RSAConference2015
Singapore | 22-24 July | Marina Bay Sands

CHANGE
Challenge today's security thinking

SESSION ID: SPO-R09

Inside of APK Protectors

Bob Pan

Mobile Security Expert
Alibaba
pxb1988@gmail.com

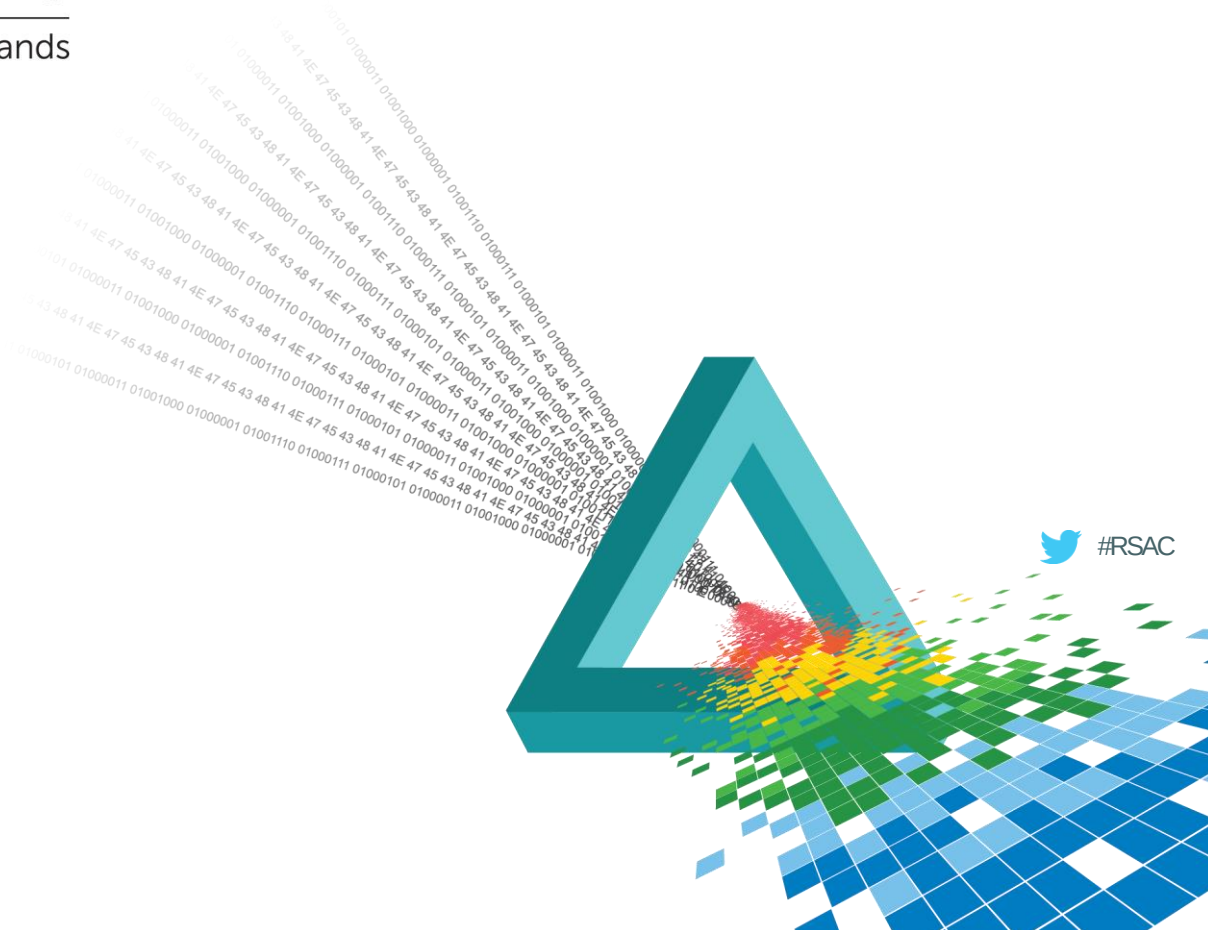
Agenda

- What is APK Protector
- Tools to reverse engineer an APK
  - Anti-tools by APK Protector
- The hide and seek game between APK Protector and hacker
- Q & A

#define APK_PROTECTOR

- Post-Developing
- Easy to apply
  - Everyone can use
- Hard to break

Anti-Tools



dex2jar

- Convert back dex to jar
- Decompile with existing decompiler
  E.x. Jd-gui

Hello.class

```
1  package p.t;
2
3  import android.os.Bundle;
4  import android.app.Activity;
5
6  public class Hello extends Activity
7  {
8      public Hello() {
9          super();
10     }
11
12     public void onCreate(final Bundle bundle) {
13         super.onCreate(bundle);
14         this.setContentView(2130903040);
15     }
16 }
```

smali/baksmali

- Disassemble dex file
- Easy to modify dex

```
# virtual methods
.method public onCreate(Landroid/os/Bundle;)V
    .registers 3
    .param p1, "savedInstanceState"    # Landroid/os/Bundle;

    .prologue
    .line 12
    invoke-super {p0, p1}, Landroid/app/Activity;->onCreate(Landroid/os

    .line 13
    const/high16 v0, 0x7f030000

    invoke-virtual {p0, v0}, Lp/t/Hello;->setContentView(I)V

    .line 14
    return-void
.end method
```

apktool

- Extract resources

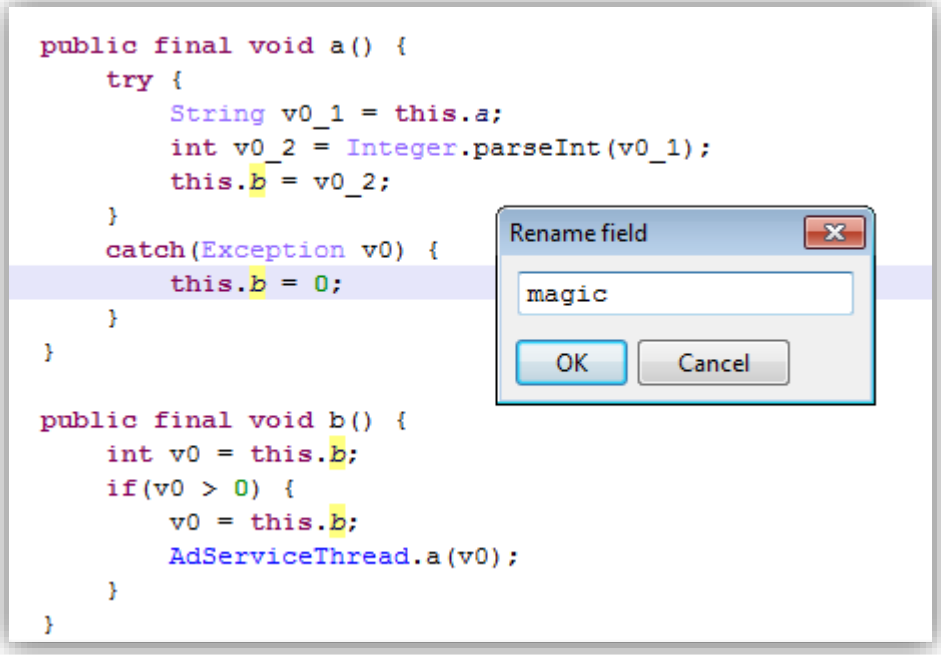
```
[bob@bob-x1 bin]$ apktool d Hello-release-unsigned.apk
I: Using Apktool 2.0.0-35f978-SNAPSHOT on Hello-release-unsigned.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /home/bob/apktool/framework/1.apk
I: Regular manifest package...
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...
```

- Disassemble dex
  - Smali based

```
[bob@bob-x1 bin]$ apktool b Hello-release-unsigned
I: Using Apktool 2.0.0-35f978-SNAPSHOT
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether resources has changed...
I: Building resources...
I: Building apk file...
[bob@bob-x1 bin]$
```

JEB

- Interactive decompiler
- Extract resources
  - Apktool based



Anti-apktool

- Bug: headerSize not handled properly

```
[bob@bob-x1 w]$ apktool d x01-bad-head-size.apk
I: Using Apktool 2.0.0-35f978-SNAPSHOT on x01-bad-head-size.apk
I: Loading resource table...
Exception in thread "main" brut.androlib.AndrolibException: Could not decode arsc file
        at brut.androlib.res.decoder.ARSCDecoder.decode(ARSCDecoder.java:52)
        at brut.androlib.res.AndrolibResources.getResPackagesFromApk(AndrolibResources.java:630)
        at brut.androlib.res.AndrolibResources.loadMainPkg(AndrolibResources.java:73)
        at brut.androlib.res.AndrolibResources.getResTable(AndrolibResources.java:65)
        at brut.androlib.Androlib.getResTable(Androlib.java:63)
        at brut.androlib.ApkDecoder.setTargetSdkVersion(ApkDecoder.java:209)
        at brut.androlib.ApkDecoder.decode(ApkDecoder.java:92)
        at brut.apktool.Main.cmdDecode(Main.java:165)
        at brut.apktool.Main.main(Main.java:81)
Caused by: java.io.IOException: Expected: 0x001c0001, got: 0xffffffff
        at brut.util.ExtDataInput.skipCheckChunkTypeInt(ExtDataInput.java:75)
        at brut.androlib.res.decoder.StringBlock.read(StringBlock.java:44)
        at brut.androlib.res.decoder.ARSCDecoder.readPackage(ARSCDecoder.java:102)
        at brut.androlib.res.decoder.ARSCDecoder.readTable(ARSCDecoder.java:78)
        at brut.androlib.res.decoder.ARSCDecoder.decode(ARSCDecoder.java:47)
        ... 8 more
```

Anti-JEB

- Bug: special class name “pnf.this.object.does.not.Exist”

```
.method public static main([String)V
          .registers 3
00000000  const-string            v0, "hello"
00000004  sget-object             v1, System->out:PrintStream
00000008  invoke-virtual          PrintStream->println(String)V, v1, v0
0000000E  invoke-static           Exist->a()V
00000014  return-void
.end method
```

Assembly | Decompiled Java | Strings | Constants | Notes

```
public class Hello {
    public Hello() {
        super();
    }

    public static void main(String[] arg2) {
        // Decompilation failed
    }
}
```

Anti-emulation

- Battery/Signal status
- IMEI
- Properties
  - “sdk” in Build.PRODUCT
- Files
  - /system/bin/qemud
  - /dev/socket/qemud
  - /dev/qemu_pipe
- Goldfish kernel

- Qemu specific behavior
  - Binary transform

http://www.dexlabs.org/blog/btdetect

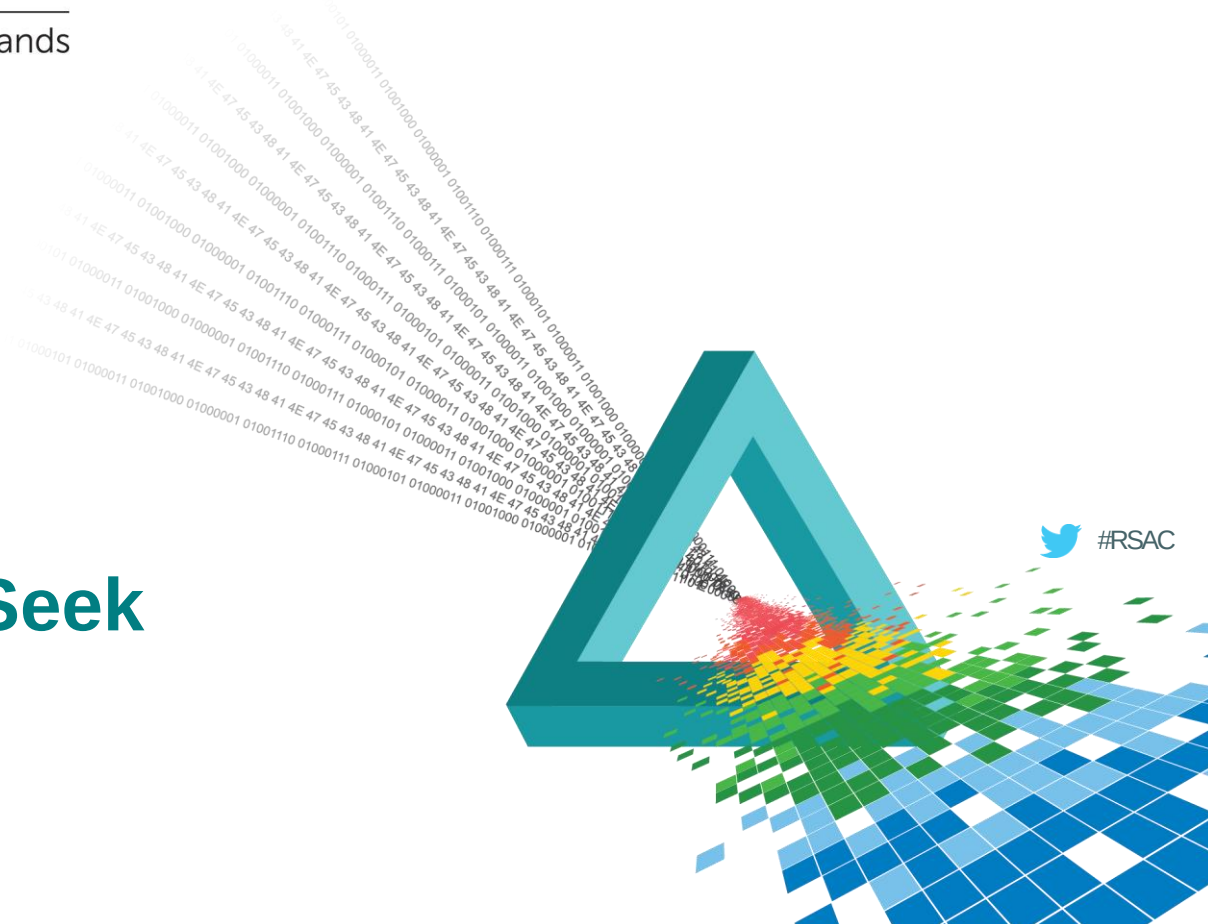
  - Low level cache

https://bluebox.com/technical/android-emulator-detection-by-observing-low-level-caching-behavior

Anti-debug

- Anti-ptrace
  - PTRACE_ME
  - A trace B trace C
- Check for /proc/pid/status
  - State: T (tracing stop)
  - TracerPid: xxxx
- Check for process named gdb

Game: Hide and Seek



Gen.1 – Hiding entire apk in resources

```
champagne:gamex/assets tstrazzere$ hexdump -C logos.png | head
00000000  42 59 11 16 18 12 12 1a  12 12 5e 91 6f 52 a4 12  |BY........^.oR..|
00000010  55 4c 58 49 12 12 58 49  12 12 1d 12 15 12 73 61  |ULXI..XI......sa|
00000020  61 77 66 61 3d 7b 71 7d  7c 3c 62 7c 75 ec d8 12  |awfa={q}|<b|u...|
00000030  12 12 12 12 50 4b 03 04  0a 00 00 08 00 00 af 84  |....PK..........|
00000040  7c 40 93 03 c4 66 bc 00  00 00 bc 00 00 00 0f 00  ||@...f..........|
00000050  07 00 61 73 73 65 74 73  2f 69 63 6f 6e 2e 70 6e  |..assets/icon.pn|
00000060  67 fe ca 00 00 00 00 00  39 35 25 24 31 38 32 38  |g.......95%$1828|
00000070  24 33 39 34 25 3e 32 38  34 3f 39 34 39 36 38 39  |$394%>284?949689|
00000080  3c 3b 29 31 39 29 25 3d  3d 39 3d 31 29 3d 38 29  |<;)19)%==9=1)=8)|
00000090  25 3d 31 3c 30 25 34 3c  3b 34 25 3c 38 31 29 25  |%=1<0%4<;4%<81)%|
```

Gen.1 – Hiding entire apk in resources

```
 Length     Date   Time    Name
--------    ----   ----    ----
     188  03-28-12 16:37   assets/icon.png
     311  03-29-12 16:25   assets/logo.png
    5666  03-27-12 22:15   res/drawable/ic_launcher.png
    2704  03-29-12 16:26   AndroidManifest.xml
     792  03-29-12 16:26   resources.arsc
   27408  03-29-12 16:26   classes.dex
     472  03-29-12 16:26   META-INF/MANIFEST.MF
     525  03-29-12 16:26   META-INF/CERT.SF
    1077  03-29-12 16:26   META-INF/CERT.RSA
--------                   -------
   39143                   9 files
```

Gen.1 – Hiding entire apk in resources

- Payload takes effect after install/update

```
Intent i = new Intent(Intent.ACTION_VIEW);
i.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
i.setDataAndType(Uri.parse("file://" + apkfile),
                 "application/vnd.android.package-archive");
ctx.startActivity(i);
```

- User interaction required
- Easy to detect by anti-malware

Gen.2 – Load dex separately

- Load dex with the Java Reflection API

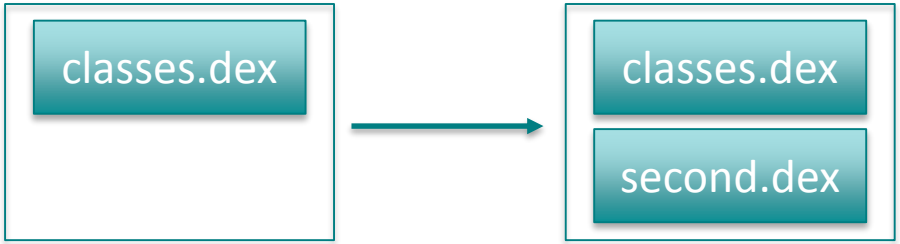
```
cl = new DexPathClassLoader("/sdcard/extra.jar",...);
clz = cl.loadClass("com.test.Action");
Action action = (Action)clz.newInstance();
```

- A simple fix to MethodID size limit (< 65535)

- Choose class carefully in secondary dex

Gen.2.1 – Modify the default ClassLoader

- Inject secondary dex to ClassLoader



- Same with Multidex in android support
- Load Stub Application
- Restore the Application

```
ctx.mOuterContext
ctx.mPackageInfo.mApplication
ctx.mPackageInfo.mActivityThread.mInitialApplicatio
ctx.mPackageInfo.mActivityThread.mAllApplications
```

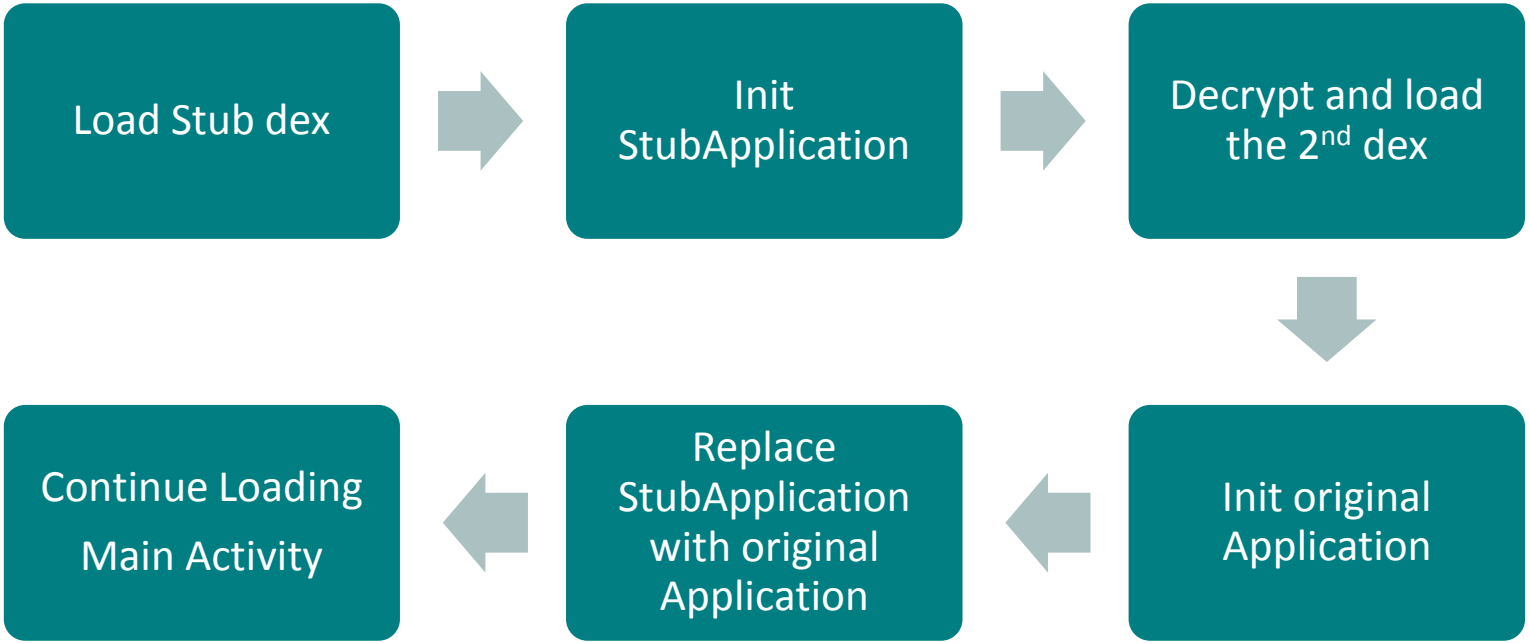
Become an APK Protector

```
  Length      Date    Time    Name
---------  ---------- -----   ----
     1440  2015-04-15 16:28   AndroidManifest.xml
     2012  2015-04-15 16:28   classes.dex
     9193  2015-04-15 14:44   res/drawable-hdpi-v4/ic_launcher.png
     2658  2015-04-15 14:44   res/drawable-ldpi-v4/ic_launcher.png
     5057  2015-04-15 14:44   res/drawable-mdpi-v4/ic_launcher.png
    14068  2015-04-15 14:44   res/drawable-xhdpi-v4/ic_launcher.png
      640  2015-04-15 16:28   res/layout/main.xml
     1692  2015-04-15 14:44   resources.arsc
---------                     -------
    36760                     8 files
```

After:

```
  Length      Date    Time    Name
---------  ---------- -----   ----
     1724  2015-06-30 14:10   AndroidManifest.xml
    59248  2015-06-30 14:10   classes.dex
     2012  2015-06-30 14:10   juicer-classes1.dex
     9193  2015-06-30 14:10   res/drawable-hdpi-v4/ic_launcher.png
     2658  2015-06-30 14:10   res/drawable-ldpi-v4/ic_launcher.png
     5057  2015-06-30 14:10   res/drawable-mdpi-v4/ic_launcher.png
    14068  2015-06-30 14:10   res/drawable-xhdpi-v4/ic_launcher.png
      640  2015-06-30 14:10   res/layout/main.xml
     1692  2015-06-30 14:10   resources.arsc
---------                     -------
    96292                     9 files
```

How it works


Load Stub dex
Init StubApplication
Decrypt and load the 2nd dex
Init original Application
Replace StubApplication with original Application
Continue Loading Main Activity

Weakness

- Dex must be extracted to file system
- Odex is generated to file system
  - Almost equal to dex file
  - It’s easy to recover to dex by Smali/baksmali

```
> baksmali -x -d framework abc.odex
> smali abc -o abc.dex
```

- Quick fix: delete the dex/odex files !

Gen.2.3 – Hijack read/write function

- Tool like Cydia Substrate
- Encrypt before write to file system
- Decrypt before read to memory

Gen.2.4 – Load dex on the fly

```
/*
 * Given an open optimized DEX file, map it into read-only shared memory and
 * parse the contents.
 *
 * Returns nonzero on error.
 */
int dvmDexFileOpenFromFd(int fd, DvmDex** ppDvmDex)
```

```
/*
 * Create a DexFile structure for a "partial" DEX.  This is one that is in
 * the process of being optimized.  The optimization header isn't finished
 * and we won't have any of the auxillary data tables, so we have to do
 * the initialization slightly differently.
 *
 * Returns nonzero on error.
 */
int dvmDexFileOpenPartial(const void* addr, int len, DvmDex** ppDvmDex)
```

Wait a sec.

- dvmDexFileOpenPartial is the key
  - Hack it we get the dex again !
- Dex is still alive in memory
  - Continuously
  - Get the Memory = Get the dex

Dumping by GDB

- Connect to target process by ***gdb -- <pid>***
- ***gcore*** to dump entire memory

```
04f879e0: 20c5 914a 1300 0000 0200 0000 0000 0000   ..J............
04f879f0: 20c5 914a 1300 0000 0400 0000 0000 0000   ..J............
04f87a00: 20c5 914a cb07 0000 6465 780a 3033 3500   ..J....dex.035.
04f87a10: 780a c71f ad5a fbba 5cc1 a3cd 12df 5d96  x....Z..\.....].
04f87a20: aa5d 921e 40ea e239 c007 0000 7000 0000  .]..@..9....p...
04f87a30: 7856 3412 0000 0000 0000 0000 f006 0000  xV4.............
04f87a40: 2500 0000 7000 0000 1000 0000 0401 0000  %...p...........
04f87a50: 0300 0000 4401 0000 0400 0000 6801 0000  ....D.......h...
04f87a60: 0c00 0000 8801 0000 0700 0000 e801 0000  ................
04f87a70: f804 0000 c802 0000 2a04 0000 3204 0000  ........*...2...
04f87a80: 4404 0000 4b04 0000 4e04 0000 6504 0000  D...K...N...e...
```

Dumping by /proc/pid/mem

- ptrace to pause target thread
- Search '**dex.035**' on each item /proc/pid/maps
- Read out data from /proc/pid/mem

- android-unpacker by strazzere
  - https://github.com/strazzere/android-unpacker

Gen.2.5 – Wipe the dex header

```
03a03250:                                       6465 790a 3033 3600  ........dey.036.
03a03260: 2800 0000 786d 0900 a06d 0900 3403 0000  (...xm...m..4...
03a03270: d870 0900 603e 0100 0000 0000 3fb9 26ca  .p..`>......?.&.
03a03280: 6465 780a 3033 3500 9caf 3f5a 1305 e97a  dex.035...?Z...z
03a03290: af37 8db3 d94a cdc8 8bed dec0 aa9f eed8  .7...J..........
03a032a0: 786d 0900 7000 0000 7856 3412 0000 0000  xm..p...xV4.....
```

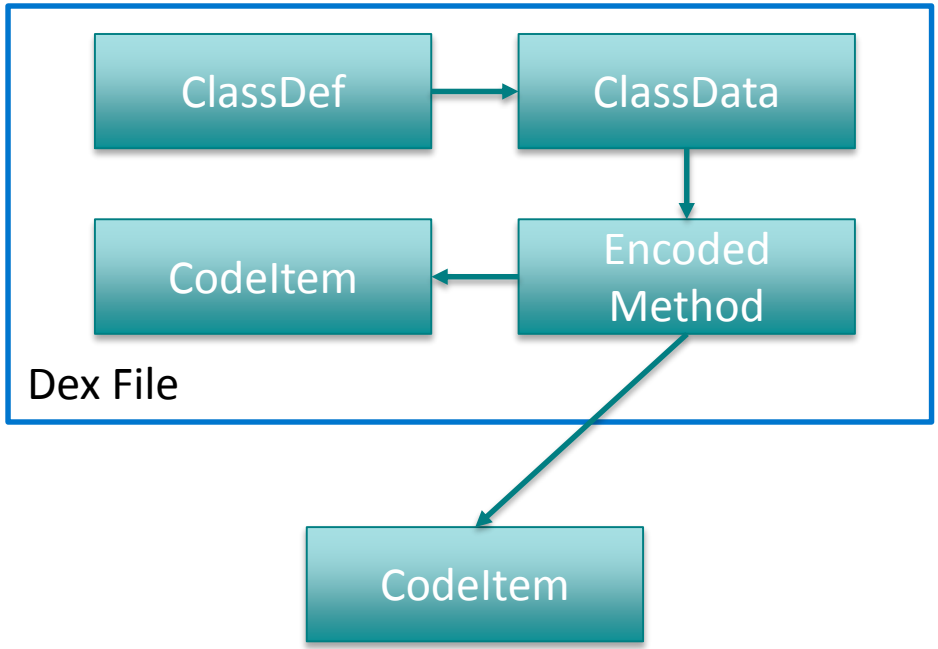
Normal

```
01e97250:                                       0000 0000 0000 0000  ................
01e97260: 0000 0000 d00e 0000 f80e 0000 7303 0000  ............s...
01e97270: 7012 0000 d800 0000 0000 0000 cb76 b092  p............v..
01e97280: 0000 0000 0000 0000 0000 0000 0000 0000  ................
01e97290: 0000 0000 0000 0000 0000 0000 0000 0000  ................
01e972a0: d00e 0000 7000 0000 7856 3412 0000 0000  ....p...xV4.....
01e972b0: 0000 0000 a007 0000 2600 0000 0000 0000  ........&.......
```

Wipped

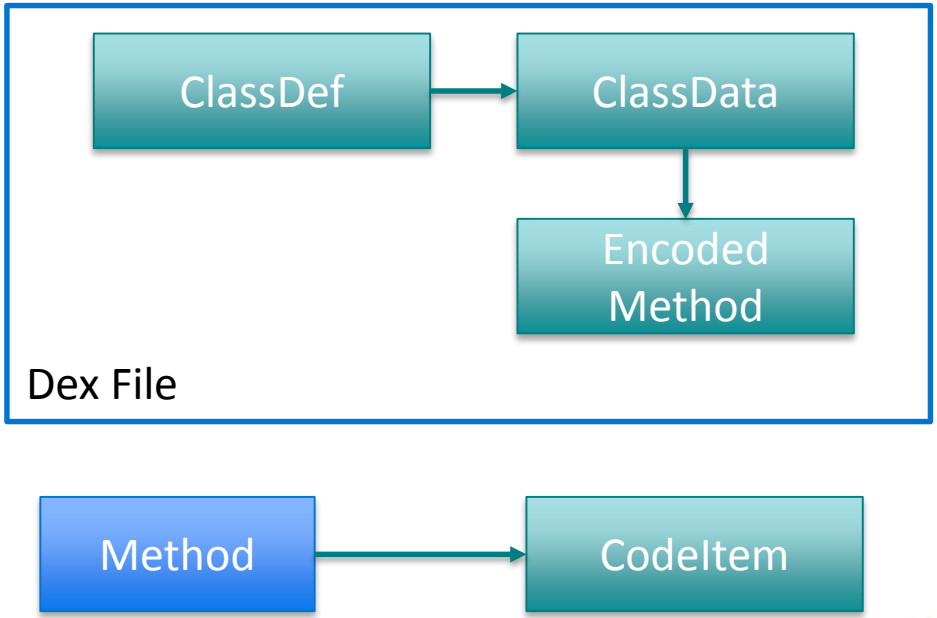
Gen.3 – Strip code from dex file

- Restore before execute
- None continuously
- Can’t dump as offset + size
  - Need rebuild dex

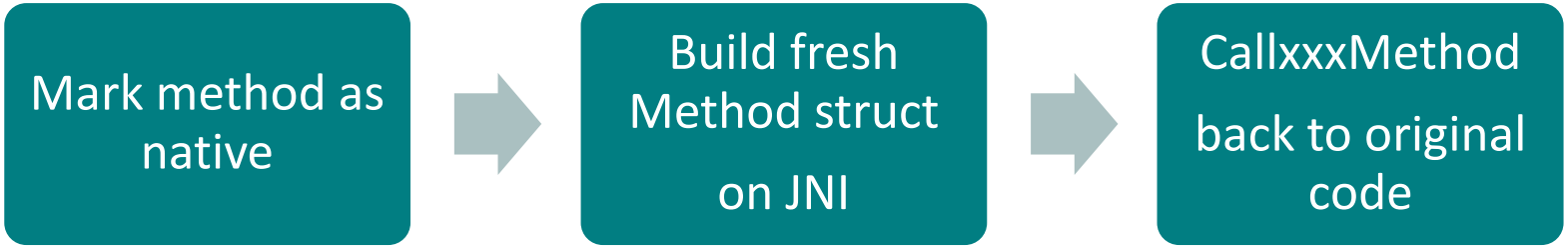


Gen.3.1 – Restore code internally

- Restore to dalvik internal Method struct
- Travel all Method struct before rebuild dex

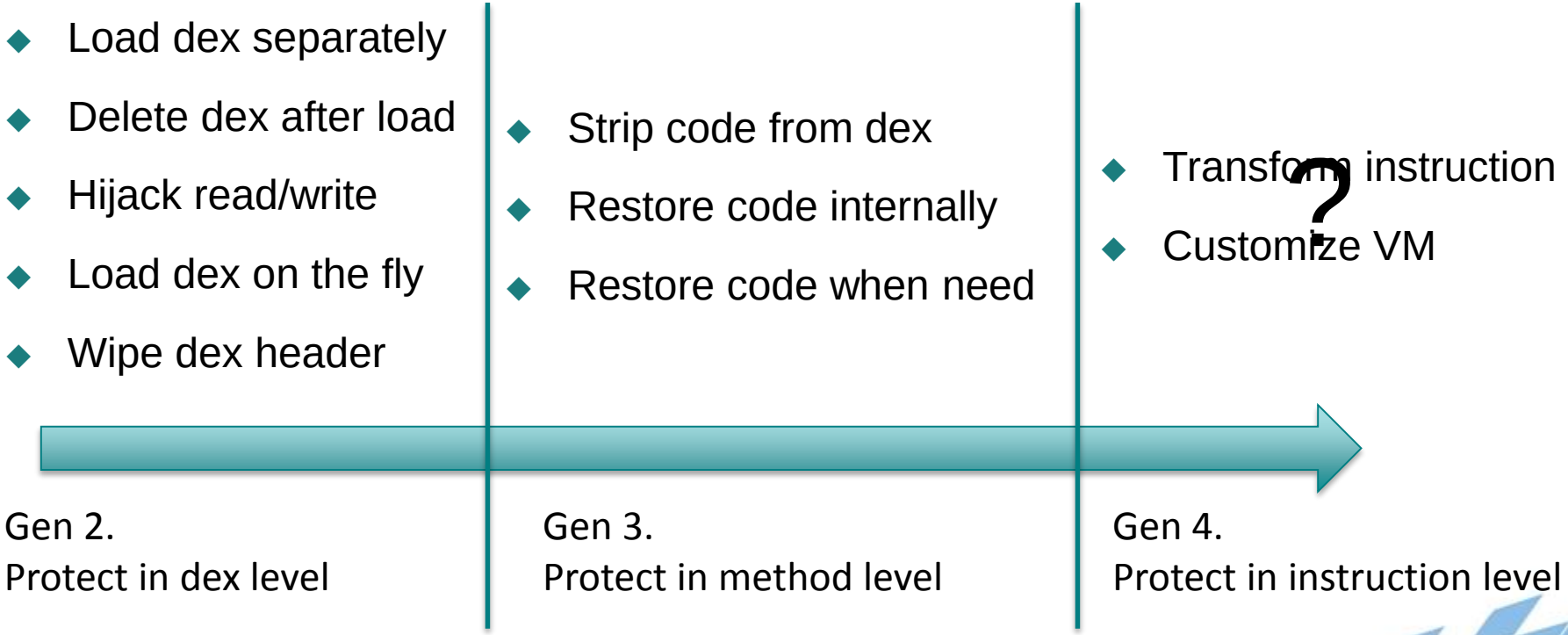


Gen.3.2 – Restore code when need



- jmethodID is a pointer to Method struct
- Harder to dump
  - Need trigger all method invocation to get code

All-in-one roadmap to hide dex



Summary

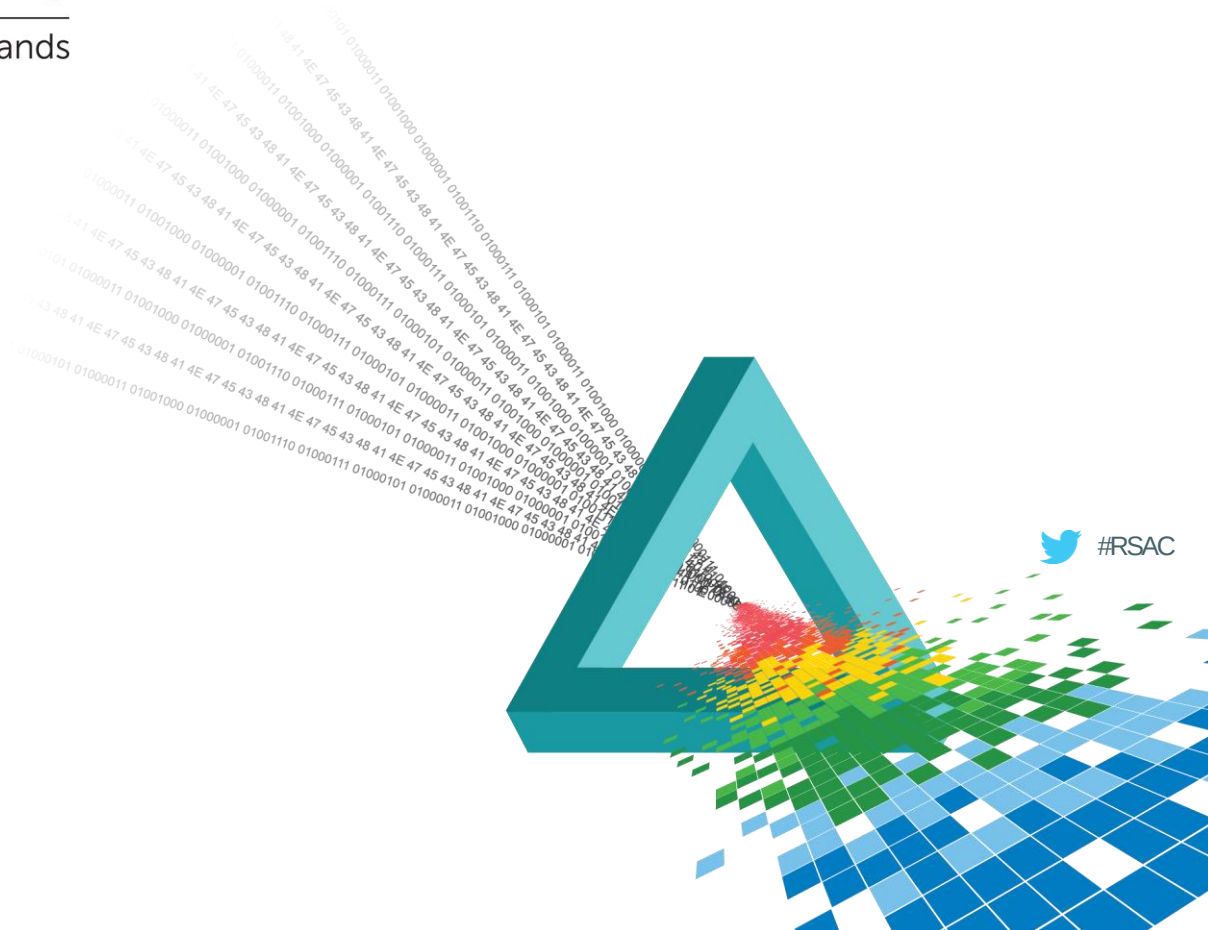
- Easy to use
- Tricks in dalvik to hide dex
- Tool bugs to prevent from disassemble
- Stop most of beginners
- It can be broken, but takes a lot of time
- Need one ? Try http://jaq.alibaba.com

RSAConference2015

Singapore | 22-24 July | Marina Bay Sands

Q & A

pxb1988@gmail.com



Apply: Build your own Apk Protector

- Multi-dex is a good start point
- You can find everything on these slides and Google
- Have fun

Reference

- Reports from teammates in Alibaba.
- STRAZZERE&SAWYER: ANDROID HACKER PROTECTION LEVEL 0
- STRAZZERE: Dex Education: Practicing Safe Dex
- http://bbs.pediy.com/showthread.php?p=1353353