TAKE A BYTE OUT OF CRIME

www.UNHcFREG.com

Android Malware INvestigation Toolkit (MINT)

UNH

UNIVERSITY OF NEW HAVEN

Zhaoheng Yang

Ibrahim Baggili (PhD)

Introduction

- What is MINT?
- Pre-development
- Development Technologies
- High-Level Features
- Video Demonstration

What is MINT?

- MINT is a tool that is developed to help investigators reverse engineer Android Malware. APK files are decompiled using the open source APKTool to Smali Files. Our tool then calculates two types of danger scores for any APK file - and uses application permissions and machine learning association rules to calculate these two types of scores.

Pre-development stage

- Malware APK sample gathering
- Benign APK sample gathering
- Machine learning (Association Rule gathering)

Pre-development – APK Samples

- We received access to the Android Malware Genome Project, North Carolina State University and other malware APK samples (n=1260)
- We downloaded approximately (n=1100) benign APK samples

Pre-development – Association Rule Generation

- It was critical that we first develop association rules from benign and malicious APK files – and then integrate those rules into MINT. To do that, we decompiled (1260 malicious applications and ~1100 benign applications), and fed the extracted permissions into WEKA. From WEKA, we were then able to generate permission association rules with confidence values above 80. This helped us generate associate rules that helped us detect if the an APK file is more likely to be malicious or benign. This helped us create what we term DFA (Danger From Association Rule Score).

Development Technologies

- MINT uses the following technologies:
  - C++
  - QT 4.8 for the GUI interface
  - APKTool for decompiling the APK files
  - SQLite database backend that stores:
    - Hash values and names of malicious APK files
    - Association Rules generated from the machine learning step
    - Permission protection level for each Android permission (Used to calculate the Danger From Permission (DFP) Protection Level Score)
    - A table outlining each permission and its related system call – used to understand the permissions in the code and for calculating the DFP scores

High-Level Features

- Easy to use GUI
- Decompile APK files (using built-in APKTool)
- View code, XML, HTML, JPG, PNG and other extracted files
- Search source code
- Display each APK profile
- Compare APK files
- Generate a call graph for each APK file
- Identify known-malicious APK files using our hashed APK malware dataset
- Calculate two types of APK Danger Scores for each APK (DFA and DFP)
- Add new known malicious APK files to the hash data set
- Identify applications with known benign and malicious association rules

Video Demonstration

- To vide our video demonstration click on the link below:
  - https://www.youtube.com/watch?v=qm34MZVQCxI

UNIVERSITY OF NEW HAVEN

TAKE A BYTE OUT OF CRIME
www.UNHcFREG.com

Contact us

- Zhaoheng Yang – zhaoheng1988@gmail.com
- Ibrahim Baggili (PhD) – ibaggili@newhaven.edu

UNIVERSITY OF NEW HAVEN

TAKE A BYTE OUT OF CRIME
www.UNHcFREG.com