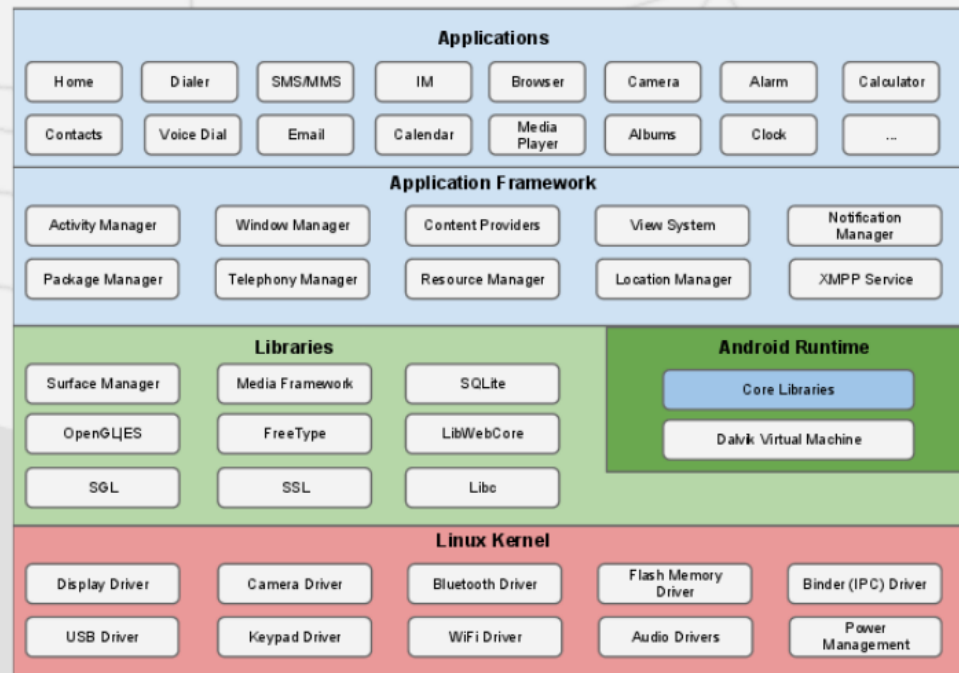


Agenda

- Android OS Overview
- Application Structure
- Static Analysis(Demo)
- Dynamic Analysis(Demo)
- Detect Intelligent Malware(Demo)



Android OS



Android Application Package (APK)

- Application Manifest
- Assets
- DEX(Dalvik executable) Code
- Libraries

Android (Pros and Cons)

- Pros:
 - Widespread Application
 - Open Source
- Cons:
 - Open Source
 - Rising number of Malware
 - No Review of Apps (Unlike Apple)



Android Analysis Techniques

- Static Analysis
- Dynamic Analysis

Static Analysis

- Reading the APK.
- Decompiling the DEX code
- Extracting the Application's detail (Manifest file)

Demonstration #1

Decompile an APK file

Demonstration #1

Decompile an APK file



Static Analysis Types

- Simple APK :
 - Source Code Visible
 - Can trace all API calls
- Obfuscated / Encrypted APK

The Need for a Reform

- Encrypted APK Activity Behavior cannot be determined
- Malware use methods to mitigate the static analysis:
 - Code Obfuscation(obfuscate patterns of malicious behaviours)
 - Class File Encryption
 - Use NDK(Native Development Kit)
 - Encryption of Resources
- Need for Dynamic Analysis

Dynamic Analysis

- Analysis of the Application behavior in real time:
 - Executes the application APK on an emulated environment
 - Detects the hidden malicious behavior in the application
 - Can help with the encrypted APKs

Demonstration #2

Static & Dynamic Analysis of an
Encrypted and Non-encrypted APK
on sandDroid



Demonstration #3

- Dynamic analysis tools are often deployed on the emulator.
- Some malware can detect that they are running on the emulator (e.g the SandDroid)
- If so, **hide it's malicious behaviors.**

Improvement Strategies

- Harden the analysis environment(Add changes)
- Trick the malware to believe that an emulator is a real device
- Once let down the guard, the malware will execute its malicious code.

#1 Changes to kernel

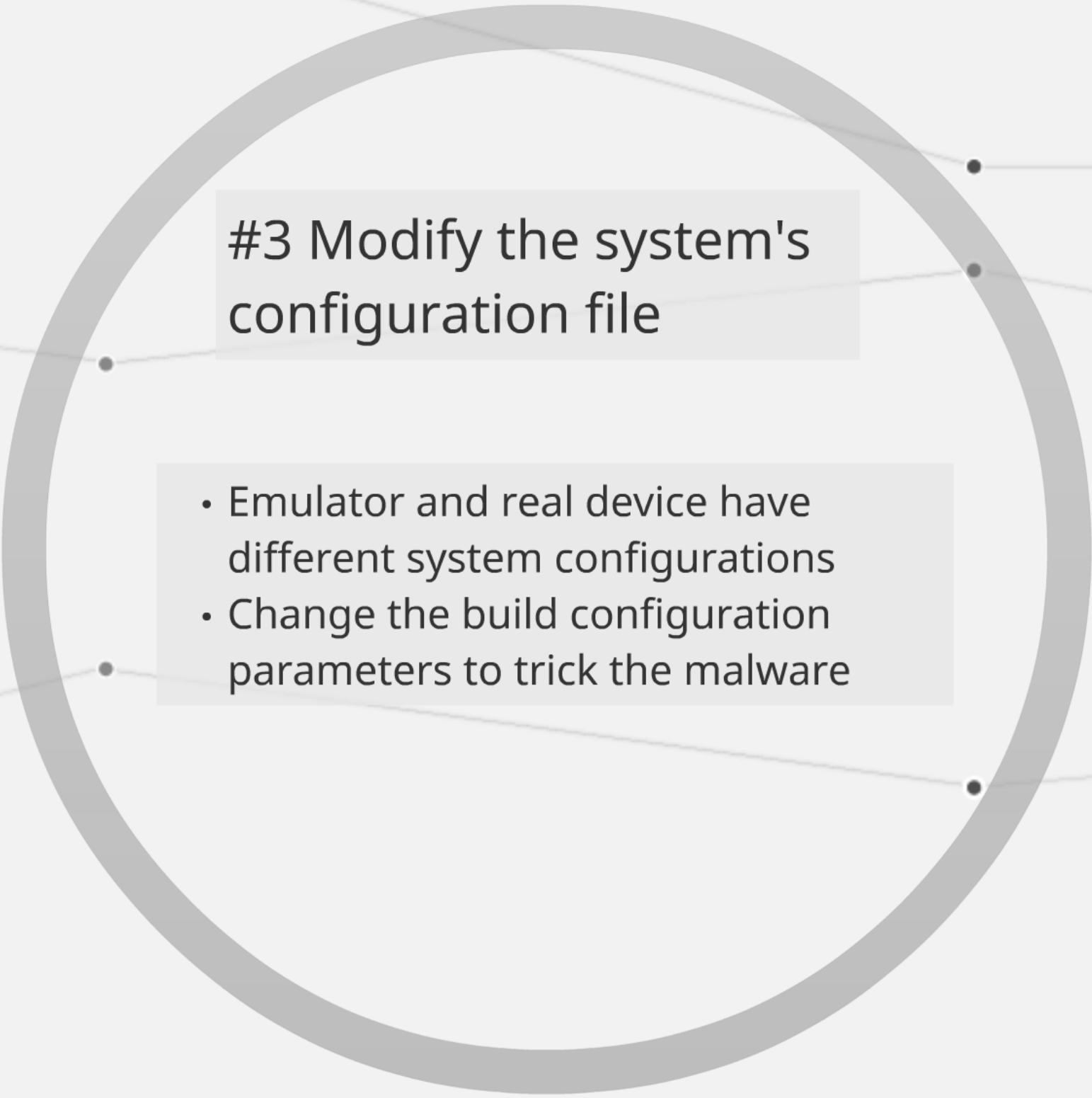
- Different file contents on emulator and real device
- System call behaves differently in emulator and real device.
- Make change to system call.
- Trick the malware to believe that the files only exist on the real device are on the emulator

system's
e

vice have
figurations
figuration
ne malware

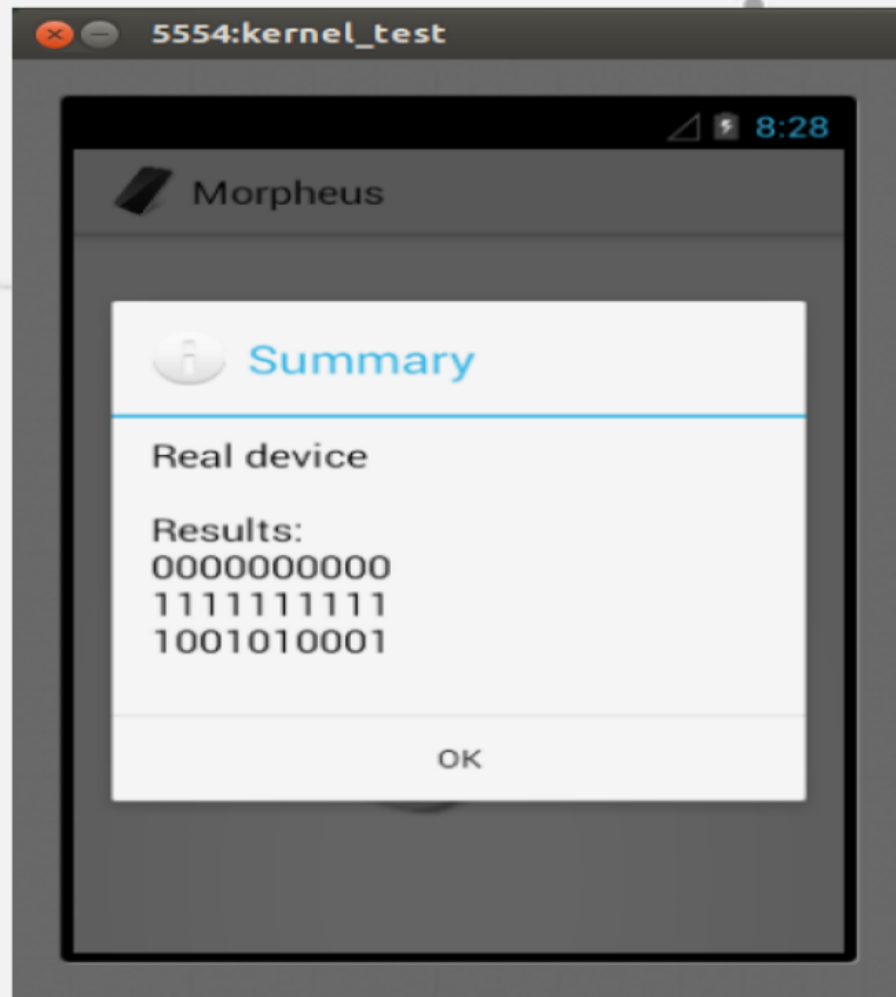
#2 Emulate user input

- Emulator doesn't have real user interactions (e.g touch button)
- Some malware only execute malicious code when there are user inputs.
- Emulate user input the trick the malware.



#3 Modify the system's configuration file

- Emulator and real device have different system configurations
- Change the build configuration parameters to trick the malware





Thank you !

Resources:

- http://www.ibm.com/support/knowledgecenter/SSHS8R_7.1.0/com.ibm.worklight.deploy.doc/admin/c_pg_obfus_intro.html
- <http://riis.com/blog/android-obfuscation/>
- <http://mobilenext.net/protect-android-app-app-pirates/>

Detecting Intelligent Malware on Dynamic Android Analysis Environments

