Android Hacker's handbook

https://r-ever-scp.tistory.com/

91913622-송태현

Android Hacker's handbook

warning

※ 이 내용은 영어 내용을 PAPAGO와 고등학교 수준의 영어실력으로 번역하며 공부 한 내용이므로 완전 정확하지는 않음

0. BOOK

>

1. Android System Architecture

>

2. Security Boundaries and Enforcement

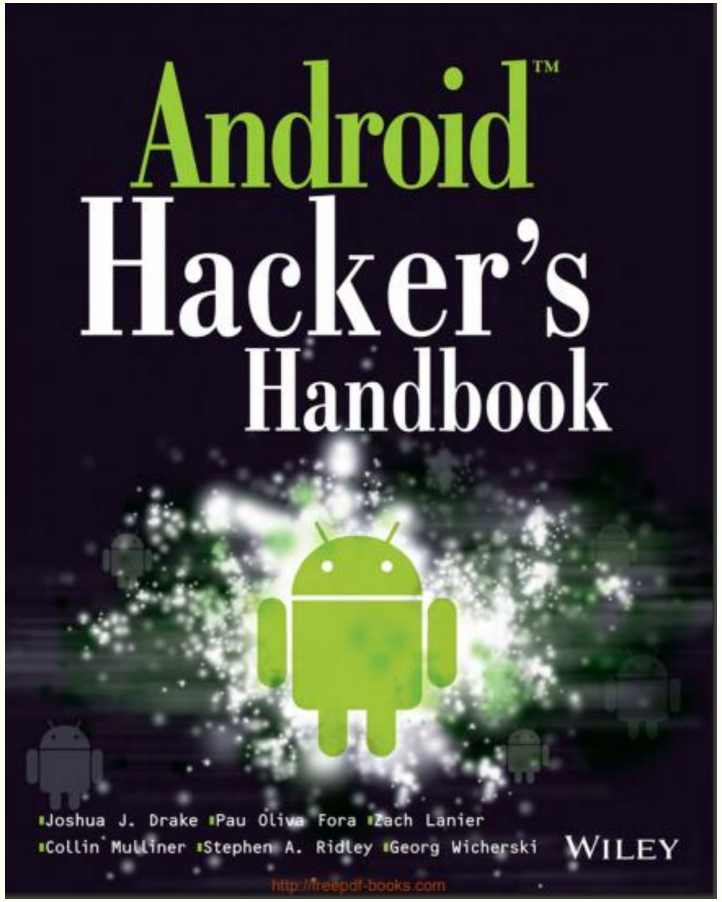
>

3. Looking Closer at the Layers

To Be Continued

Android Hacker's handbook

0. BOOK



Android™
Hacker's
Handbook
Joshua J. Drake Pau Oliva Fora Zach Lanier
Collin Mulliner Stephen A. Ridley Georg Wicherski
WILEY
http://freepdf-books.com

Name : Android Hacker's Handbook

상영이형

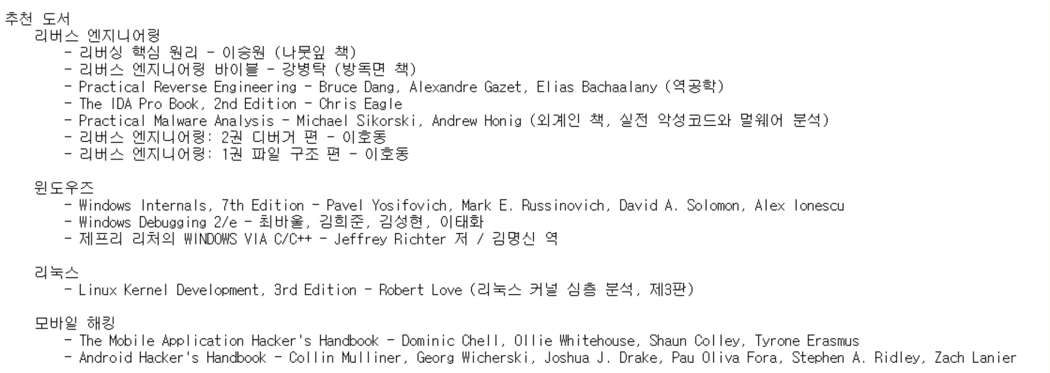
https://py0zz1.tistory.com/117?category=841836

Android Hacker's Handbook [번역]

py0zz1.tistory.com

이거 읽자 태현아

오후 11:56



추천 도서

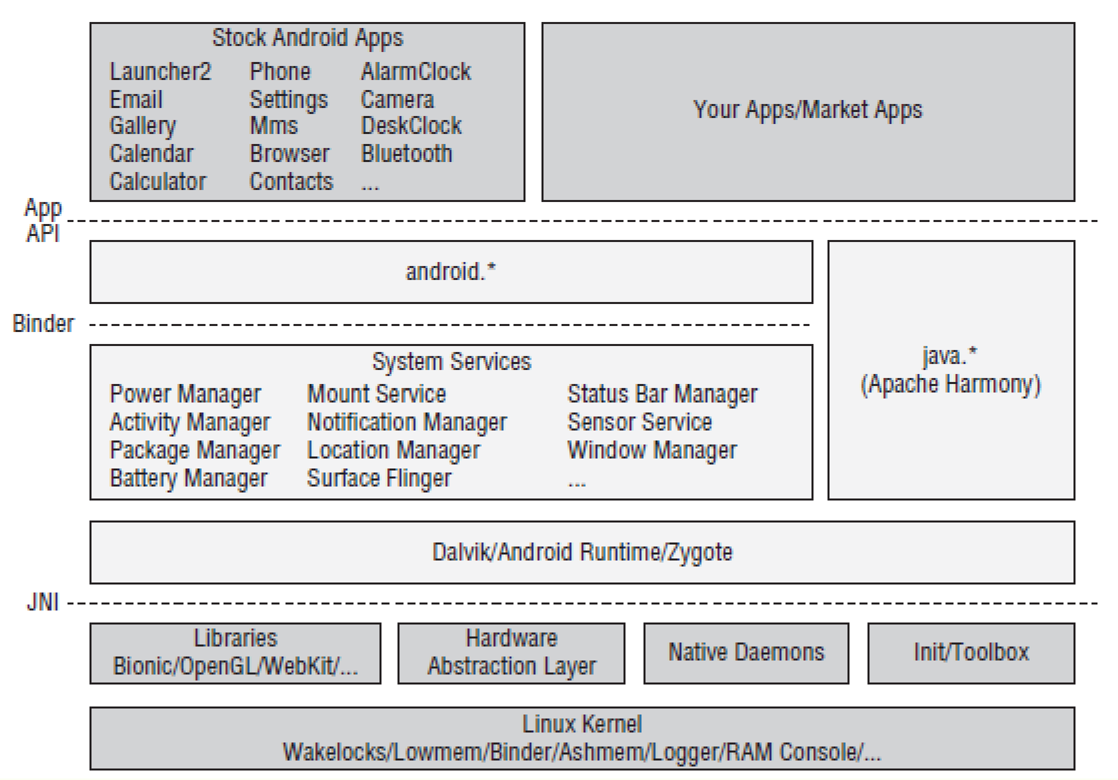
리버스 엔지니어링
- 리버싱 핵심 원리 - 이승원 (나뭇잎 책)
- 리버스 엔지니어링 바이블 - 강병탁 (방독면 책)
- Practical Reverse Engineering - Bruce Dang, Alexandre Gazet, Elias Bachaalany (역공학)
- The IDA Pro Book, 2nd Edition - Chris Eagle
- Practical Malware Analysis - Michael Sikorski, Andrew Honig (외계인 책, 실전 악성코드와 멀웨어 분석)
- 리버스 엔지니어링: 2권 디버거 편 - 이호동
- 리버스 엔지니어링: 1권 파일 구조 편 - 이호동

윈도우즈
- Windows Internals, 7th Edition - Pavel Yosifovich, Mark E. Russinovich, David A. Solomon, Alex Ionescu
- Windows Debugging 2/e - 최바울, 김희준, 김성현, 이태화
- 제프리 리처의 WINDOWS VIA C/C++ - Jeffrey Richter 저 / 김명신 역

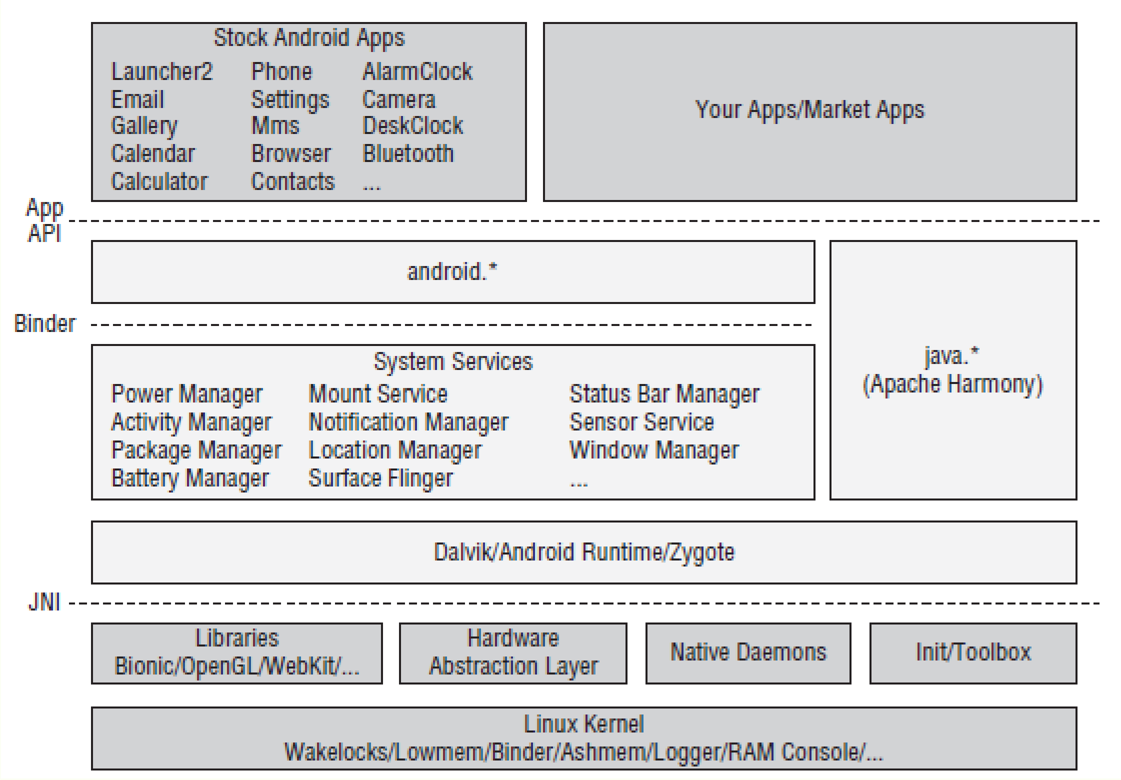
리눅스
- Linux Kernel Development, 3rd Edition - Robert Love (리눅스 커널 심층 분석, 제3판)

모바일 해킹
- The Mobile Application Hacker's Handbook - Dominic Chell, Ollie Whitehouse, Shaun Colley, Tyrone Erasmus
- Android Hacker's Handbook - Collin Mulliner, Georg Wicherski, Joshua J. Drake, Pau Oliva Fora, Stephen A. Ridley, Zach Lanier

1. Android System Architecture



1. Android System Architecture



Stock Android Apps

Launcher2 Email Gallery Calendar Calculator

Phone Settings Mms Browser Contacts

AlarmClock Camera DeskClock Bluetooth ...

Your Apps/Market Apps

App API

android.*

java.*
(Apache Harmony)

Binder

System Services

Power Manager Activity Manager Package Manager Battery Manager

Mount Service Notification Manager Location Manager Surface Flinger

Status Bar Manager Sensor Service Window Manager ...

Dalvik/Android Runtime/Zygote

JNI

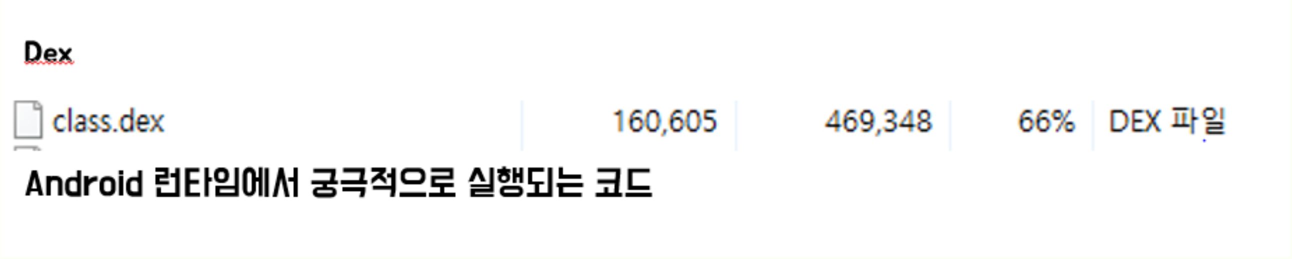
Libraries
Bionic/OpenGL/WebKit/...

Hardware
Abstraction Layer

Native Daemons

Init/Toolbox

Linux Kernel
Wakelocks/Lowmem/Binder/Ashmem/Logger/RAM Console/...

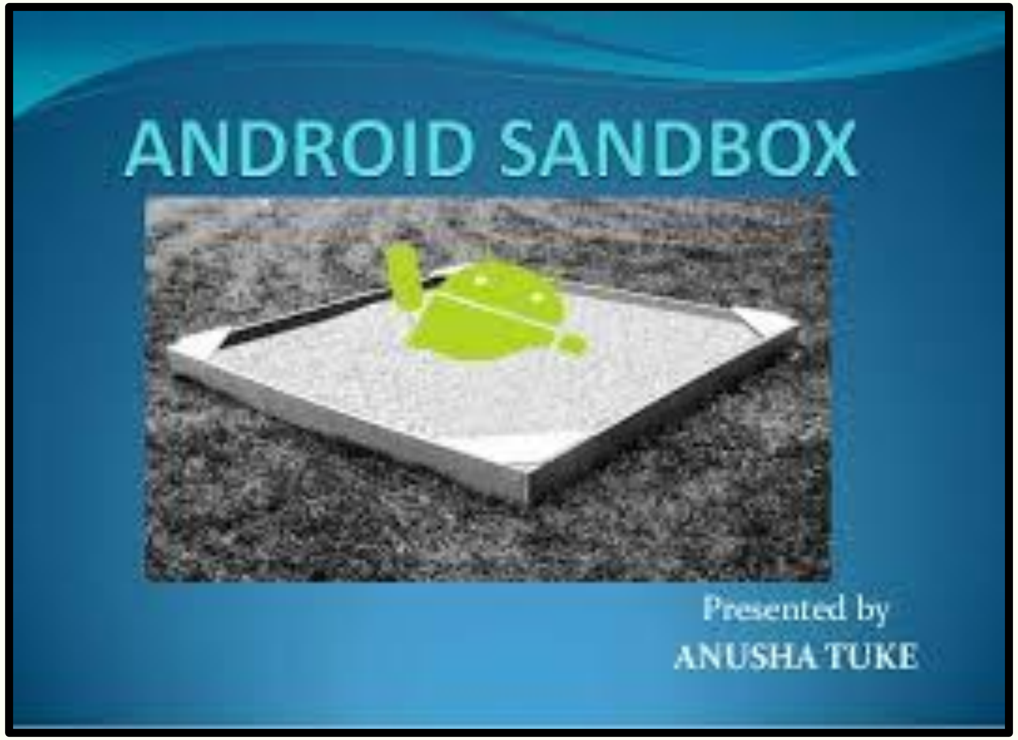


Dex

class.dex	160,605	469,348	66%	DEX 파일

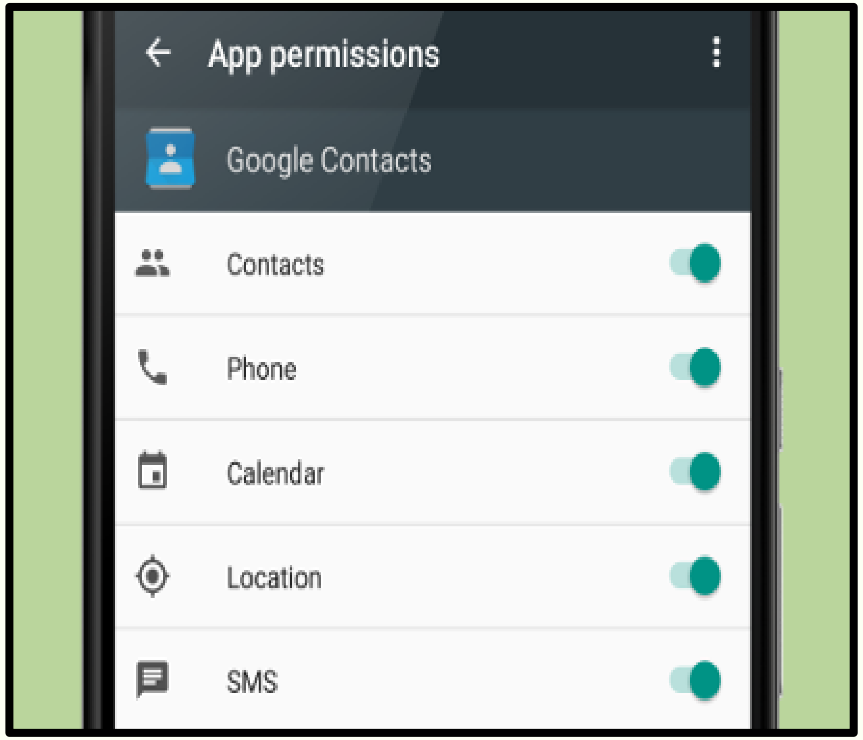
Android 런타임에서 궁극적으로 실행되는 코드

2. Security Boundaries and Enforcement



ANDROID SANDBOX

Presented by
ANUSHA TUKE

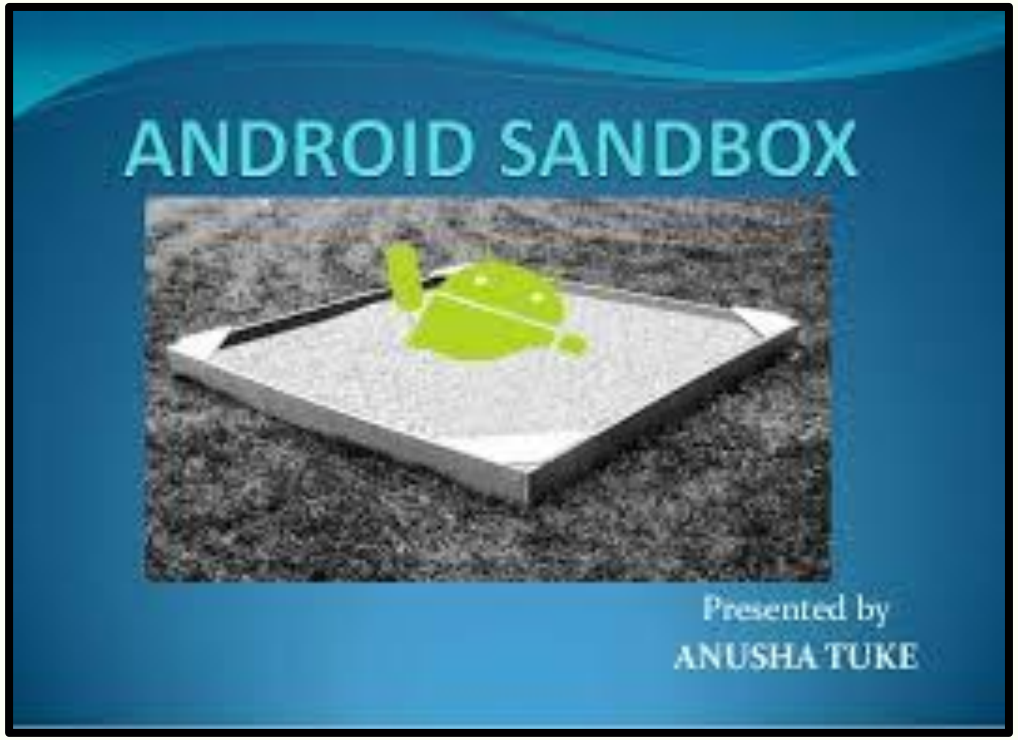


App permissions

Google Contacts

Contacts	●
Phone	●
Calendar	●
Location	●
SMS	●

2. Security Boundaries and Enforcement

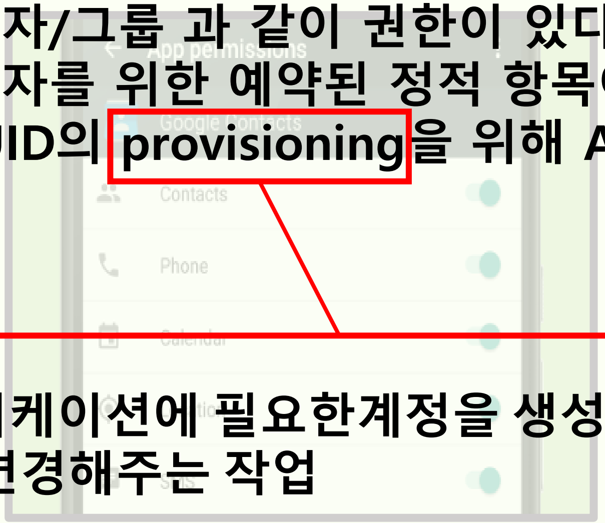


ANDROID SANDBOX

Presented by
ANUSHA TUKE

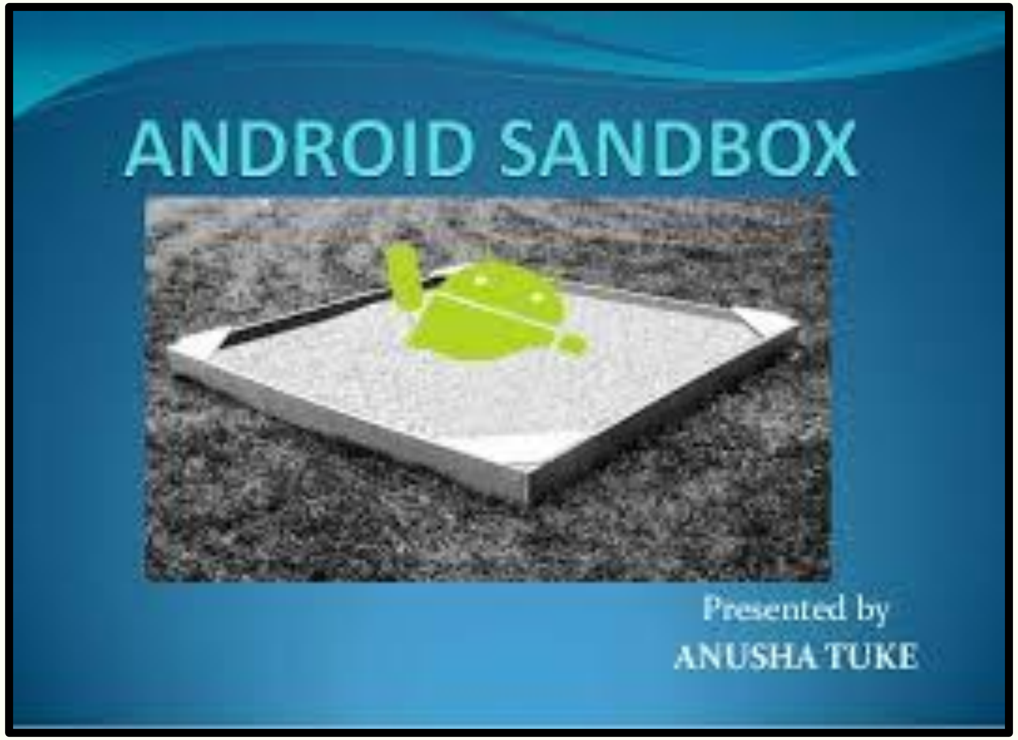
Android ID(AID)

- 사용자/그룹 과 같이 권한이 있다
- 사용자를 위한 예약된 정적 항목이 있다.
- 앱 UID의 provisioning을 위해 AID 범위를 예약



어플리케이션에 필요한계정을 생성하거나 접근권한을 변경해주는 작업

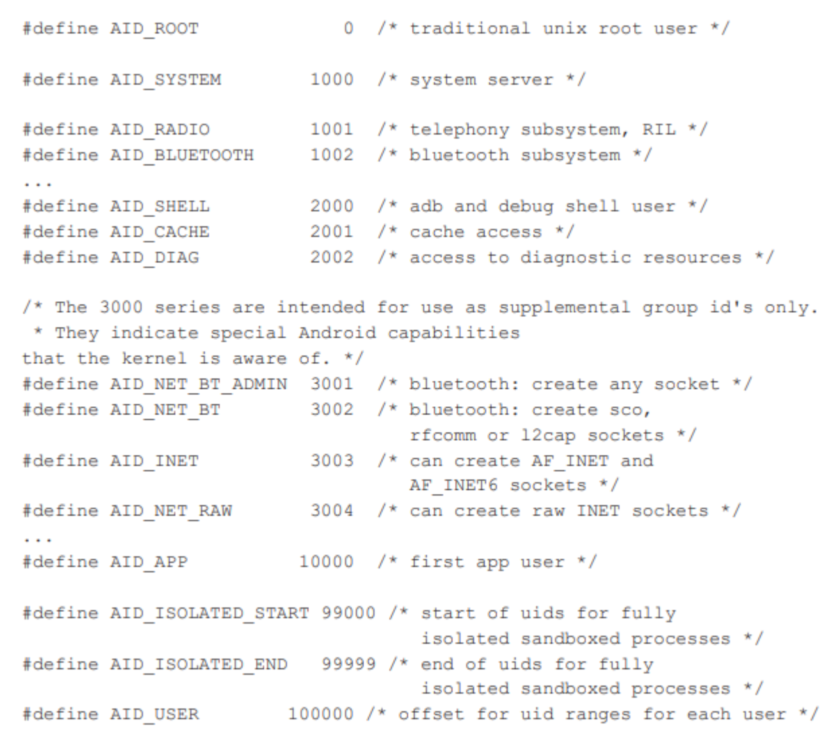
2. Security Boundaries and Enforcement



ANDROID SANDBOX

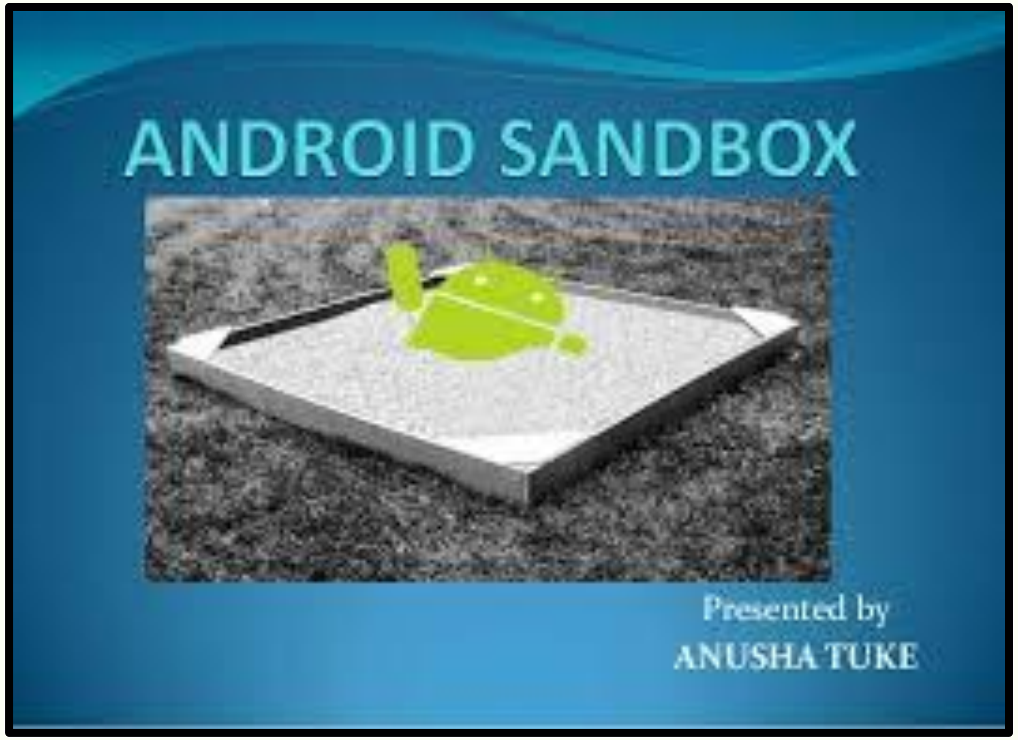
Presented by
ANUSHA TUKE

symstem/core/include/private/android_filesystem_config.h



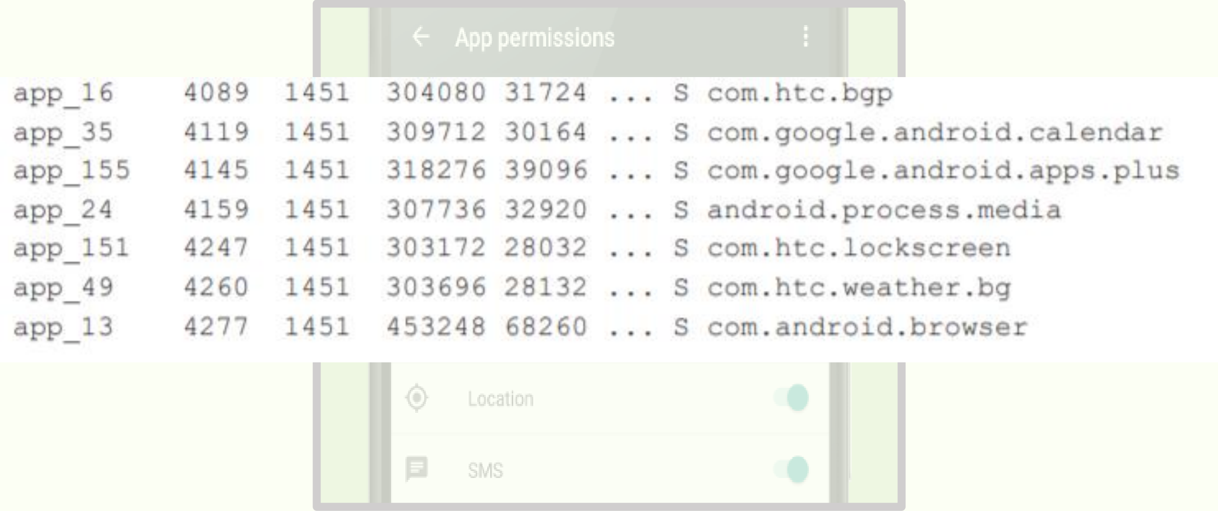
```
#define AID_ROOT             0  /* traditional unix root user */

#define AID_SYSTEM        1000  /* system server */

#define AID_RADIO         1001  /* telephony subsystem, RIL */
#define AID_BLUETOOTH     1002  /* bluetooth subsystem */
...
#define AID_SHELL         2000  /* adb and debug shell user */
#define AID_CACHE         2001  /* cache access */
#define AID_DIAG          2002  /* access to diagnostic resources */

/* The 3000 series are intended for use as supplemental group id's only.
 * They indicate special Android capabilities
that the kernel is aware of. */
#define AID_NET_BT_ADMIN  3001  /* bluetooth: create any socket */
#define AID_NET_BT        3002  /* bluetooth: create sco,
                                   rfcomm or l2cap sockets */
#define AID_INET          3003  /* can create AF_INET and
                                   AF_INET6 sockets */
#define AID_NET_RAW       3004  /* can create raw INET sockets */
...
#define AID_APP          10000  /* first app user */

#define AID_ISOLATED_START 99000 /* start of uids for fully
                                    isolated sandboxed processes */
#define AID_ISOLATED_END   99999 /* end of uids for fully
                                    isolated sandboxed processes */
#define AID_USER        100000 /* offset for uid ranges for each user */
```

2. Security Boundaries and Enforcement



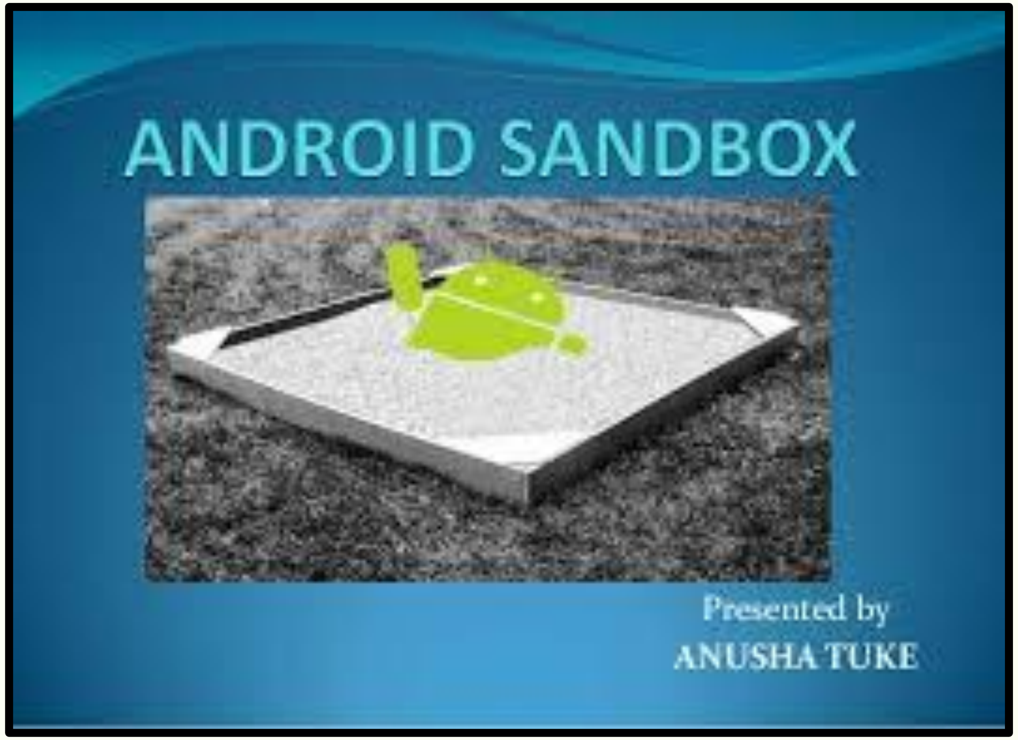
ANDROID SANDBOX

Presented by
ANUSHA TUKE



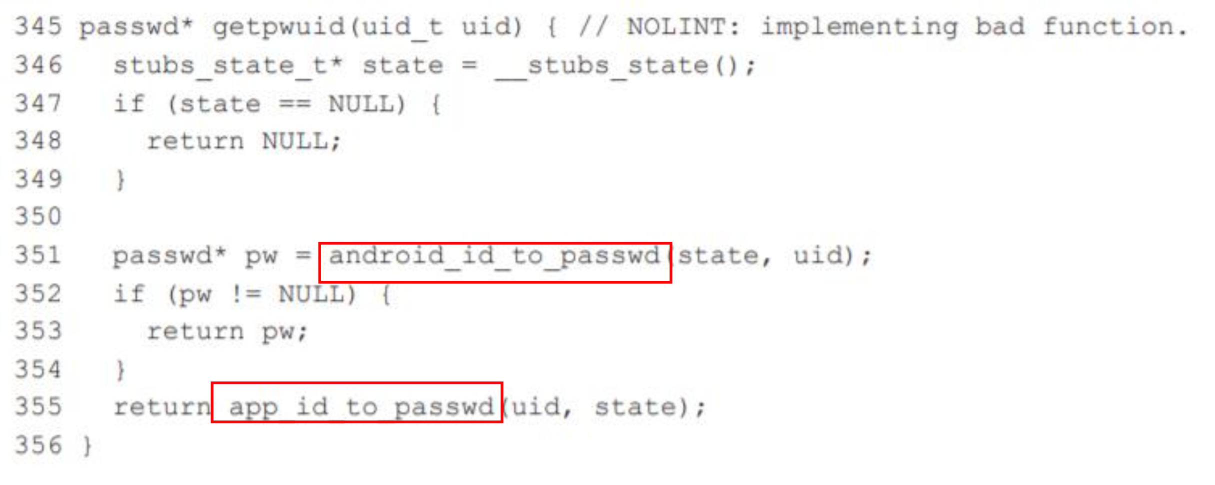
```
app_16     4089  1451  304080 31724 ... S com.htc.bgp
app_35     4119  1451  309712 30164 ... S com.google.android.calendar
app_155    4145  1451  318276 39096 ... S com.google.android.apps.plus
app_24     4159  1451  307736 32920 ... S android.process.media
app_151    4247  1451  303172 28032 ... S com.htc.lockscreen
app_49     4260  1451  303696 28132 ... S com.htc.weather.bg
app_13     4277  1451  453248 68260 ... S com.android.browser
```

2. Security Boundaries and Enforcement

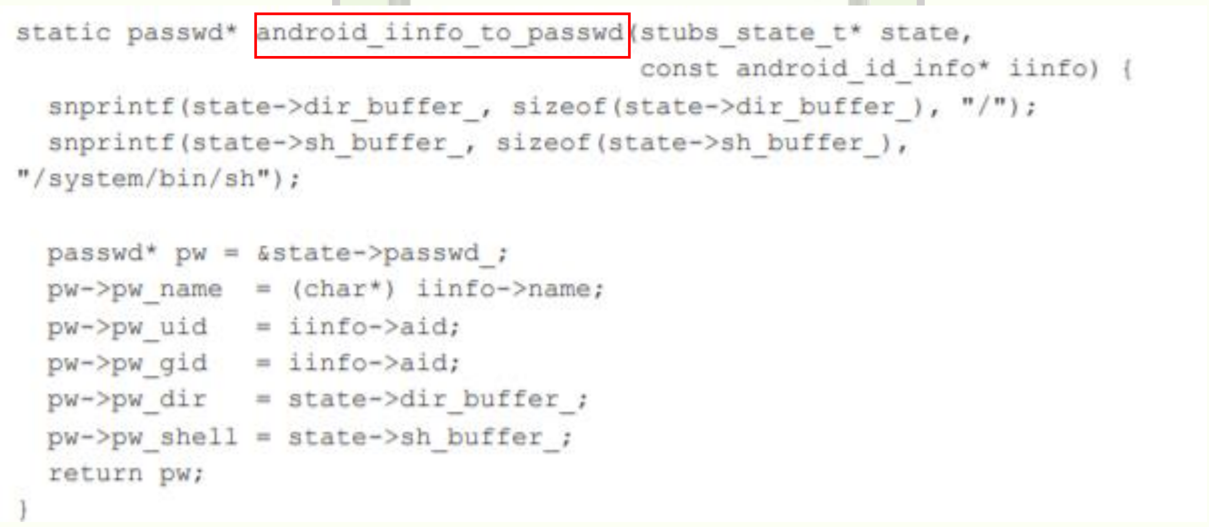


ANDROID SANDBOX

Presented by
ANUSHA TUKE

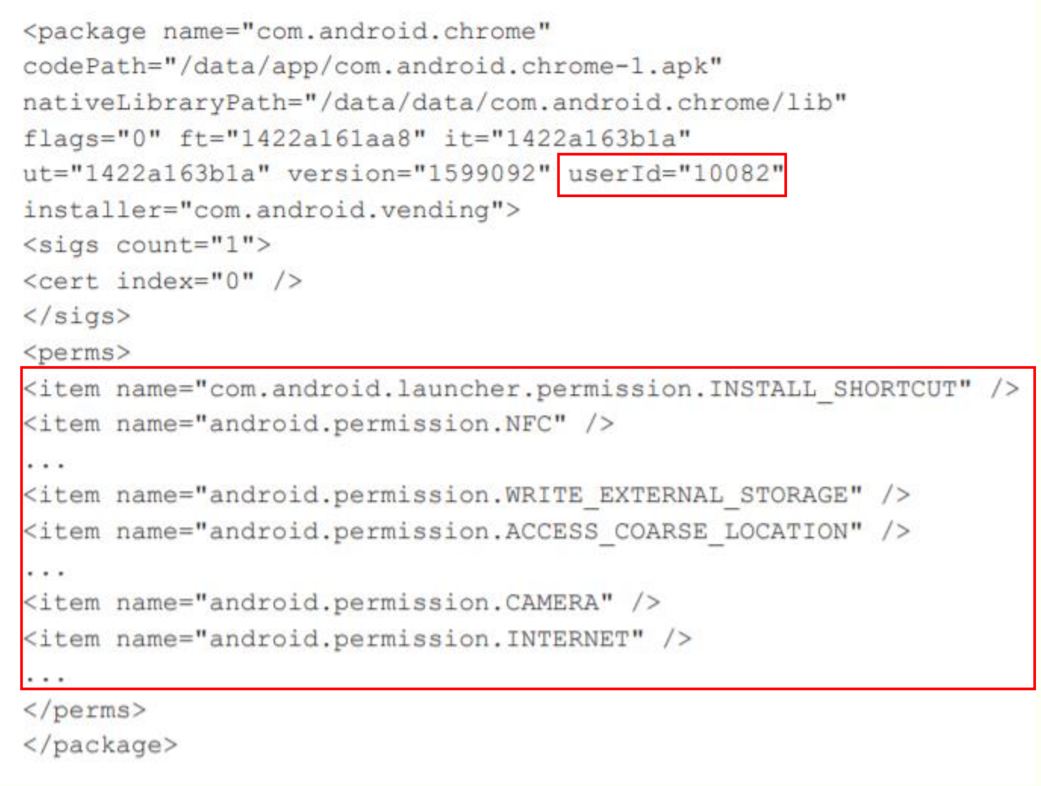


```
passwd* getpwuid(uid_t uid) { // NOLINT: implementing bad function.
  stubs_state_t* state = __stubs_state();
  if (state == NULL) {
    return NULL;
  }

  passwd* pw = android_id_to_passwd(state, uid);
  if (pw != NULL) {
    return pw;
  }
  return app_id_to_passwd(uid, state);
}
```

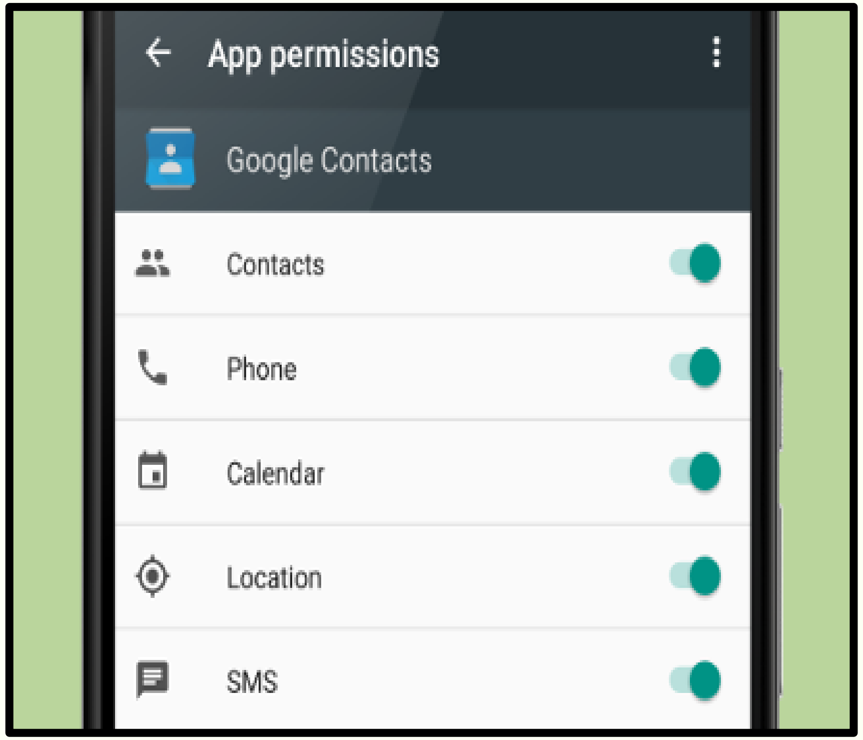


```
static passwd* android_iinfo_to_passwd(stubs_state_t* state,
                                       const android_id_info* iinfo) {
  snprintf(state->dir_buffer_, sizeof(state->dir_buffer_), "/");
  snprintf(state->sh_buffer_, sizeof(state->sh_buffer_),
"/system/bin/sh");

  passwd* pw = &state->passwd_;
  pw->pw_name  = (char*) iinfo->name;
  pw->pw_uid   = iinfo->aid;
  pw->pw_gid   = iinfo->aid;
  pw->pw_dir   = state->dir_buffer_;
  pw->pw_shell = state->sh_buffer_;
  return pw;
}
```

2. Security Boundaries and Enforcement



```
<package name="com.android.chrome"
codePath="/data/app/com.android.chrome-1.apk"
nativeLibraryPath="/data/data/com.android.chrome/lib"
flags="0" ft="1422a161aa8" it="1422a163b1a"
ut="1422a163b1a" version="1599092" userId="10082"
installer="com.android.vending">
<sigs count="1">
<cert index="0" />
</sigs>
<perms>
<item name="com.android.launcher.permission.INSTALL_SHORTCUT" />
<item name="android.permission.NFC" />
...
<item name="android.permission.WRITE_EXTERNAL_STORAGE" />
<item name="android.permission.ACCESS_COARSE_LOCATION" />
...
<item name="android.permission.CAMERA" />
<item name="android.permission.INTERNET" />
...
</perms>
</package>
```



App permissions

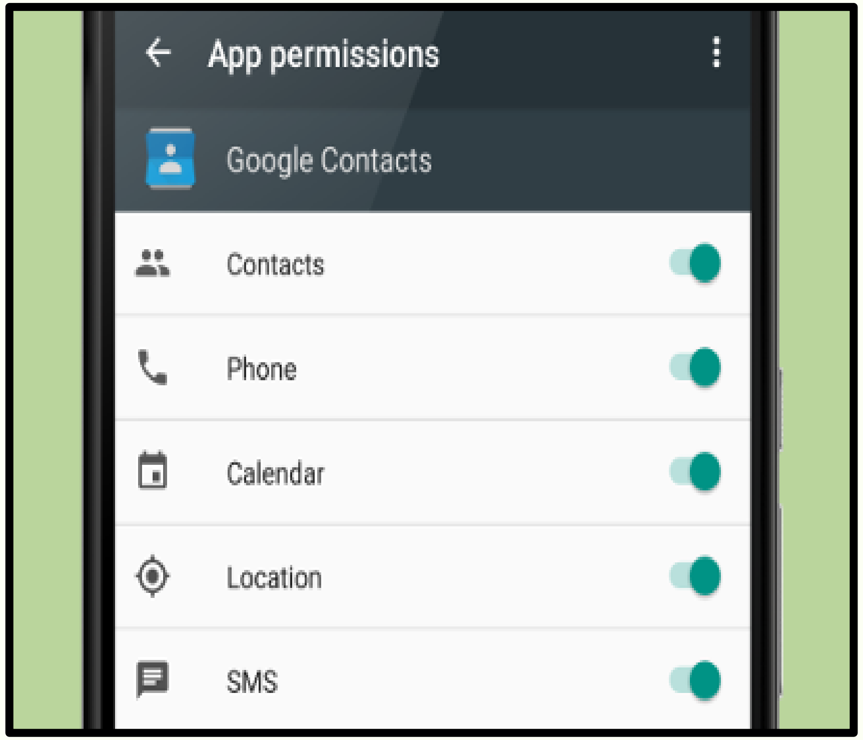
Google Contacts

- Contacts ●
- Phone ●
- Calendar ●
- Location ●
- SMS ●

/data/system/packages.xml

Android Hacker's handbook

2. Security Boundaries and Enforcement


App permissions

Google Contacts

Contacts ●

Phone ●

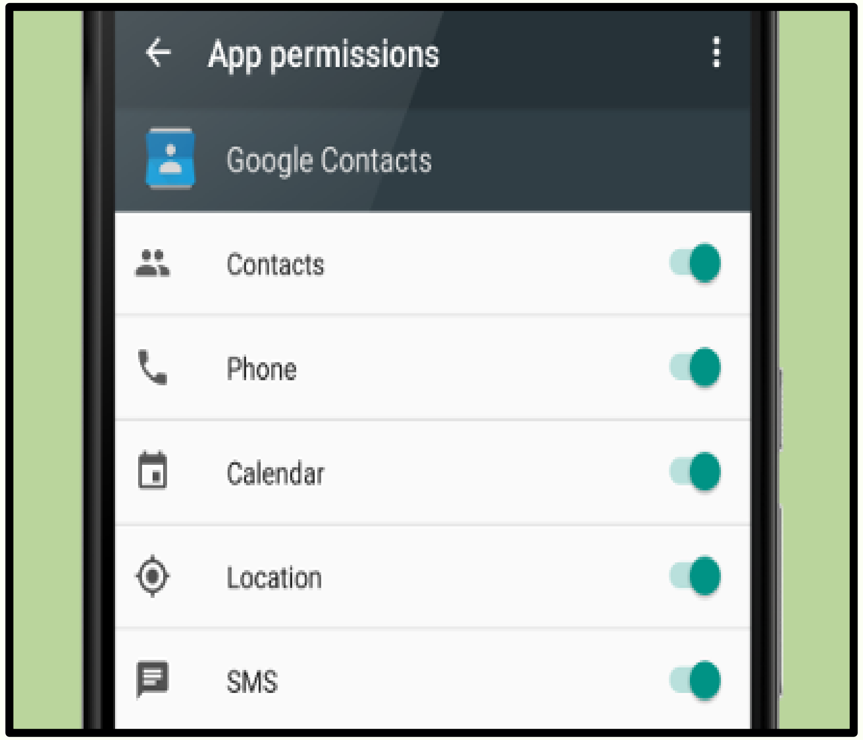
Calendar ●

Location ●

SMS ●

Android Hacker's handbook

2. Security Boundaries and Enforcement

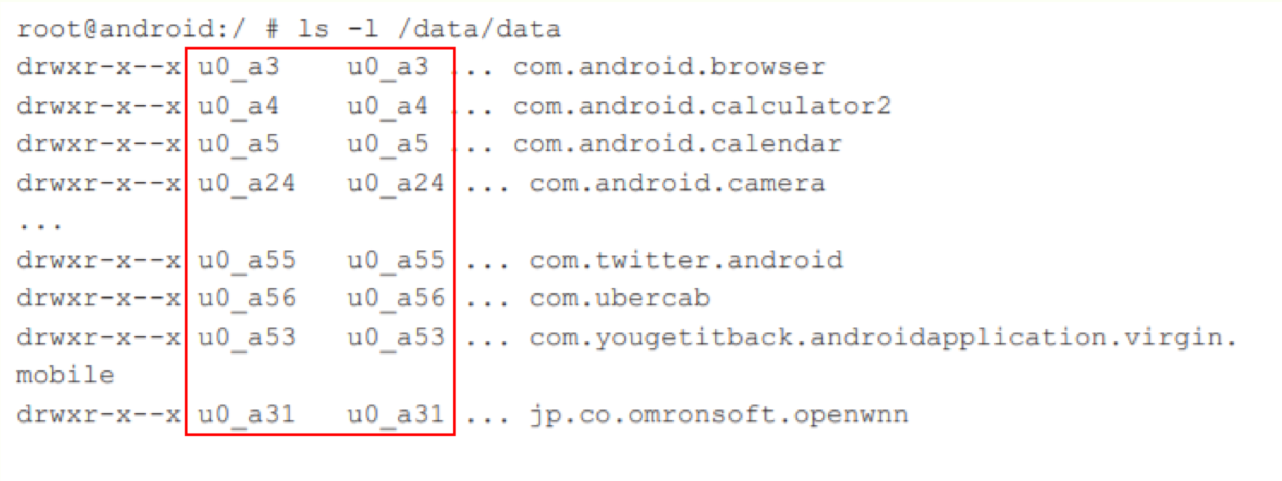


App permissions

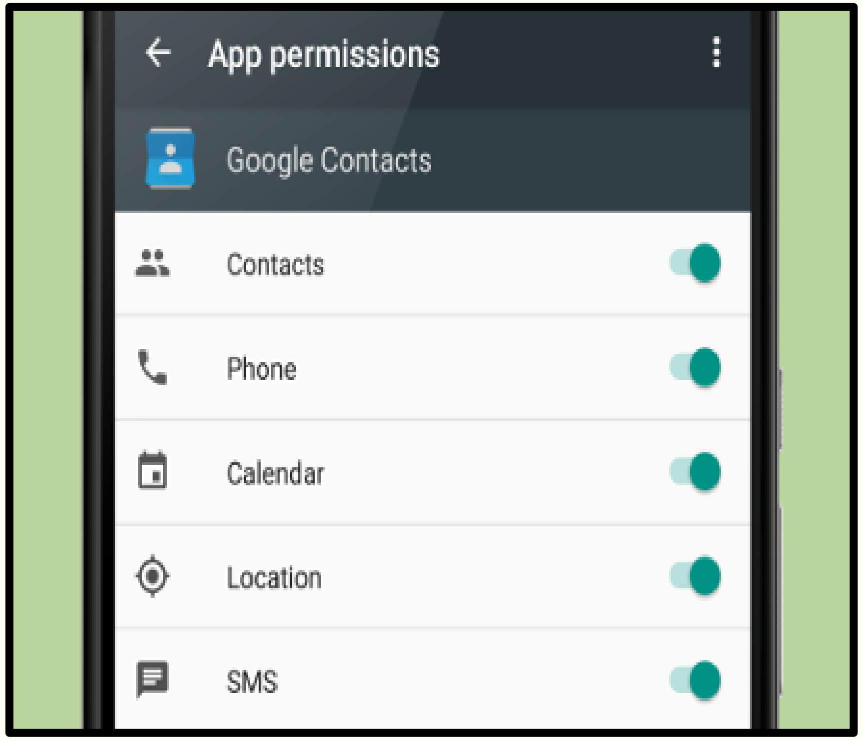
Google Contacts

- Contacts ●
- Phone ●
- Calendar ●
- Location ●
- SMS ●

2. Security Boundaries and Enforcement



```
root@android:/ # ls -l /data/data
drwxr-x--x u0_a3     u0_a3   ... com.android.browser
drwxr-x--x u0_a4     u0_a4   ... com.android.calculator2
drwxr-x--x u0_a5     u0_a5   ... com.android.calendar
drwxr-x--x u0_a24    u0_a24  ... com.android.camera
...
drwxr-x--x u0_a55    u0_a55  ... com.twitter.android
drwxr-x--x u0_a56    u0_a56  ... com.ubercab
drwxr-x--x u0_a53    u0_a53  ... com.yougetitback.androidapplication.virgin.
mobile
drwxr-x--x u0_a31    u0_a31  ... jp.co.omronsoft.openwnn
```



App permissions

Google Contacts

Contacts ●

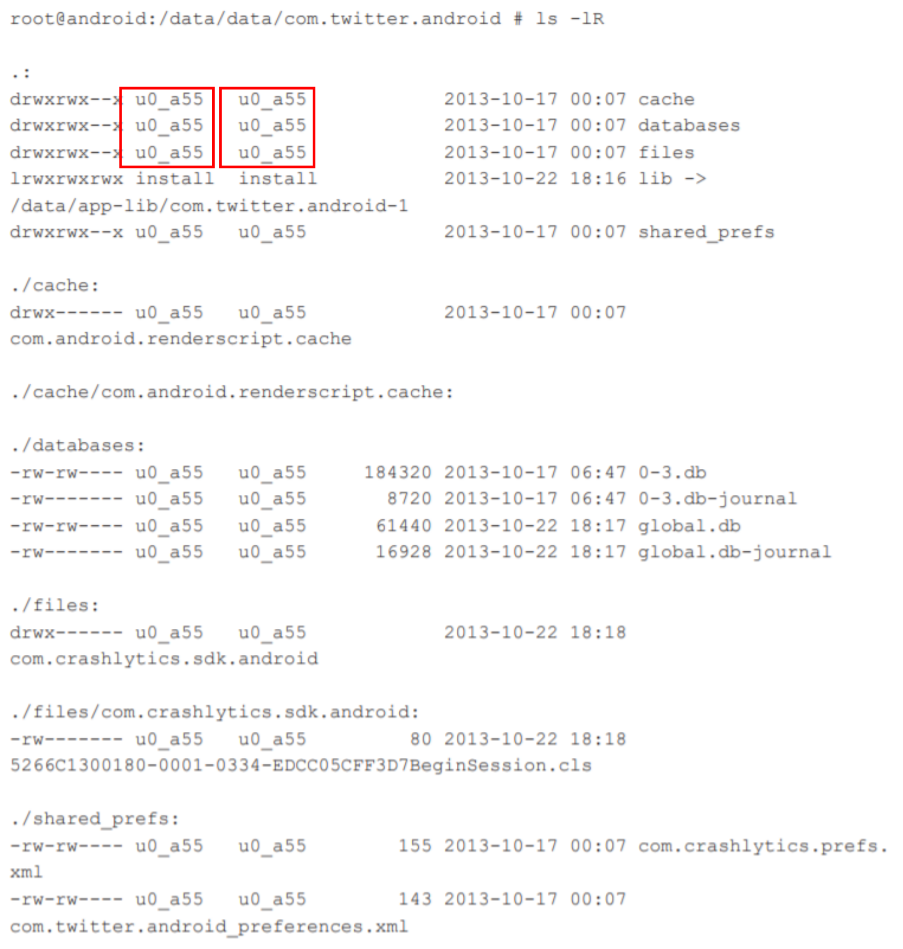
Phone ●

Calendar ●

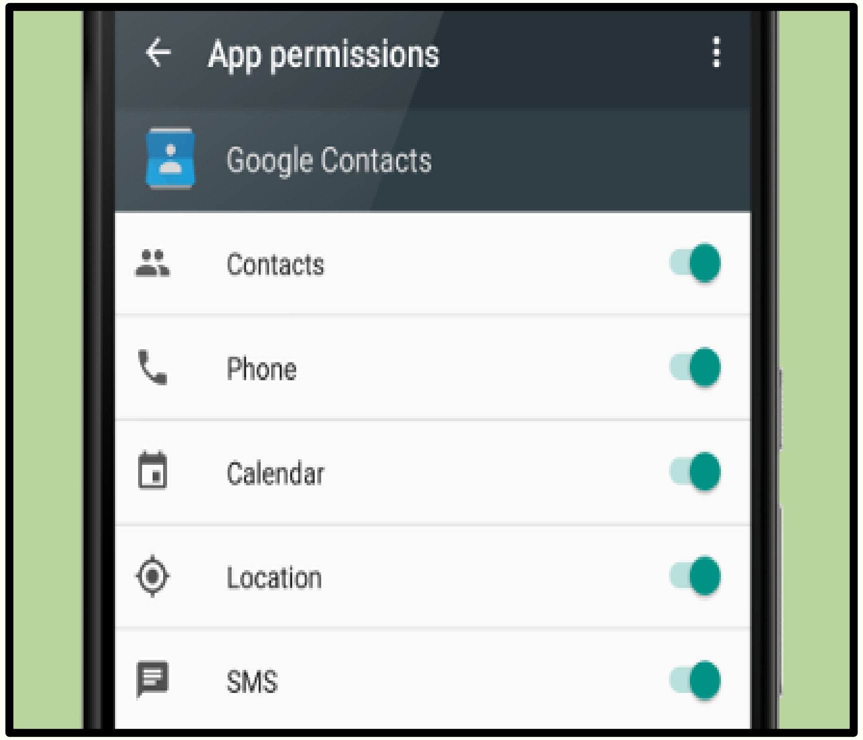
Location ●

SMS ●

2. Security Boundaries and Enforcement



```
root@android:/data/data/com.twitter.android # ls -lR

.:
drwxrwx--x u0_a55   u0_a55            2013-10-17 00:07 cache
drwxrwx--x u0_a55   u0_a55            2013-10-17 00:07 databases
drwxrwx--x u0_a55   u0_a55            2013-10-17 00:07 files
lrwxrwxrwx install  install           2013-10-22 18:16 lib ->
/data/app-lib/com.twitter.android-1
drwxrwx--x u0_a55   u0_a55            2013-10-17 00:07 shared_prefs

./cache:
drwx------ u0_a55   u0_a55            2013-10-17 00:07
com.android.renderscript.cache

./cache/com.android.renderscript.cache:

./databases:
-rw-rw---- u0_a55   u0_a55     184320 2013-10-17 06:47 0-3.db
-rw------- u0_a55   u0_a55       8720 2013-10-17 06:47 0-3.db-journal
-rw-rw---- u0_a55   u0_a55      61440 2013-10-22 18:17 global.db
-rw------- u0_a55   u0_a55      16928 2013-10-22 18:17 global.db-journal

./files:
drwx------ u0_a55   u0_a55            2013-10-22 18:18
com.crashlytics.sdk.android

./files/com.crashlytics.sdk.android:
-rw------- u0_a55   u0_a55         80 2013-10-22 18:18
5266C1300180-0001-0334-EDCC05CFF3D7BeginSession.cls

./shared_prefs:
-rw-rw---- u0_a55   u0_a55        155 2013-10-17 00:07 com.crashlytics.prefs.
xml
-rw-rw---- u0_a55   u0_a55        143 2013-10-17 00:07
com.twitter.android_preferences.xml
```



App permissions

Google Contacts

Contacts ●

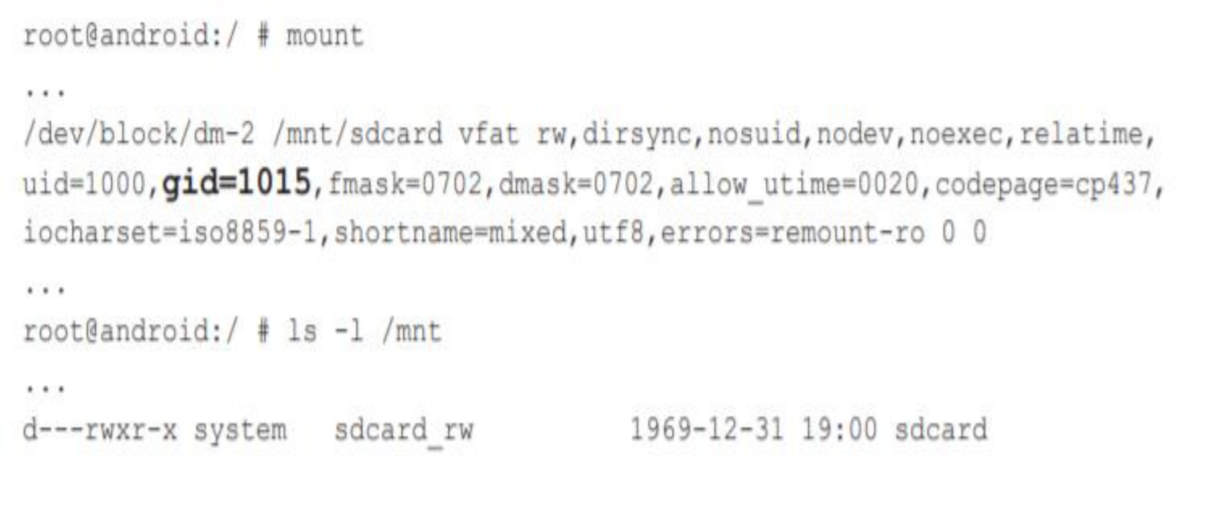
Phone ●

Calendar ●

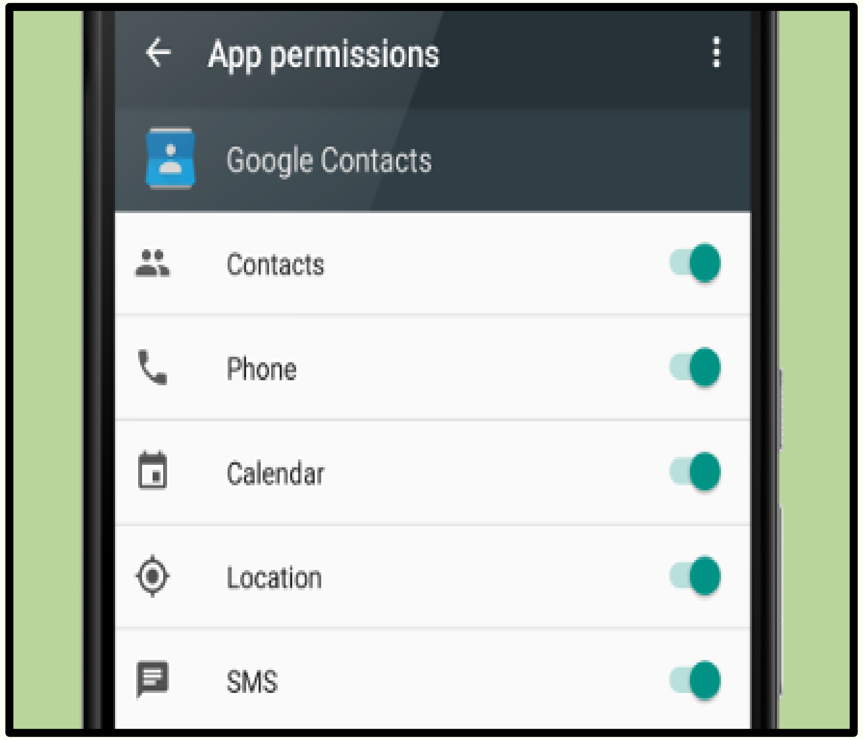
Location ●

SMS ●

2. Security Boundaries and Enforcement



```
root@android:/ # mount
...
/dev/block/dm-2 /mnt/sdcard vfat rw,dirsync,nosuid,nodev,noexec,relatime,
uid=1000,gid=1015,fmask=0702,dmask=0702,allow_utime=0020,codepage=cp437,
iocharset=iso8859-1,shortname=mixed,utf8,errors=remount-ro 0 0
...
root@android:/ # ls -l /mnt
...
d---rwxr-x system   sdcard_rw          1969-12-31 19:00 sdcard
```



App permissions

Google Contacts

Contacts ●

Phone ●

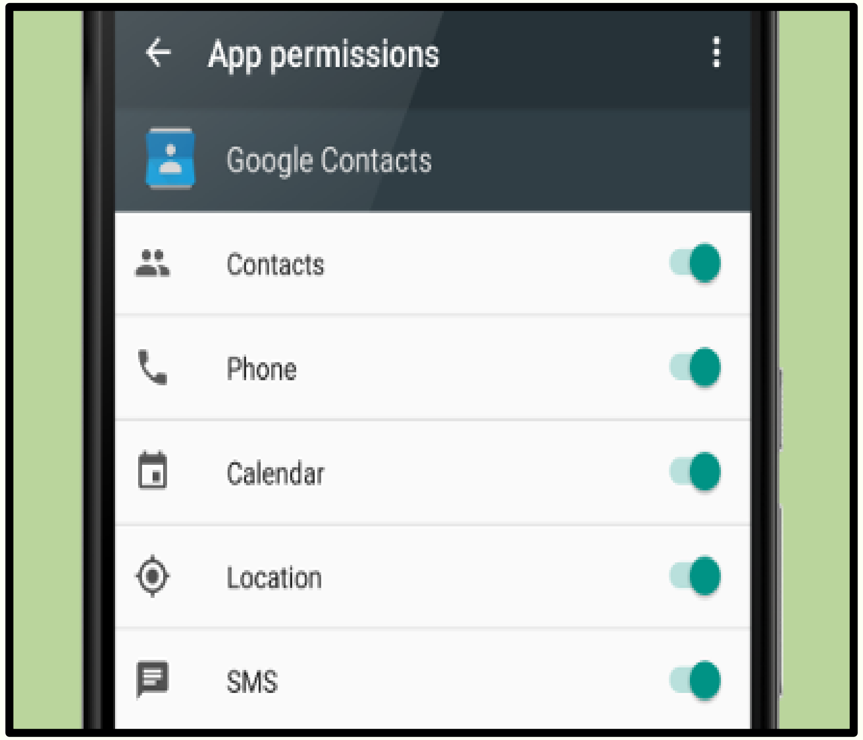
Calendar ●

Location ●

SMS ●

Android Hacker's handbook

2. Security Boundaries and Enforcement



App permissions

Google Contacts

Contacts ●

Phone ●

Calendar ●

Location ●

SMS ●

3. Looking Closer at the Layers