2ª MOSTRA DE TALENTOS DA GRADUAÇÃO

PUC CAMPINAS
PONTIFÍCIA UNIVERSIDADE CATÓLICA

Centro de Ciências Exatas, Ambientais e de Tecnologias (CEATEC)

Androlherme

Guilherme Ramos Candido (guilhermercandido@gmail.com)

Orientador: Prof. Me. Edmar Roberto Santana de Rezende

CENTRO DE CIÊNCIAS EXATAS, AMBIENTAIS E DE TECNOLOGIAS
FACULDADE DE ENGENHARIA DE COMPUTAÇÃO

Introdução

Nos dias atuais, a tecnologia está cada vez mais acessível para a população e, por este motivo, o número de usuários de smartphones aumenta gradativamente a cada ano.
Os smartphones não são mais utilizados apenas para ligações, mas sim em qualquer tipo de situação, desde conversas através de aplicativos de mensagens, postagens de fotos e vídeos e também transações bancárias.
Diante desse cenário, a oportunidade para crimes através desses aplicativos também aumentou.
Os *malwares* para dispositivos *Android* estão presentes inclusive na loja de aplicativos oficial e eles podem causar sérios danos aos usuários, desde senha de redes sociais divulgadas até roubo de informações sobre transações bancárias.
Com todas essas opções de uso, a necessidade de segurança envolvendo esses tipos de aparelhos vem aumentando. O uso de ferramentas que auxiliam na análise e classificação de *malware* com características semelhantes em família podem auxiliar a combater essa onda de ataques maliciosos pois possibilitam a criação de estratégias de combate a um conjunto de aplicativos maliciosos semelhantes.

Objetivo

As ferramentas existentes para análise de *malware*, em sua maioria, realizam uma análise estática, ou seja, realizam a análise a partir da extração do APK e do arquivo AndroidManifest.xml.
Embora essas ferramentas consigam exibir as informações corretamente, as informações são exibidas sem nenhum tipo de tratamento ou muitas vezes são superficiais para um especialista chegar a alguma conclusão, fazendo com que a dificuldade em interpretar essas informações seja muito alta. A classificação do *malware* em uma família com características semelhantes também se torna muito complicado utilizando essas ferramentas.
Por este motivo, o objetivo deste projeto é desenvolver uma ferramenta capaz de facilitar a análise e classificação de aplicativos *Android* maliciosos simplificando o processo de extração de informações sobre o APK e automatizando o processo de classificação do aplicativo malicioso em famílias de malware com características semelhantes.

Materiais e métodos

Para realizar o treino do algoritmo de classificação *Support Vector Machine* (SVM), foi utilizada a base de *malwares* já rotulados do Drebin. A partir dela, foram escolhidas três famílias (Plankton, DroidKungFu, GinMaster) para realizar a extração das duas *features* principais, que são: *Opcodes* e Permissões.
Para melhorar o resultado da classificação, foram utilizados dois recursos: *StratifiedKFold e* TF-IDF.
Para a análise do *malware*, foram utilizadas algumas técnicas de visualização, como gráficos e tabelas interativas.

Resultados e Discussões

Após realizar a submissão de 1000 *malwares*, o classificador foi treinado utilizando três *features* diferentes: Permissões, *Opcodes* e Permissões + *Opcodes*.

A acurácia média obtida entre os três classificadores foi de 88%.

O classificador treinado apenas com as permissões obteve uma acurácia de 98%, como é demonstrado na matriz de confusão representada pela Figura 1.

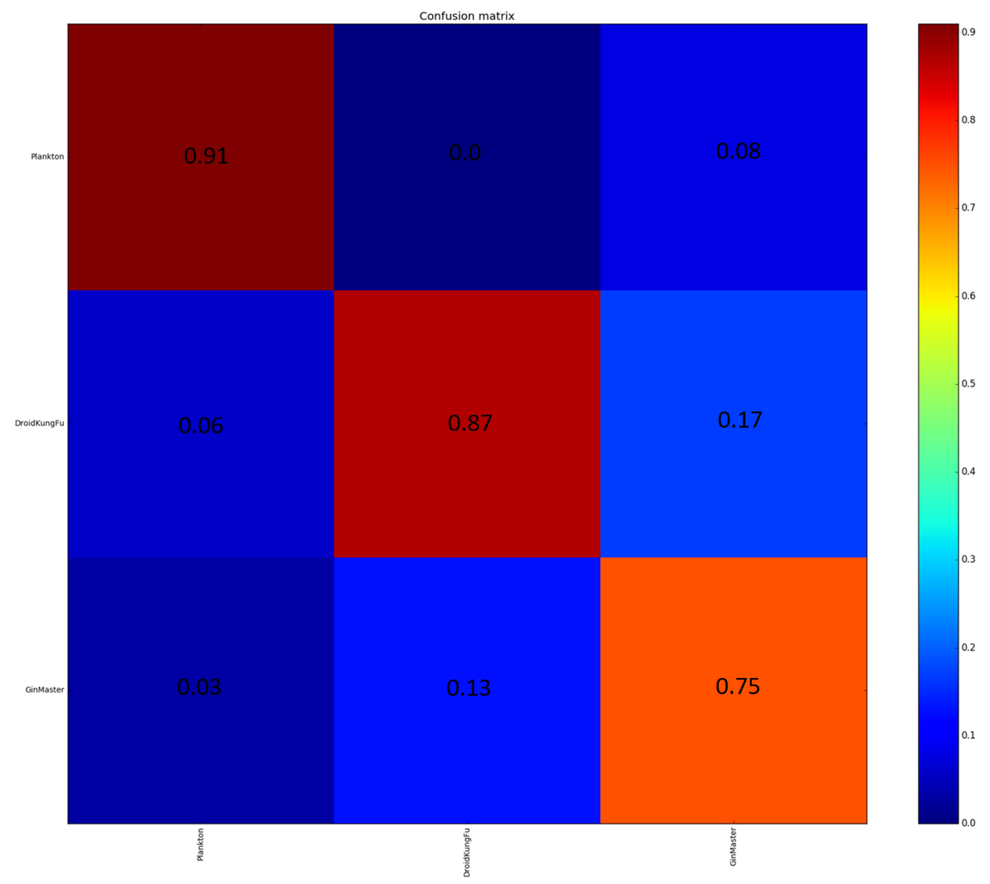

Figura 1 – Matriz de confusão

Comparando o Androlherme com outras ferramentas que realizam este tipo de na análise, o esforço para submeter o aplicativo, analisa-lo e classifica-lo foi classificado como baixo. Isto é devido à facilidade em identificar as informações relevantes e a possibilidade de relacionar essas informações, de forma que a análise fique mais conclusiva.

Conclusões

- A utilização das técnicas de visualização proveu muita facilidade para realizar a análise
- A acurácia eficaz do aprendizado de máquina, permite agrupar os exemplares em família. Com isso, é possível verificar quais as características principais de cada família e, a partir disso, desenvolver um mecanismo para poder combater este tipo de aplicativo malicioso