spite of everything, There is and will be risks by downloading apps from App store or by doing daily basis operations.