移动恶意代码高级对抗技术解析和前瞻

演讲人：Tom:Pan（潘宣辰）

安天实验室/AVL移动安全团队

主讲人介绍

- 潘宣辰，Tom:Pan
  - AVLMobileSecurityTeam@AntiyLabs
  - Founder&Leader
- 技术涉猎较广，主攻手机恶意代码取证，手机反病毒引擎和自动化分析技术，以及移动网络安全。
- tompan@antiy.cn

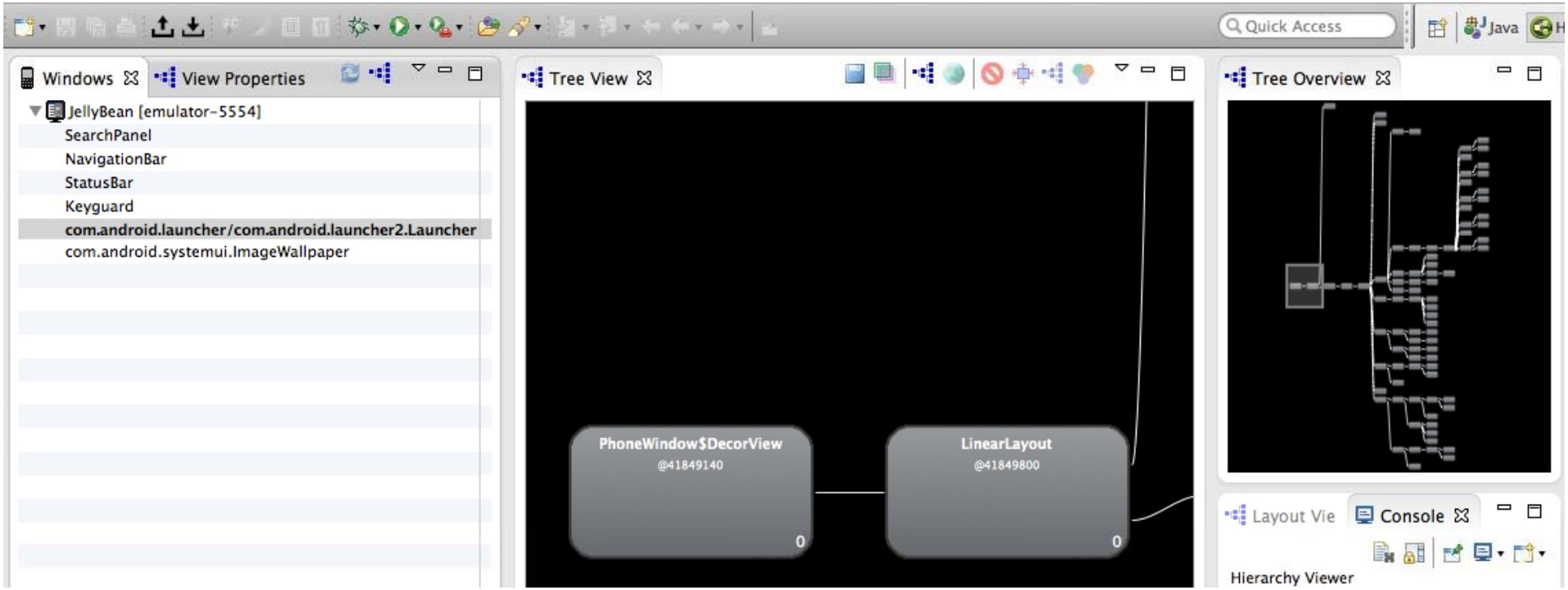
万万没想到1

大家都喜欢出来露脸了~~恶意代码不可耻么？

18 2014-03-25, 22:38:15

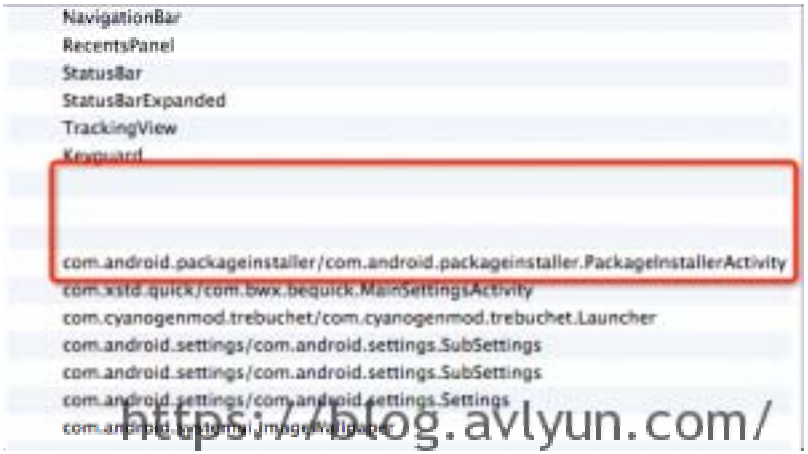
好吧,我承认这个文件是我写的,大家不用分析了,其实没那么多ad的,很多都没申请到key,关键是admob,不过这货号已经被封了,被几个黄色app搞死了,然后更新渠道没了,现在已经不玩了,快一年的事情了,在这被看到,太菜幸了

万万没想到2

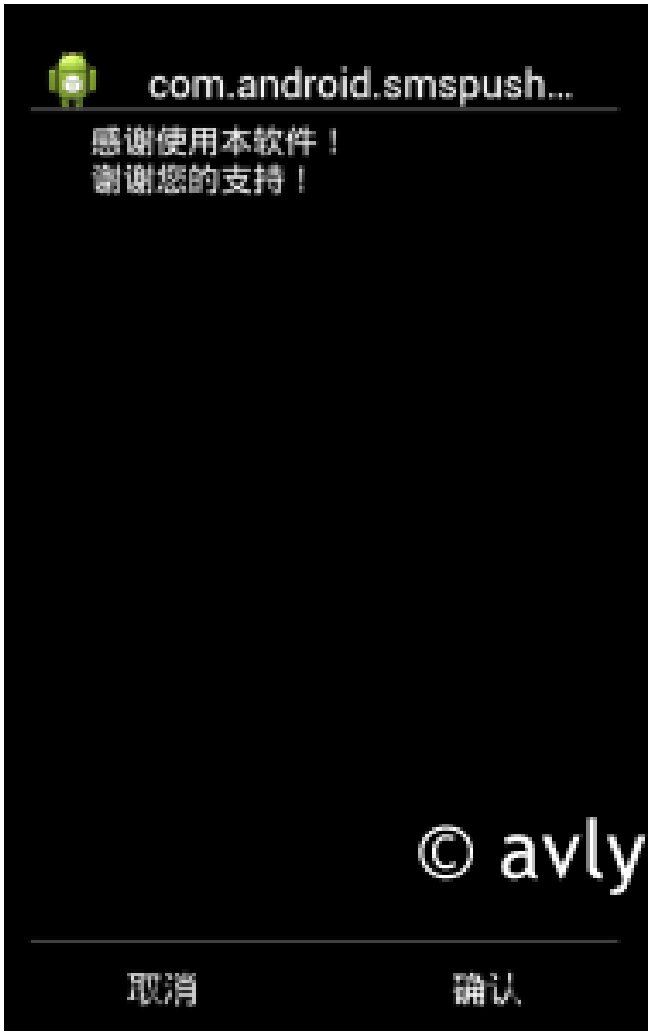

Quick Access
Java
Windows
View Properties
JellyBean [emulator-5554]
SearchPanel
NavigationBar
StatusBar
Keyguard
com.android.launcher/com.android.launcher2.Launcher
com.android.systemui.ImageWallpaper
Tree View
PhoneWindow$DecorView
@41849140
0
LinearLayout
@41849800
0
Tree Overview
Layout Vie
Console
Hierarchy Viewer

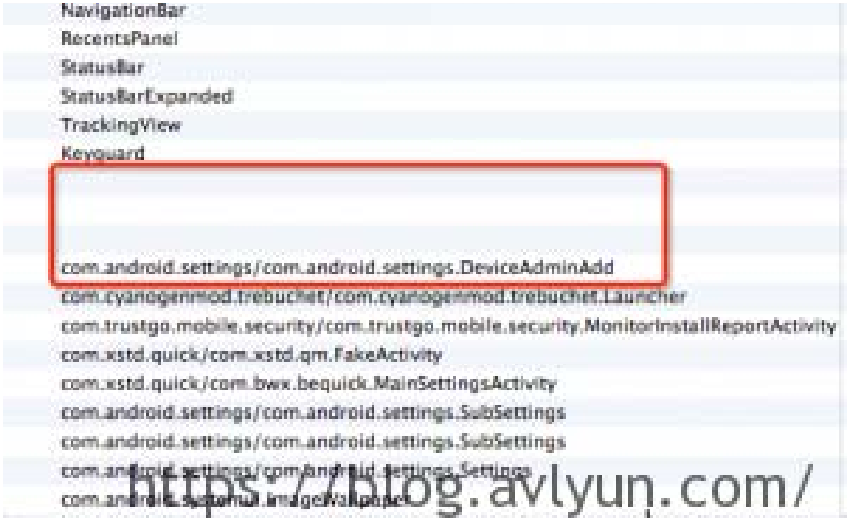
万万没想到2


com.android.smspush...
本协议约定com.android.smspush软件与用户之间关于"com.android.smspush"软件服务（以下简称"服务"）的权利义务。
1、尊重用户个人隐私信息的私有性是com.android.smspush软件的一贯制度，com.android.smspush软件采取技术措施和其他必要措施，确保用户个人隐私信息安全。
2、com.android.smspush软件在未经用户同意不向任何第三方公开、透露用户个人隐私信息。
3、用户不得利用本服务制作、上载、复制、发布、传播如下法律、法规和政策禁止的内容：
(1) 反对宪法所确定的基本原则的；
(2) 危害国家安全，
取消 请点击
(3) 损害国家荣誉和利益的；
(4) 煽动民族仇恨、民族
族团结的；
© avly
36秒


NavigationBar
RecentsPanel
StatusBar
StatusBarExpanded
TrackingView
Keyguard
com.android.packageinstaller/com.android.packageinstaller.PackageInstallerActivity
com.xsld.quick/com.bwx.bequick.MainSettingsActivity
com.cyanogenmod.trebuchet/com.cyanogenmod.trebuchet.Launcher
com.android.settings/com.android.settings.SubSettings
com.android.settings/com.android.settings.SubSettings
com.android.settings/com.android.settings.Settings
https://blog.avlyun.com/

万万没想到2


com.android.smspush...
感谢使用本软件！
谢谢您的支持！
© avly
取消
确认


NavigationBar
RecentsPanel
StatusBar
StatusBarExpanded
TrackingView
Keyguard
com.android.settings/com.android.settings.DeviceAdminAdd
com.cyanogenmod.trebuchet/com.cyanogenmod.trebuchet.Launcher
com.trustgo.mobile.security/com.trustgo.mobile.security.MonitorInstallReportActivity
com.xstd.quick/com.xstd.qm.FakeActivity
com.xstd.quick/com.bwx.bequick.MainSettingsActivity
com.android.settings/com.android.settings.SubSettings
com.android.settings/com.android.settings.SubSettings
https://blog.avlyun.com/

万万没想到3

- 过去主要窃取的是文本信息
  - 短信、联系人信息、当前地理位置
- 恶意代码进行环境录音、通话录音、拍照
  - 音频、视频、sdcard内容
  - 4G网络，网络传输速度更快，上传更隐蔽

万万没想到3

- 一个隐私窃取并上传至邮箱的木马
  - 2013年7月，2013年10月被发现和捕获
  - 5个用于上传的邮箱账户
  - 9K条带有短信、录音的隐私邮件，总共1.79G，累计300小时录音

万万没想到4

环境录音

通话录音

位置信息

万万没想到5

朝鲜黑客APT攻击了运营商，银行，脱了银行用户库 → 攻击者拿到了不少用户卡号和隐私信息 → 攻击者通过钓鱼向用户植入了盗号木马，强行卸载官方客户端，安装假冒客户端 → 凑齐了帐号和密码，可以分钱了 → AhnLab还是很牛的，基本3天时间就被杀了

银行数据

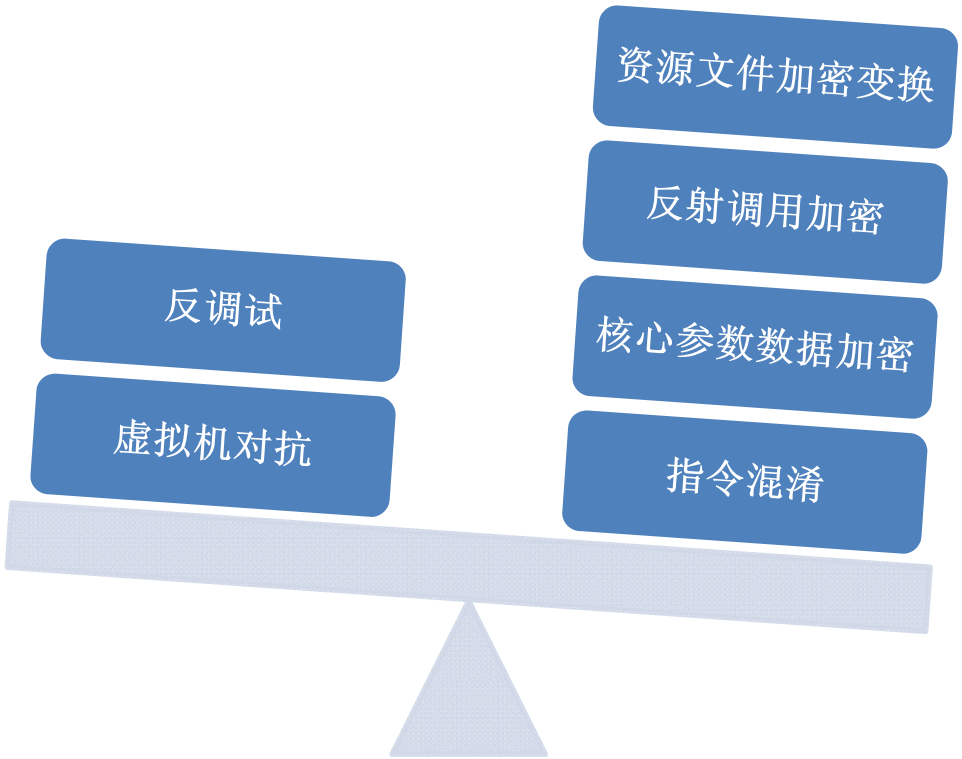
	客户端	姓名	身份证	银行卡号	银行密码	转账密码	银行名	证书密码	口令卡	银行证书
⊟ 01[redacted]										
1	01[redacted]	씨는	8[redacted]59	[redacted]	[redacted]	[redacted]	NH	[redacted]	显示	下载
2	0[redacted]4	ㄱㅅ	84[redacted]1	[redacted]	[redacted]	[redacted]	HA	[redacted]	显示	下载
3	01[redacted]704	sdxd	8[redacted]5	[redacted]	[redacted]	[redacted]	NH	[redacted]	显示	下载
⊟ +82[redacted]734										
4	+82[redacted]734	정다정	[redacted]15	[redacted]	[redacted]	[redacted]	NH	[redacted]	显示	下载
⊟ 01[redacted]										
5	01[redacted]	이광삼	7[redacted]6	[redacted]	[redacted]	无	KB	[redacted]	显示	下载
⊟ 01[redacted]20										
6	01[redacted]	예재민	941120-1123910	3560793733943	[redacted]	[redacted]	NH	[redacted]	显示	下载

移动恶意代码高级对抗技术

逆向对抗技术

动态对抗技术

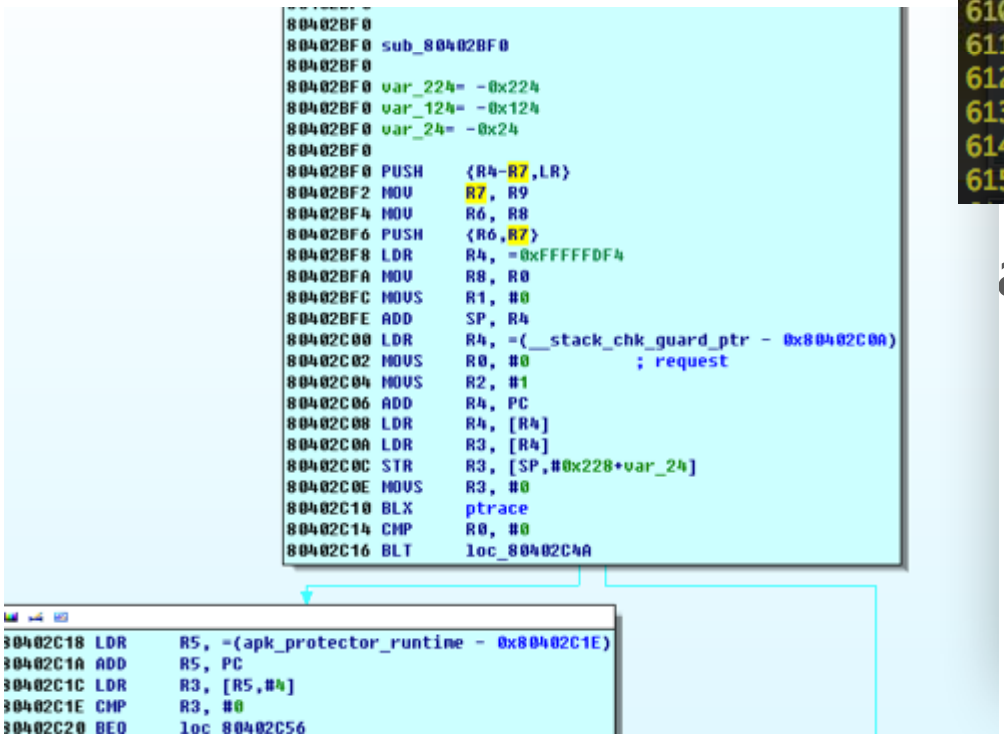
静态对抗技术



反调试

基于ptrace

ptrace(PTRACE_TRACEME,0,0,0)



gDvm结构

```
604     /* when using a native debugger, set this to sup
605     bool        nativeDebuggerActive;
606
607     /*
608      * JDWP debugger support.
609      *
610      * Note: Each thread will normally determine whe
611      * for it by referring to its subMode flags. "d
612      * seen as "debugger is making requests of 1 or
613      */
614     bool        debuggerConnected;      /* debugger
615     bool        debuggerActive;         /* debugger
```

android.os.Debug类

public static boolean **isDebuggerConnected** ()

Determine if a debugger is currently attached.

模拟器对抗

- 检测模拟器特定参数
  - DeviceId、IMEI、phone number etc.

```
public boolean isQEmuEnvDetected() {
        if(FindEmulator.hasKnownDeviceId(getApplicationContext()) ||
                FindEmulator.hasKnownImei(getApplicationContext()) ||
                FindEmulator.hasKnownPhoneNumber(getApplicationContext()) ||
                FindEmulator.hasPipes() ||
                FindEmulator.hasQEmuDriver() ||
                FindEmulator.hasQEmuFiles()) {
            log("QEmu environment detected.");
            return true;
        } else {
            log("QEmu environment not detected.");
            return false;
        }
    }
```

```
public boolean isEmulator()
{
  return (Build.MODEL.equals("sdk")) || (Build.MODEL.equals("google_sdk"));
}
```

资源加密

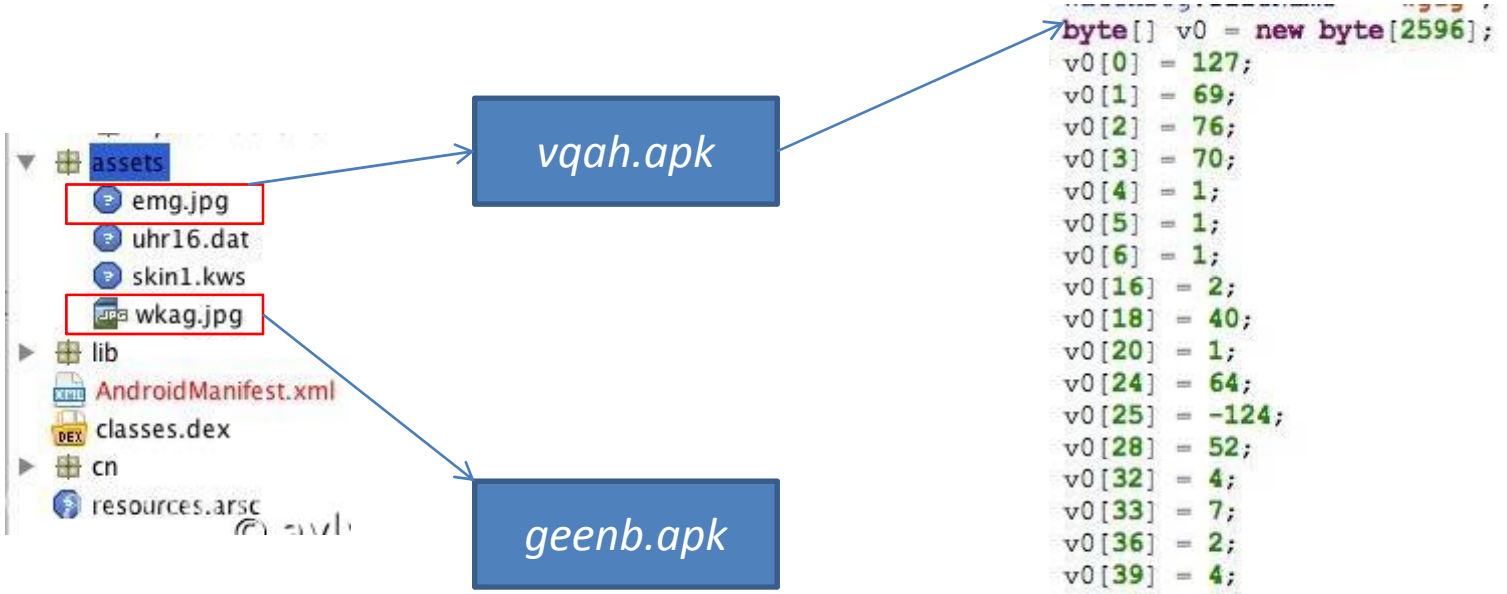
- 配置信息、解密密钥藏匿于加密的资源文件、图片文件


```
00000000  03 00 36 44 36 45 11 3E 60 53 5A 5D 56 11 44 56  ..6D6E.>`SZ]V.DV
00000010  54 66 63 5A 65 6A 00 B1 DE E3 D9 D3 D9 E1 E4 E2  TfcZej..........
00000020  11 D9 11 DE D6 11 E3 DF DC ED DB DF 11 DF E3 11  ................
00000030  DB DF DD E0 D1 DE D9 D9 11 36 44 36 45 1F 00 59  .........6D6E..Y
00000040  65 65 61 2B 20 20 55 63 60 5A 55 56 63 1F 5F 56  eea+  Uc`ZUVc._V
00000050  65 20 5D 60 52 55 20 23 25 24 27 20 56 5E 64 50  e ]`RU #%$' V^dP
00000060  63 54 1F 52 61 5C                                cT.Ra\
```

```
00000000  89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52  .PNG........IHDR
00000010  00 00 00 13 00 00 00 13 08 03 00 00 00 45 8E C6  .............E..
00000020  FE 00 00 00 09 70 48 59 73 00 00 0B 13 00 00 0B  .....pHYs.......
00000030  13 01 00 9A 9C 18 00 00 01 49 69 43 43 50 50 68  .........IiCCPPh
00000040  6F 74 6F 73 5B 74 5D 4F 51 36 62 31 75 6C 3E 5B  otos[t]OQ6b1ul>[
00000050  2F 74 5D 00 00 78 DA AD 8E 3D 4B C3 50 14 40 4F  /t]..x...=K.P.@O
00000060  AA 28 88 D4 0A 41 1C DF 24 2D 68 A5 D8 C1 8C E9  .(...A..$-h.....
```

文件加密

- 可执行文件隐藏于代码或者图片文件



反射调用加密

- 无法直接反编译得到调用函数

```java
Object v14 = Class.forName(iF$đ.ć(-29, -20131, 556)).getDeclaredConstructor(String.class).newInstance
        (iF$đ.ć(-17, -20190, 33));
if(!Class.forName(iF$đ.ć(-29, -20131, 556)).getMethod(iF$đ.ć(-33, -20138, 192), null).invoke
        (v14, null).booleanValue()) {
    v0 = iF$đ.ć(-27, -20131, 179);
    v14 = Class.forName(iF$đ.ć(-29, -20131, 556)).getDeclaredConstructor(String.class).newInstance
            (Class.forName(iF$đ.ć(-25, -20131, 142)).getMethod(iF$đ.ć(-30, -20134, 532), String
            .class, String.class).invoke(null, v0, v0));
    if(!Class.forName(iF$đ.ć(-29, -20131, 556)).getMethod(iF$đ.ć(-33, -20138, 192), null).invoke
            (v14, null).booleanValue()) {
        v14 = Class.forName(iF$đ.ć(-19, -20140, 446)).getMethod(iF$đ.ć(-14, -20134, 391), null
                ).invoke(null, null);
```

```java
ClassLoader v0 = this.getClass().getClassLoader();
Field v1 = ClassLoader.class.getDeclaredField(Application.be("□秖□豆傢榜"));
v1.setAccessible(true);
Object v2 = v1.get(v0);
Object v3 = Class.forName(Application.be("__invalid_string_21E21")).getConstructors()[0].newInstance
        (this);
```

核心参数数据加密

- 参数加密

```
static {
    a.a = b.a("XDD@;..VJDQR/WYS@/^UD;9192.C@.CI^S/QSDY_^>");
    a.b = b.a("XDD@;..VJDQR/WYS@/^UD;9190.ZCUBFUB.C@");
```

```
v3 = PhWingsRts.writeFileData("==Pz8/M3Q6U+iSjENQ8/");
String[] v2 = v1.split(v3);
v3 = "Q=Xa5OPq/t/5/jJB1CJiID3e";
v3 = PhWingsRts.writeFileData(v3);
v4 = PhWingsRts.writeFileData("M=pra3ZyemxCQ76N0jWH") + v1;
```

```
ckmoaxh.ckmoaxh = cogst.cogst("wt)Mw");
ckmoaxh.jqmsmhfjt = cogst.cogst("w)@joJ");
ckmoaxh.uuyrgsgfj = cogst.cogst("w))F5");
ckmoaxh.rnnnnhpq = cogst.cogst
ckmoaxh.eavaf = cogst.cogst("@ao@");
ckmoaxh.jesanw = cogst.cogst("@aq@");
ckmoaxh.mrvxkb = cogst.cogst(")M/vo");
ckmoaxh.qlwej = cogst.cogst("v/i@b@Pwi@/v_Pi@/v");
ckmoaxh.akfewobyt = cogst.cogst("v/i@b@Pwi@/voJ)o");
```

核心参数数据加密

- URL加密

```
aUl_U_          DCB "?-#1.(#",0x24,"-&(-",0x24,"?.&(-",0
                                        ; DATA XREF: .text:00000E06↑o
                                        ; .text:off_E5C↑o ...
                ALIGN 4
a33             DCB "'33/"
                DCD 0x20EEEEF9
                DCD 0x32ED232D,0x2A2E2325,0x2E22ED22,0xEFF8F92C,0xE2E2EFF8,0x2F333327,0x20EEEEF9,0x26ED232D,0x2E2E2C2E,0xED31242D
                DCD 0xF92C2E22,0xEFF8EFF8,0x33[illegible]E2,0xEEF92F33,0x232D20EE,0x2F2E2AED,0x2E2F2E2F,0x2C2E22ED,0xF8EFF8F9,0x27E2E2EF
                DCD 0xF92F3333,0x2036EEEE,0x20[illegible]2F,0x28222033,0x22ED2822,0xF8F92C2E,0xE2EFF8EF,0x333327E2,0xEEEEF92F,0x2F34ED36
                DCD 0xEDF8F7F8,0xF92C2E22,0xEF[illegible]F8,    0

                ; DCD 0x74
aUl_U_          DCB "/androidengine/loginA" ; DATA XREF: .text:00000E06↑o
                                        ; .text:off_E5C↑o ...
                ALIGN 4
aHttpAnd_sfdokc DCB "http://and.sfdokc.com:9090##http://and.gomooner.com:9090##http:/"
                DCB "/and.koppopo.com:9090##http://wap.datacici.com:9090##http://w.up"
                DCB "989.com:9090",0
                DCB    0
                DCB    0
                DCB    0
```

CFG混淆

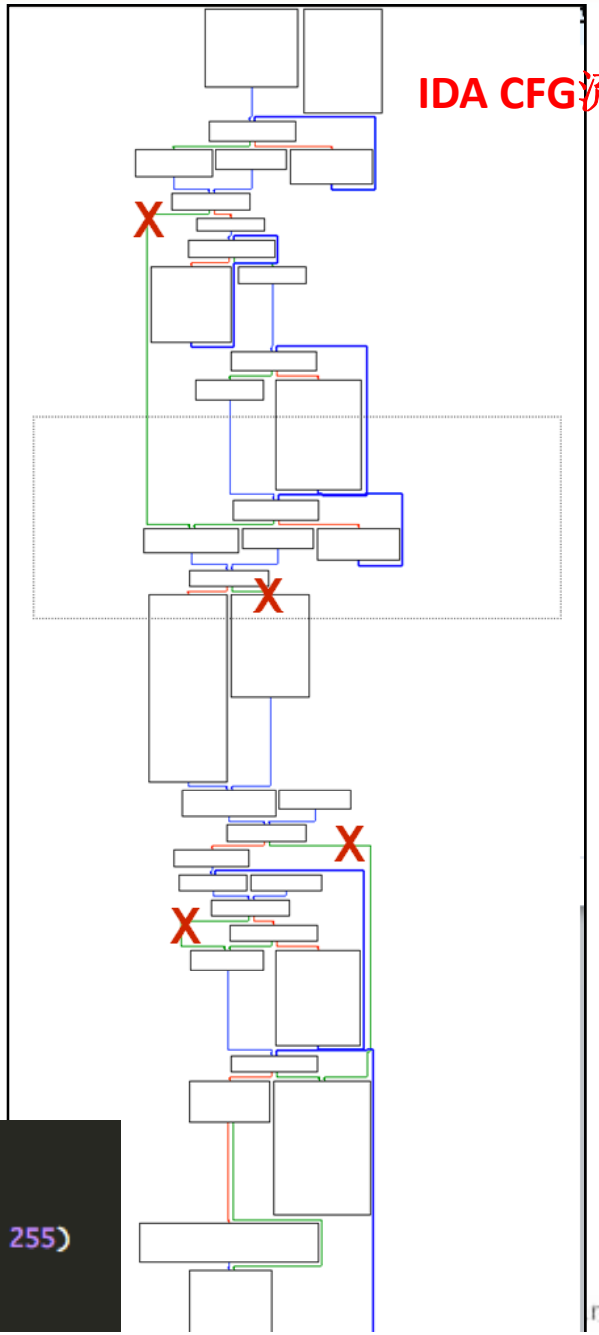
JEB反编译结果

```
private static final int round(byte[] arr, int sh) {
    int v1 = 0;
    int v0 = arr[14] << 16;
    int v2 = 0;
    while(v2 == 0) {
        v2 = 3;
        int lb = sh & 255;
        try {
            return arr[sh >> 24 & 255] << 24 | (arr[lb] & 255 | (arr[sh >> 8 & 255] & 255) << 8 | (
                    arr[sh >> 16 & 255] & 255) << 16);
        }
        catch(Exception v3_1) {
            continue;
        }
    }

    while(v1 == 0) {
        v1 = 2;
        v2 = sh & 127;
        try {
            return arr[v2] >> 8;
        }
        catch(Exception v2_1) {
            continue;
        }
    }

    return v0;
}
```

函数真实实现

```
private static int round(byte[] arr, int sh) {
    int lb = sh & 255;

    return arr[sh >> 24 & 255] << 24 | (arr[lb] & 255 | (arr[sh >> 8 & 255] & 255)
            << 8 | (arr[sh >> 16 & 255] & 255) << 16);
}
```

IDA CFG流程



花指令

```
.method private static final shift(I, I)I
        .registers 4
00000000  add-int                  v0, p1, p0
00000004  shr-int/lit8             v0, v0, 0x18
00000008  ushr-int                 v0, p0, p1
0000000C  neg-int                  v1, p1
0000000E  shl-int                  v1, p0, v1
00000012  or-int                   p1, v0, v1
00000016  return                   p1
.end method
```

无意义指令

dex2jar+JD-Gui反编译结果

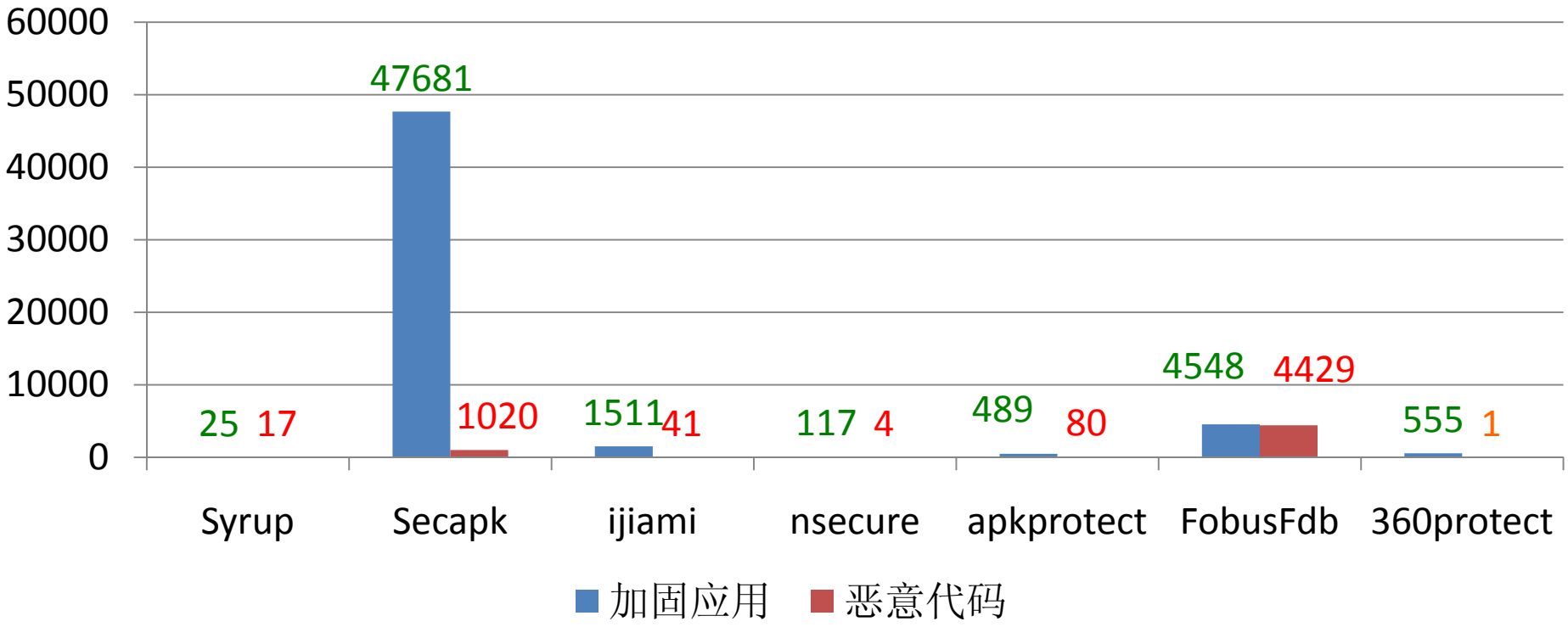
```
private static final int bdafd(int paramInt1, int paramInt2)
{
  (paramInt2 + paramInt1 >> 24);
  return paramInt1 >>> paramInt2 | paramInt1 << -paramInt2;
}
```

加壳技术

壳	描述	加壳类别简介
Syrup	加密填充dex头部	DEX动态加载+DEX头部填充
Secapk	梆梆加固	DEX动态加载+ELF加壳+SO加壳
ijiami	爱加密加固	DEX动态加载+SO加壳
nsecure	恶意代码或广告件使用	DEX动态加载
apkprotect	国外加固方案	DEX opcode修改
FobusFdb	恶意代码采用的纯Java加壳	DEX动态加载
360protect	360应用加固	DEX opcode修改
DSdk	广告件加壳	DEX动态加载
dexunshell	恶意代码或广告件使用	DEX动态加载+DEX尾部填充
netonshell	一种私人的加壳技术	DEX动态加载
txprotect	腾讯应用加固	DEX索引结构修改

Android加壳技术

- 商用加固方案——应用保护
- 恶意代码、风险软件，流氓软件对抗手段

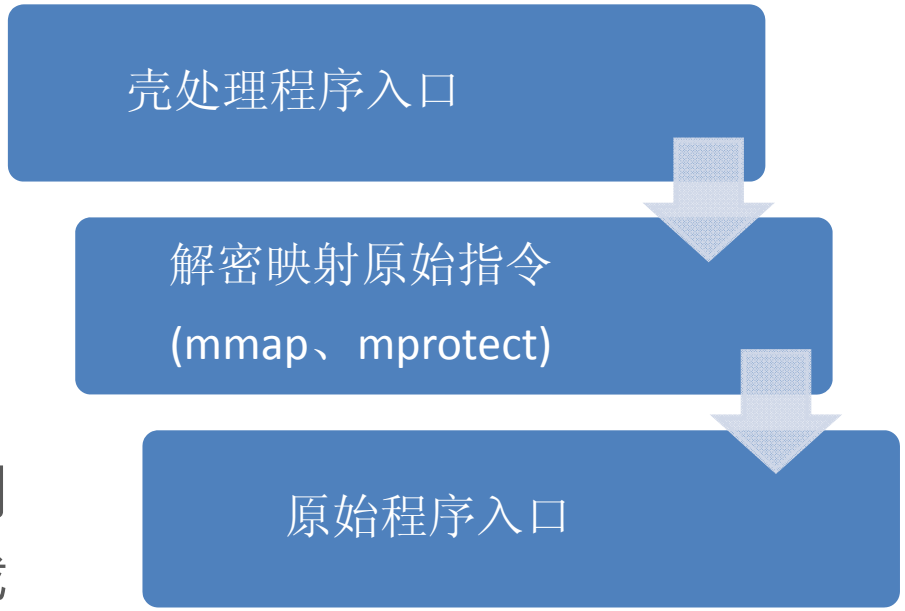


加壳技术

- 可执行文件+壳
  - PC时代
    - PE格式（UPX、ASProtect、VMP等等）
  - Android
    - DEX
    - ELF，SO

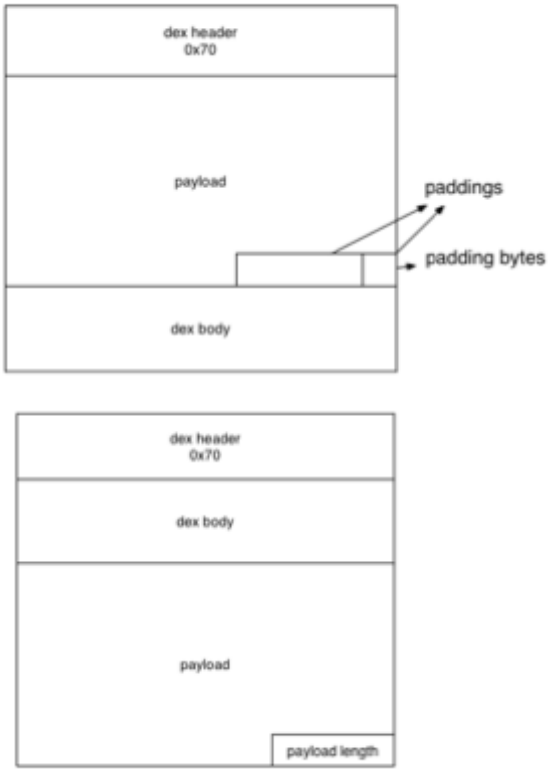
Android加壳技术-ELF/SO加壳

- ELF加壳技术
  - 与PE加壳类似
  - 入口由ELF头部的entry point指向
- SO加壳技术
  - 原理与ELF类似，入口不同
    - 由/system/bin/linker加载
  - 三种入口方式:
    - .init_proc
    - .init_array
    - JNI_OnLoad



Android加壳技术-DEX加壳

- DEX加壳主要有三种技术思路
  - 加密原有的可执行文件并植入新的APK或DEX，运行时首先分离释放加密的可执行文件，然后进行解密，最后动态加载
    - 加密可执行文件植入方式
      - DEX头部
      - DEX尾部
      - 资源文件(assets/或者res/raw/)



Android加壳技术-DEX加壳

– 运行时修改DEX opcode

```
.method public toByteArray()[B
    .registers 4

    .prologue
    .line 62
:try_start_0
0000                    nop
0000                    nop
0000                    nop
0000                    nop
0000                    nop
0000                    nop
    .line 63
0000                    nop
0000                    nop
0000                    nop
0000                    nop
    .line 64
0000                    nop
0000                    nop
0000                    nop
    .line 65
0000                    nop
0000                    nop
0000                    nop:try_end_10
    .catch Ljava/io/IOException; {:try_start_0 .. :try_end_10} :catch_11

    .line 66
0000                    nop
    .line 67
```

```
.method public toByteArray()[B
    .registers 4

    .prologue
    .line 62
:try_start_0
6E10 A732 0300          invoke-virtual    {p3}, Lcom/google/protobuf/AbstractMessageLite;->getSerializedSize()I

0A00                    move-result       p0
2300 F711               new-array         p0, p0, [B
    .line 63
7110 8C33 0000          invoke-static     {p0},
Lcom/google/protobuf/CodedOutputStream;->newInstance([B)Lcom/google/protobuf/CodedOutputStream;

                        move-result-object    p1
    .line 6
6E20 AB32 1300          invoke-virtual    {p3, p1},
Lcom/google/protobuf/AbstractMessageLite;->writeTo(Lcom/google/protobuf/CodedOutputStream;)V

    .line 65
6E10 5833 0100          invoke-virtual    {p1}, Lcom/google/protobuf/CodedOutputStream;->checkNoSpaceLeft()V
:try_end_10
    .catch Ljava/io/IOException; {:try_start_0 .. :try_end_10} :catch_11

    .line 66
1100                    return-object     p0
    .line 67
:catch_11
0D00                    move-exception    p0
```

Android加壳技术-DEX加壳

– 修改DEX结构，无法正常寻址代码段混乱，运行时完成地址和属性索引的修复

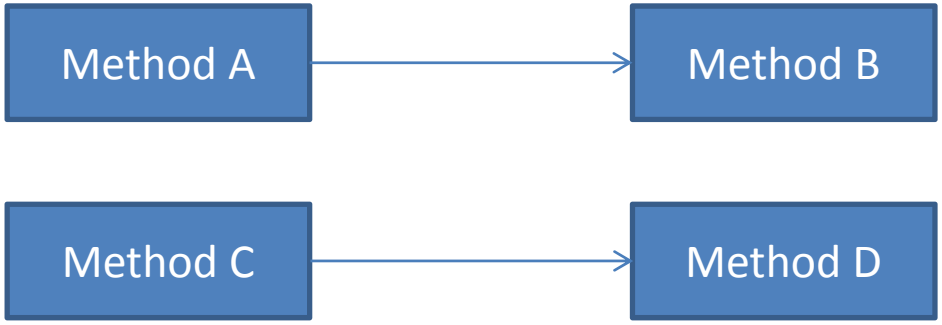
struct encoded_method method[0]	public void com.exar
struct uleb128 method_idx_diff	0x20
struct uleb128 access_flags	(0x1) ACC_PUBLIC
struct uleb128 code_off	0x154C
struct code_item code	3 registers, 2 in argu

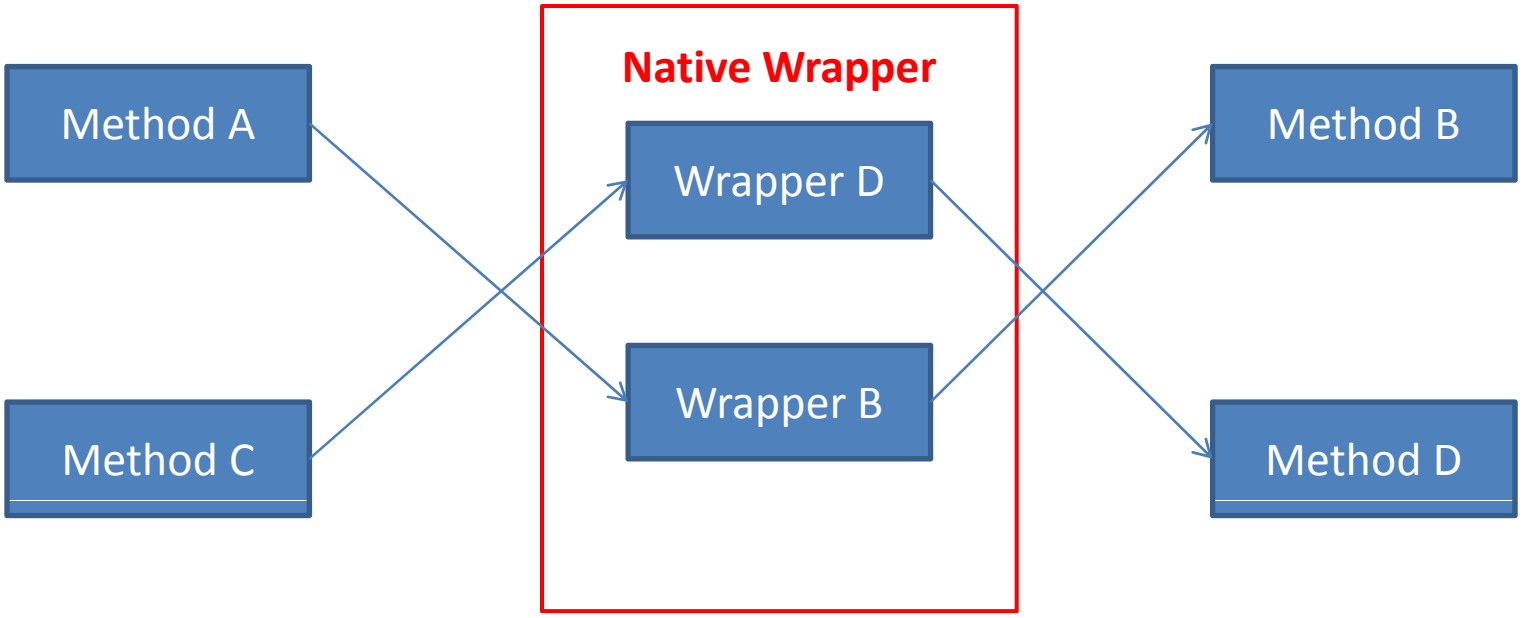
struct encoded_method method[1]	public native void com.example.bar
struct uleb128 method_idx_diff	0x1
struct uleb128 access_flags	(0x101) ACC_PUBLIC ACC_NATIV
struct uleb128 code_off	0x0

struct encoded_method method[2]	public void com.exar
struct uleb128 method_idx_diff	0x1
struct uleb128 access_flags	(0x1) ACC_PUBLIC
struct uleb128 code_off	0x1678
struct code_item code	2 registers, 1 in argu

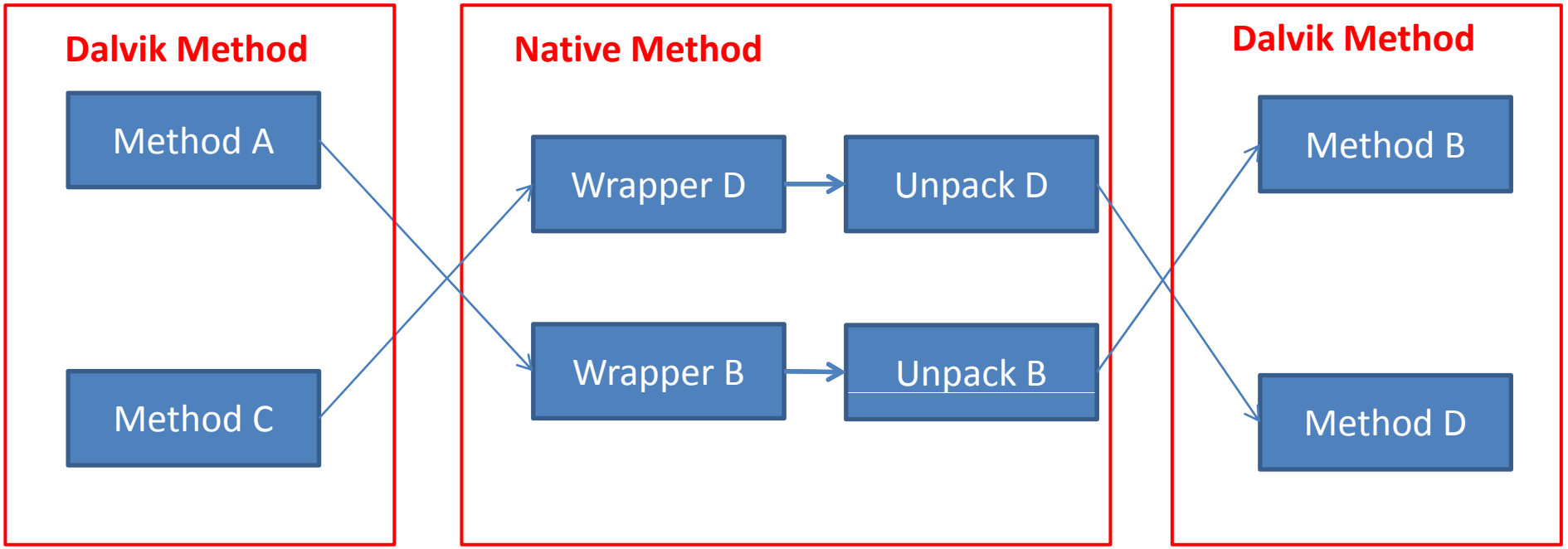
```
1540h: 11 01 D8 02 02 01 28 E5 12 01 28 FB 03 00 02 00
1550h: 02 00 00 00 4A 4B 00 00 0D 00 00 00 6F 20 01 00
1560h: 21 00 62 00 09 00 38 00 07 00 62 00 09 00 6E 20
1570h: 01 00 20 00 0E 00 00 00 08 00 01 00 05 00 02 00
1580h: 54 4B 00 00 6B 00 00 00 6F 10 02 00 07 00 71 00
1590h: 59 00 00 00 0C 04 39 04 05 00 71 10 61 00 07 00
15A0h: 62 00 07 00 71 00 59 00 00 00 0C 04 1F 04 23 00
15B0h: 5B 74 08 00 54 74 08 00 6E 20 69 00 04 00 0C 01
15C0h: 6E 10 94 00 01 00 0C 04 1F 04 05 00 69 04 09 00
15D0h: 62 04 09 00 38 04 3A 00 71 00 39 00 00 00 0C 03
15E0h: 62 04 09 00 6E 10 1E 00 07 00 0C 05 6E 30 34 00
15F0h: 43 05 62 04 08 00 6E 10 1E 00 07 00
1610h: 6E 30 AD 00 54 06 0C 04 71 10 9C 00 04 00 0A 04
1620h: 14 05 66 66 06 40 2E 04 04 05 3C 04 07 00 62 04
1630h: 09 00 6E 20 40 00 43 00 71 00 55 00 00 00 62 04
1640h: 09 00 6E 10 02 00 04 00 0E 00 0D 02 6E 10 9A 00
1650h: 02 00 12 04 69 04 09 00 28 BC 0D 04 28 EE 00 00
1660h: 0E 00 00 00 16 00 01 00 40 00 00 00 18 00 04 00
1670h: 02 01 37 61 01 37 69 00 02 00 01 00 01 00 00 00
1680h: 8F 4B 00 00 0D 00 00 00 6F 10 03 00 01 00 62 00
16A0h: 00 00 02 00 01 00 01 00 00 00 97 4B 00 00
```

Native Wrapper

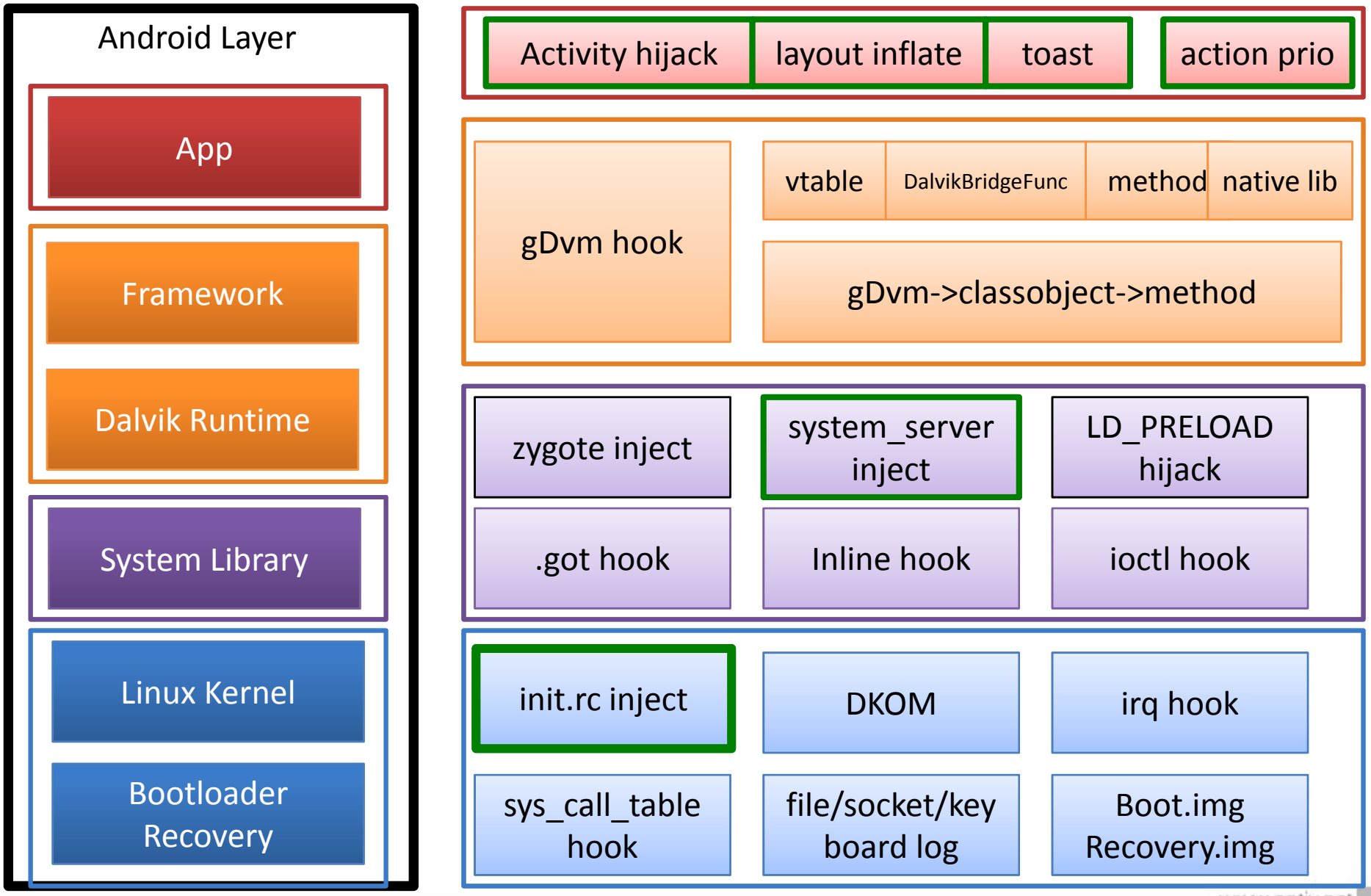




Native Wrapper&Opcode Recovery



Rootkit Roadmap



Rootkit—案例1

- sys_call_table
  - 读取

    /proc/kallsym

    /dev/mem

```
root@hammerhead:/ # cat /proc/kallsyms  | grep sys_call
c0107544 T sys_call_table
c02bd848 t proc_sys_call_handler
```

  - 写入

    orig_open = sys_call_table[__NR_OPEN]

    sys_call_table[_NR_OPEN] = my_open

    /dev/mem

    /dev/exynos-mem CVE-2012-6422

Rootkit—案例1

- 如果在ROM中植入ko
  - Android自带insmod命令
  - 第三方ROM有自定义的ko文件
- 系统调用HOOK技术
  - ARM下syscall

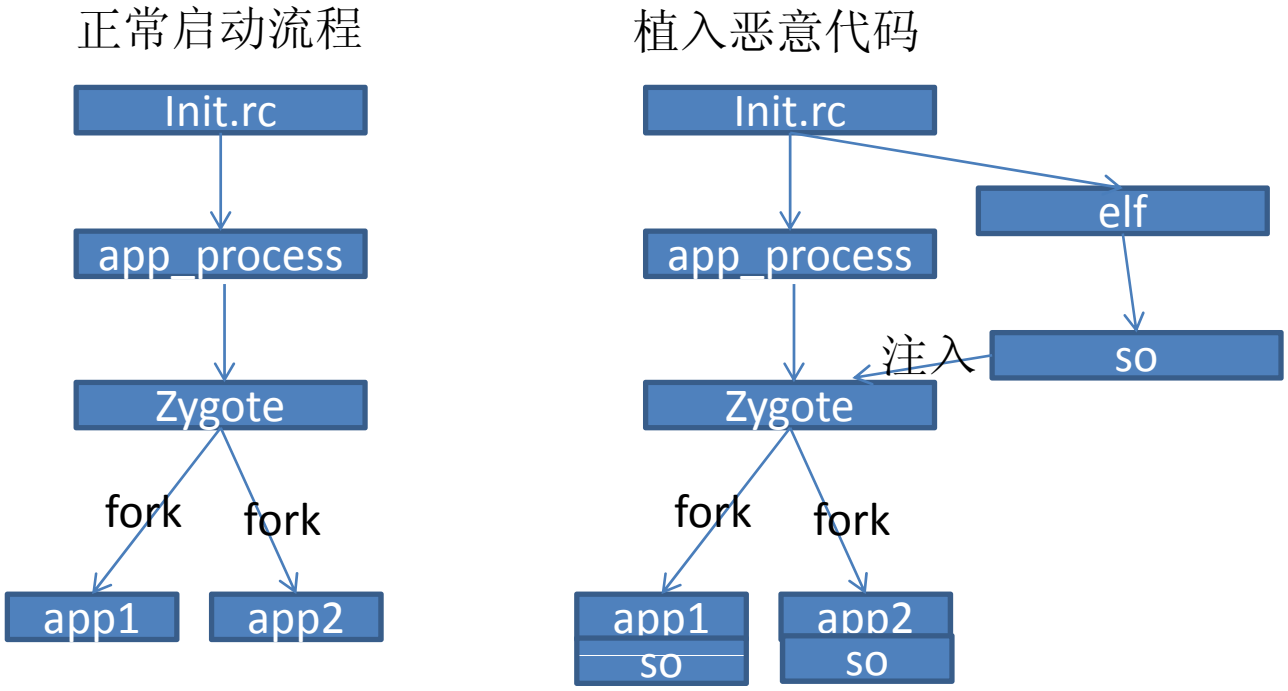
    mov r7, xxx

    svc 0
  - sys_call_table

sys call 调用	攻击目标
sys_read,sys_readv,sys_pread,sys_preadv	记录键盘输入
sys_open	隐藏文件
sys_ioctl	屏蔽内核消息

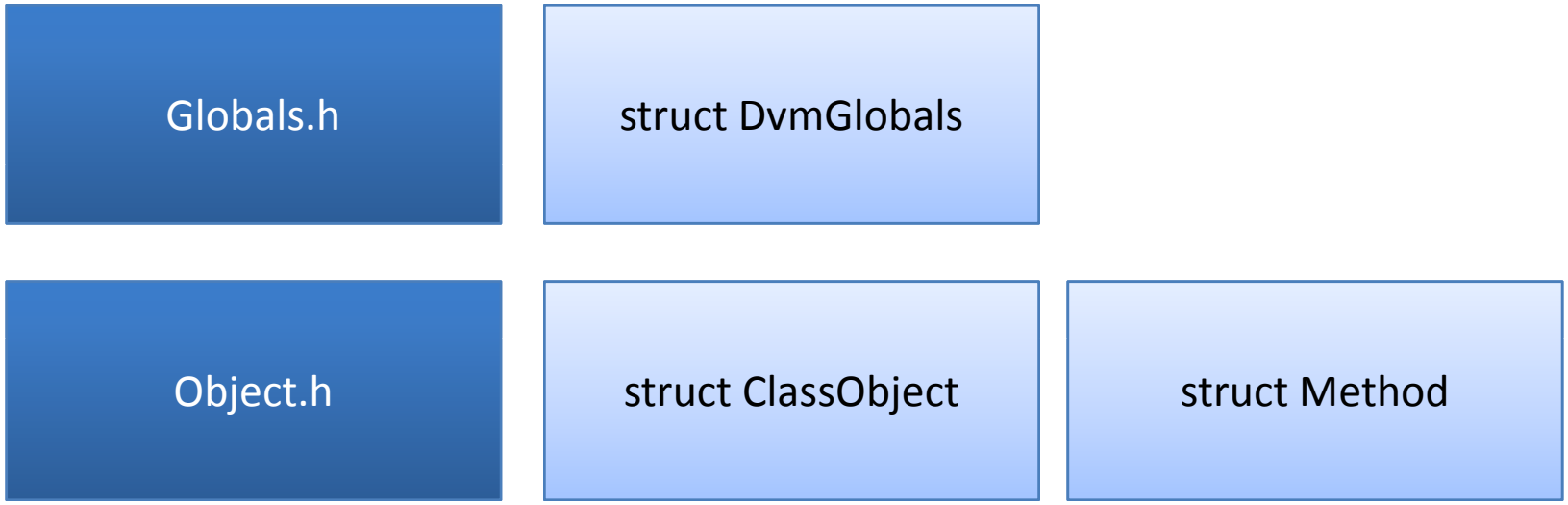
Rootkit—案例2

- 将ROM关键文件替换为恶意代码文件
- 启动时注入Zygote



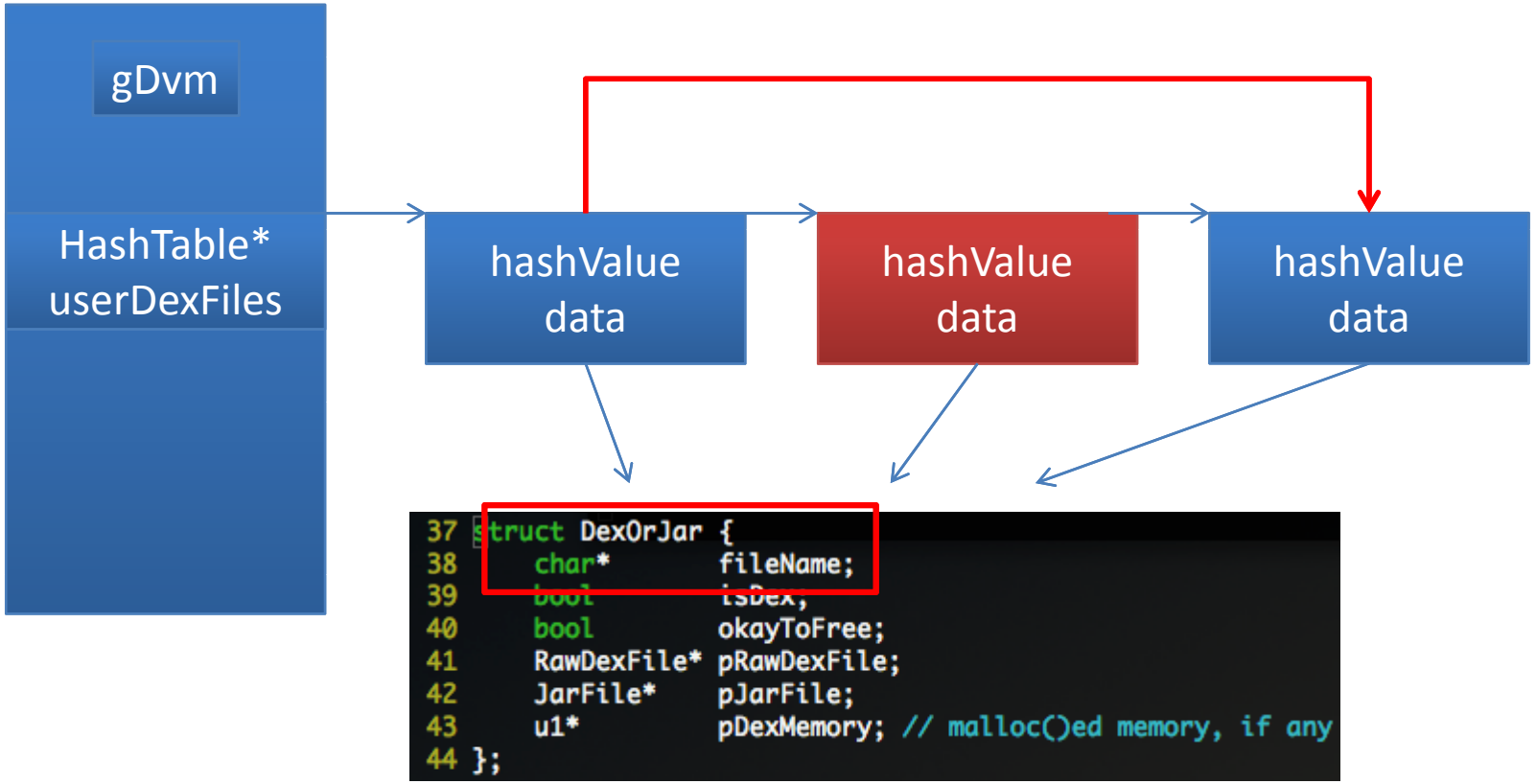
Dalvik VM隐藏技术

- see see gDvm



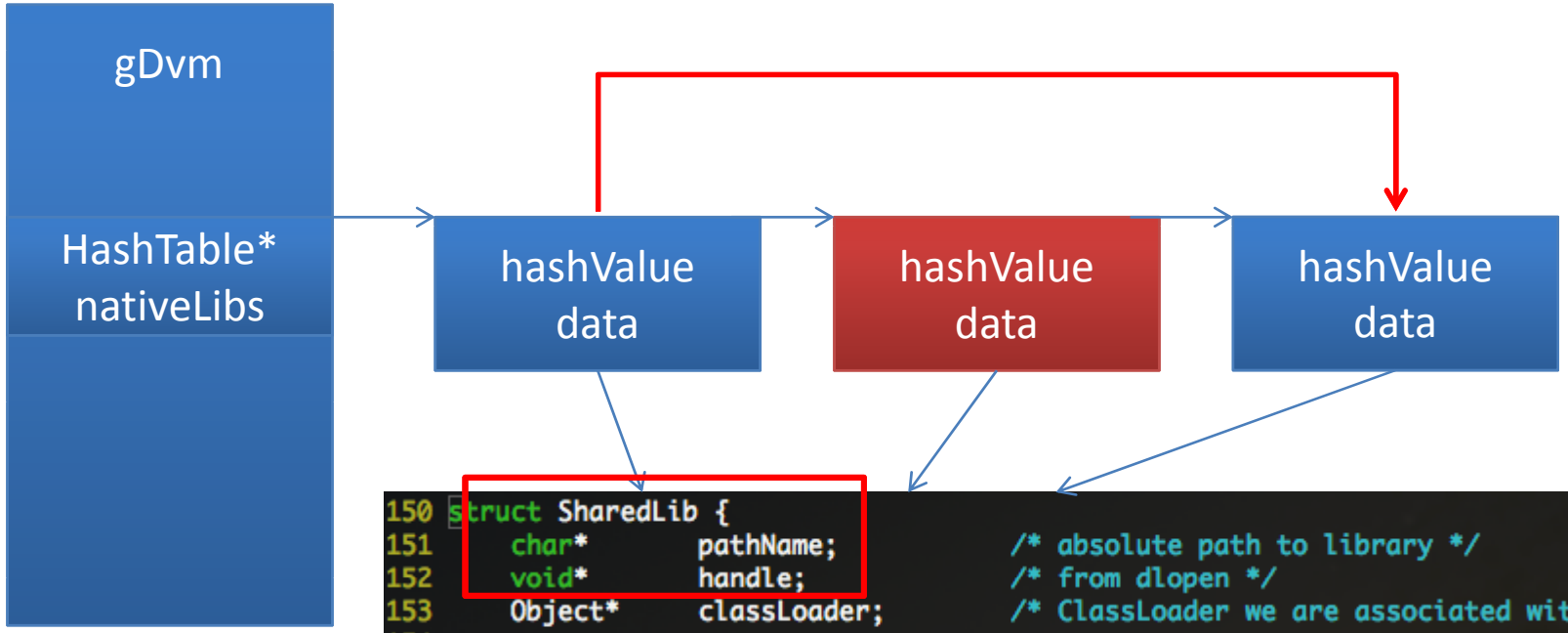
Dalvik VM隐藏技术

- 加载的DEX隐藏？



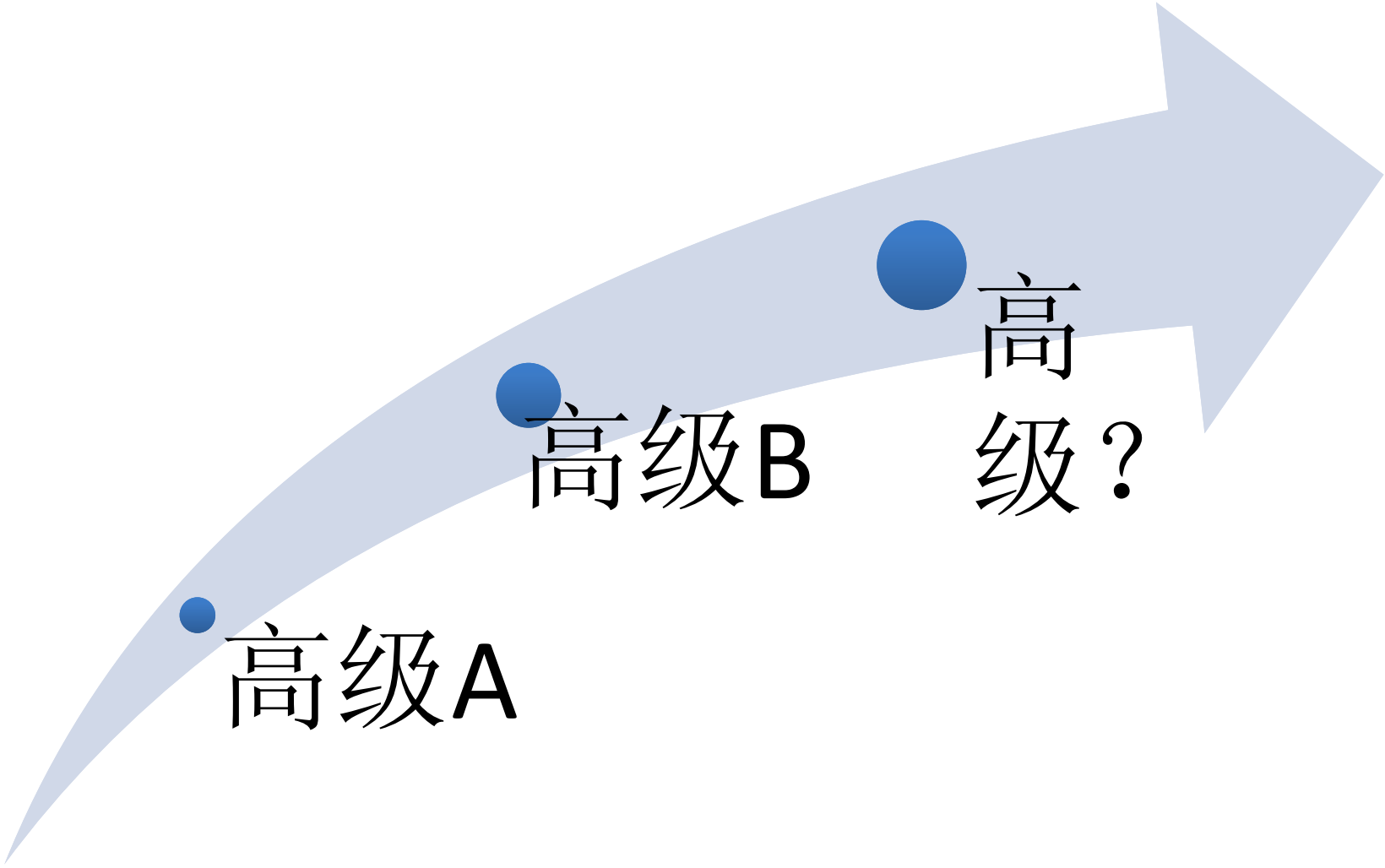
Dalvik VM隐藏技术

- 加载的LIB隐藏？



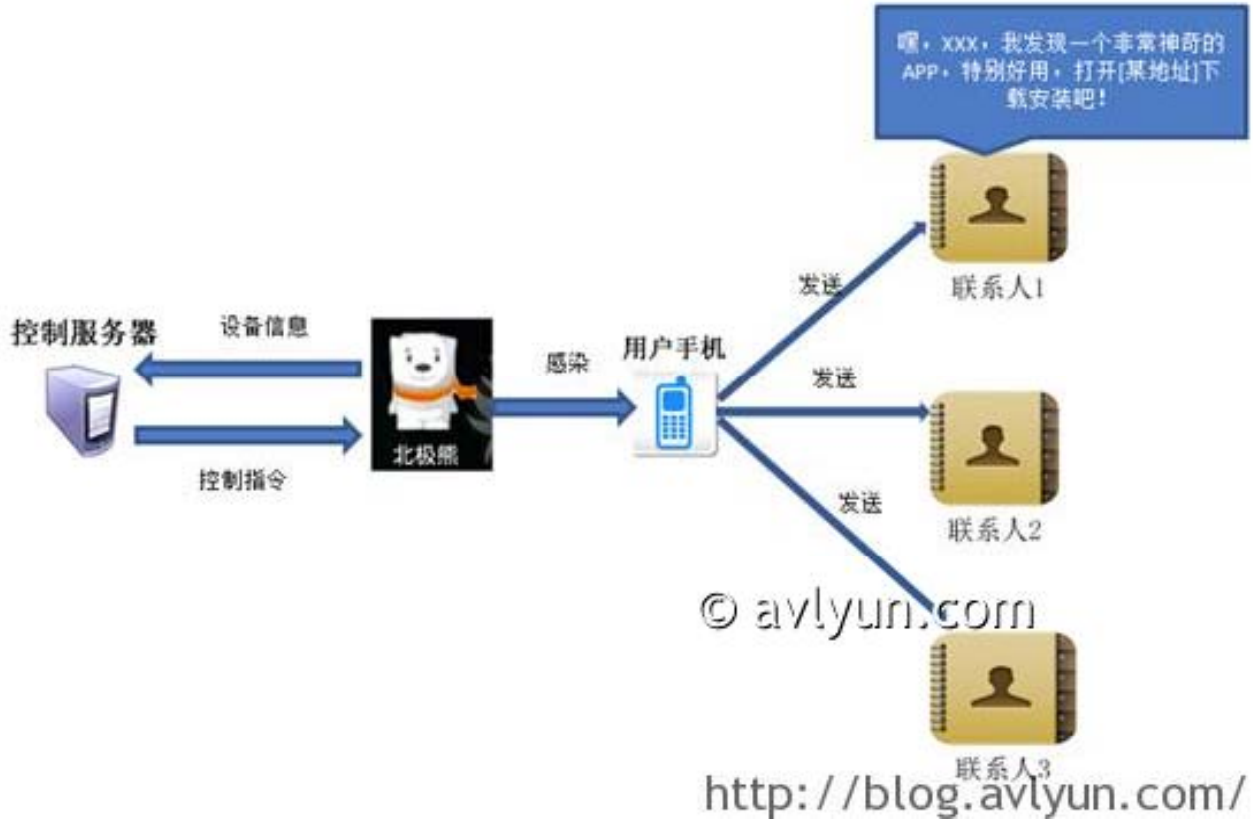

```
150 struct SharedLib {
151     char*       pathName;           /* absolute path to library */
152     void*       handle;             /* from dlopen */
153     Object*     classLoader;        /* ClassLoader we are associated with */
154
155     pthread_mutex_t onLoadLock;     /* guards remaining items */
156     pthread_cond_t  onLoadCond;     /* wait for JNI_OnLoad in other thread */
157     u4              onLoadThreadId; /* recursive invocation guard */
158     OnLoadState     onLoadResult;   /* result of earlier JNI_OnLoad */
159 };
```

高级高级技术



案例一 短信蠕虫

- “北极熊”
  - 获取短信联系人，遍历联系人发送诱骗短信

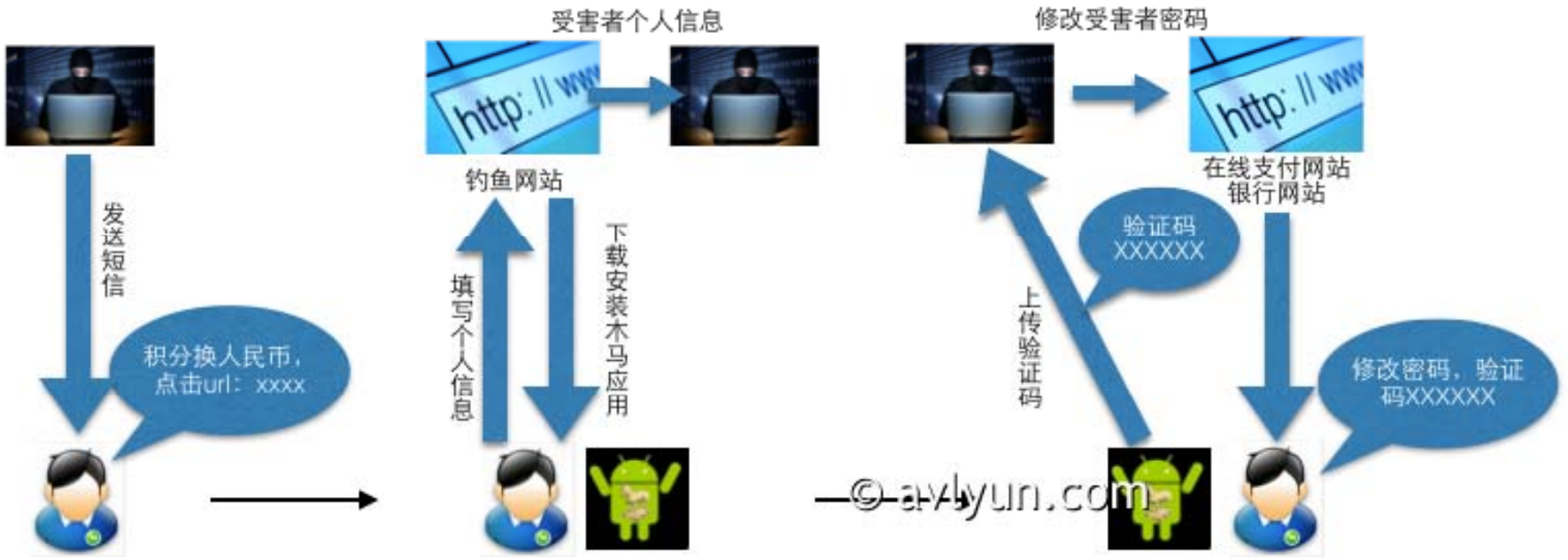


案例二 钓鱼木马攻击

- 钓鱼网站与木马结合

攻击场景一

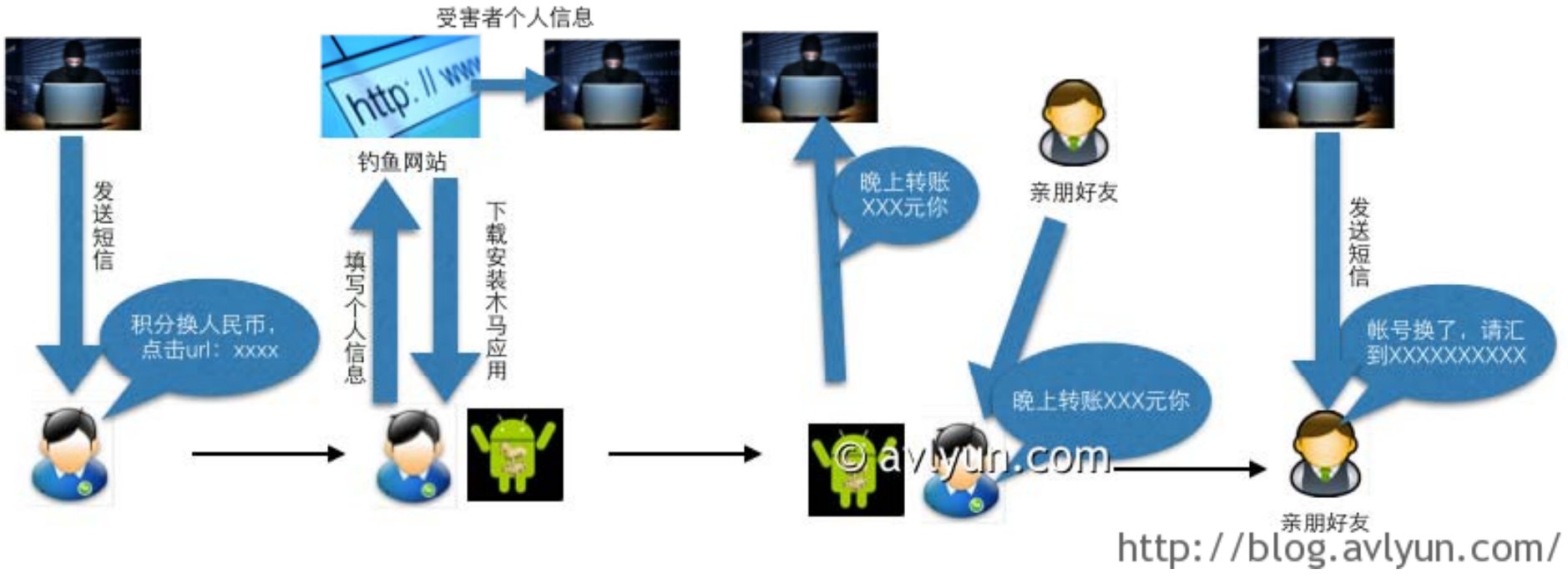
攻击者利用钓鱼网站在用户手机植入短信转发木马，攻击者修改用户在线支付平台密码，并拦截转发短信验证码



http://blog.avlyun.com/

攻击场景二

攻击者利用钓鱼网站在用户手机植入短信转发木马，攻击者获取受害者和其他人的短信内容，根据获取的短信内容伪造短信诱骗更多的用户



攻击场景三

- 2013年6月：
  - 在广州某电脑城购买三星Note II手机（大量预装软件）
- 2013年8月16日：
  - 在淘宝远程刷机
- 2014年3月15日：
  - 在海南航空官网购买机票，填入相关信息，开通工行快捷支付（是在海南航空官网直接开通么？（是的））），之后并无再使用快捷支付进行支付
- 2014年3月25日：
  - 收到网易邮箱验证码短信（修改密码验证码？（好像是））
- 2014年3月28日：
  - 手机所有短信被删除，通讯录被删除，高德导航的地图数据被删除
  - 工行快捷支付被异常转账，从28日开始持续4日。

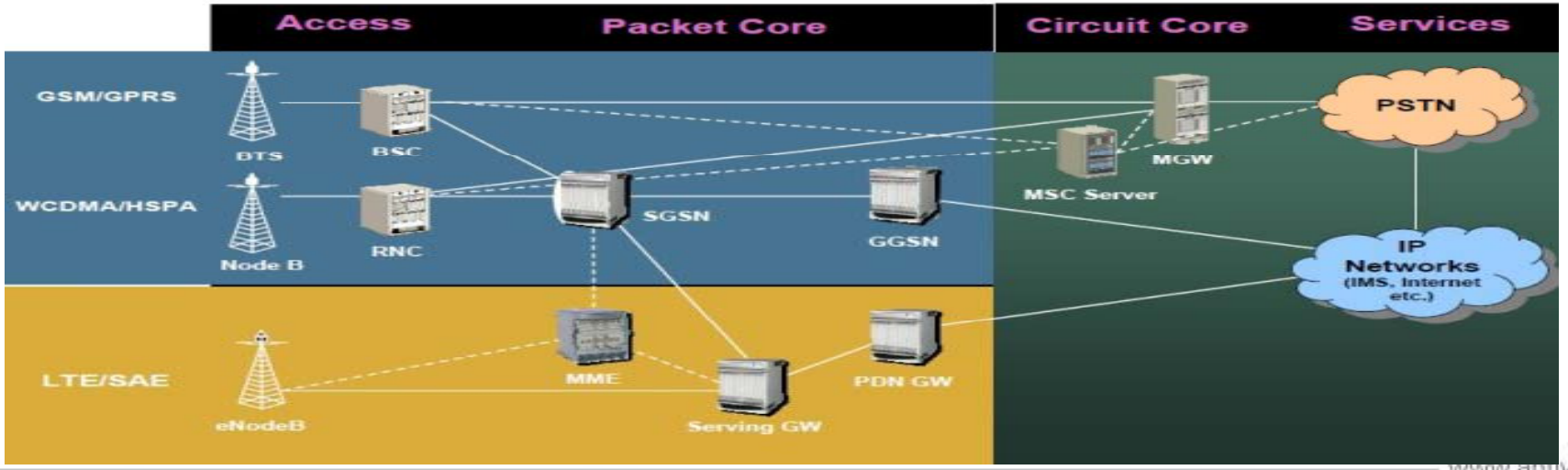
高级思维

技术对抗

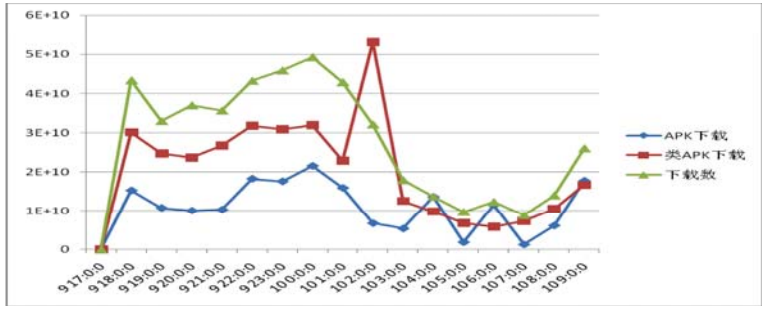
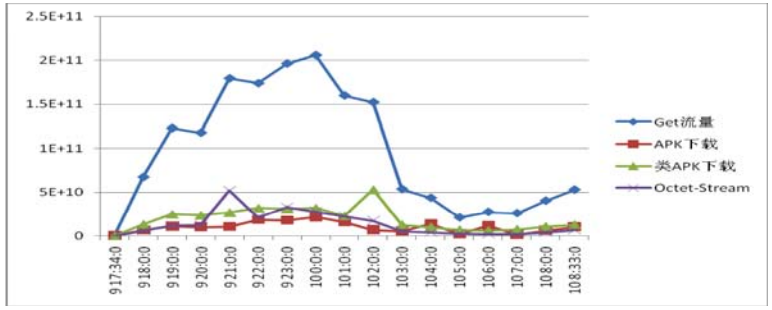
思想对抗

前瞻——4G时代网络安全技术变化

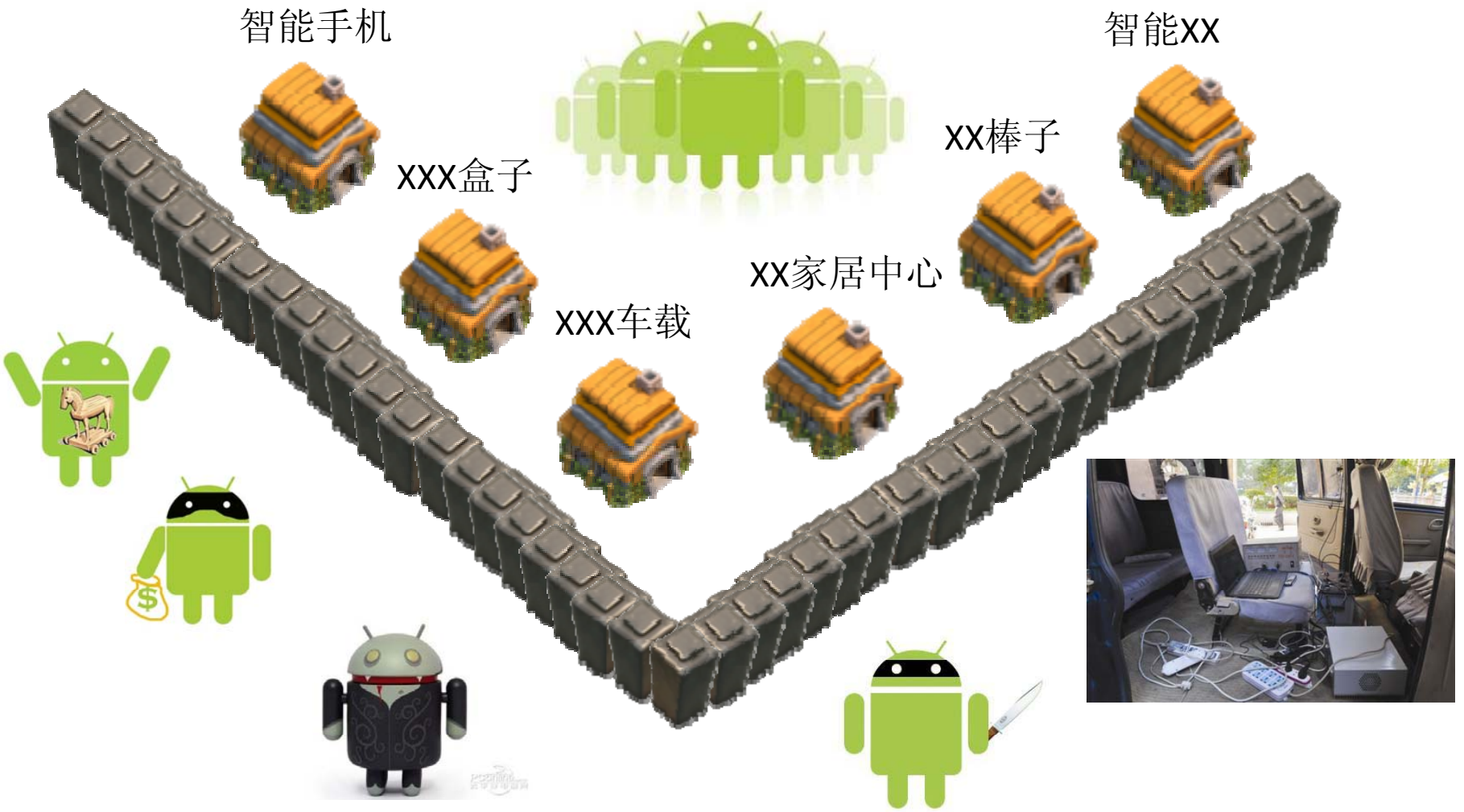
- 差异点
  - PDN GW/Serving GW，GGSN/SGSN
  - GTPv2，GTPv1
  - IPv6，全网数据融合为IP分组



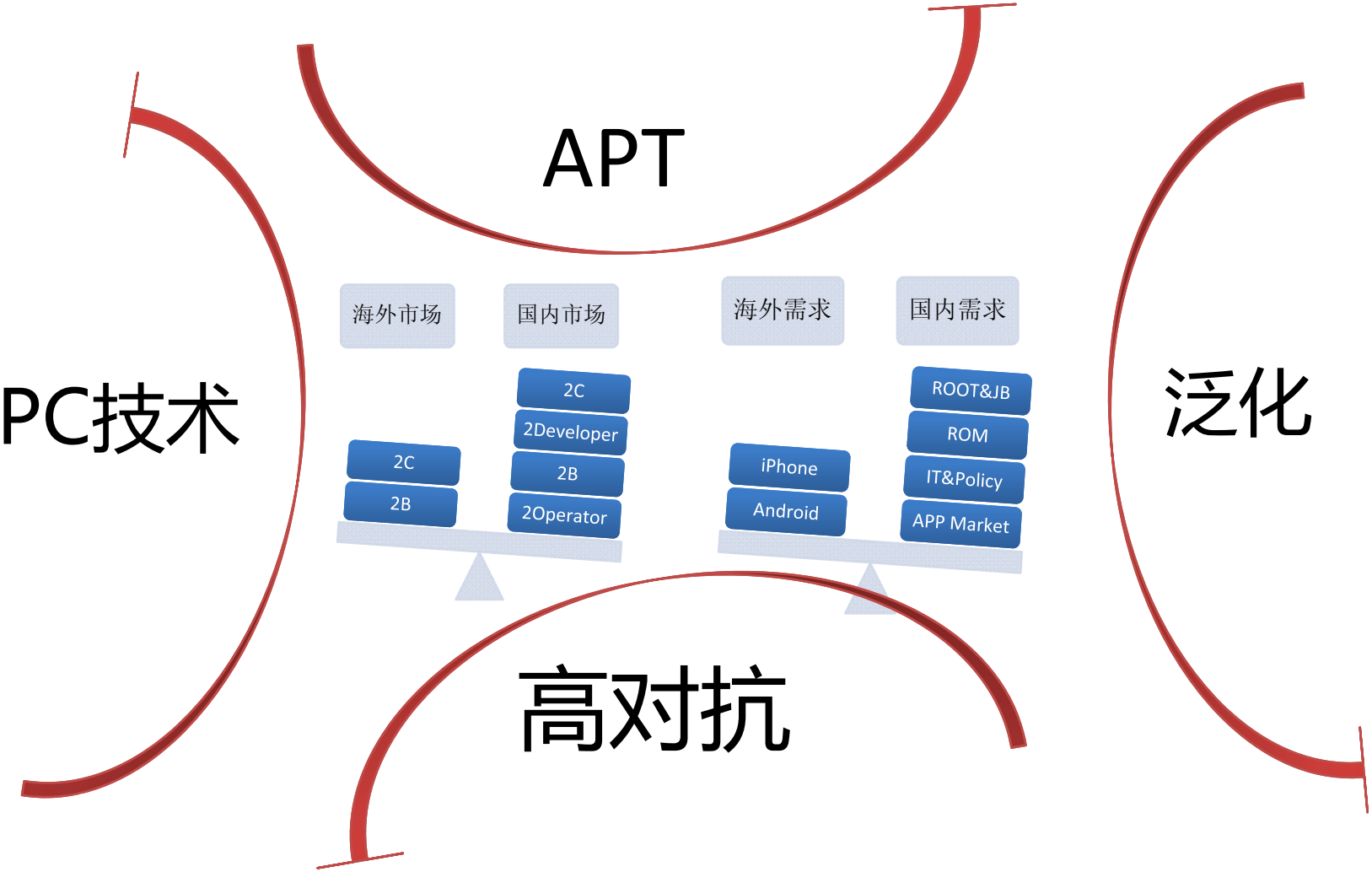
前瞻——4G时代，持续性加强，碎片增多

	2011	2012	2013
恶意代码行为	1k，2k 成功率一般	10几k 环境录音 短信为主	通话/环境录音，短信，位置，帐号
应用网络行为			
终端网络行为	50% <10	10% >50	>10,000

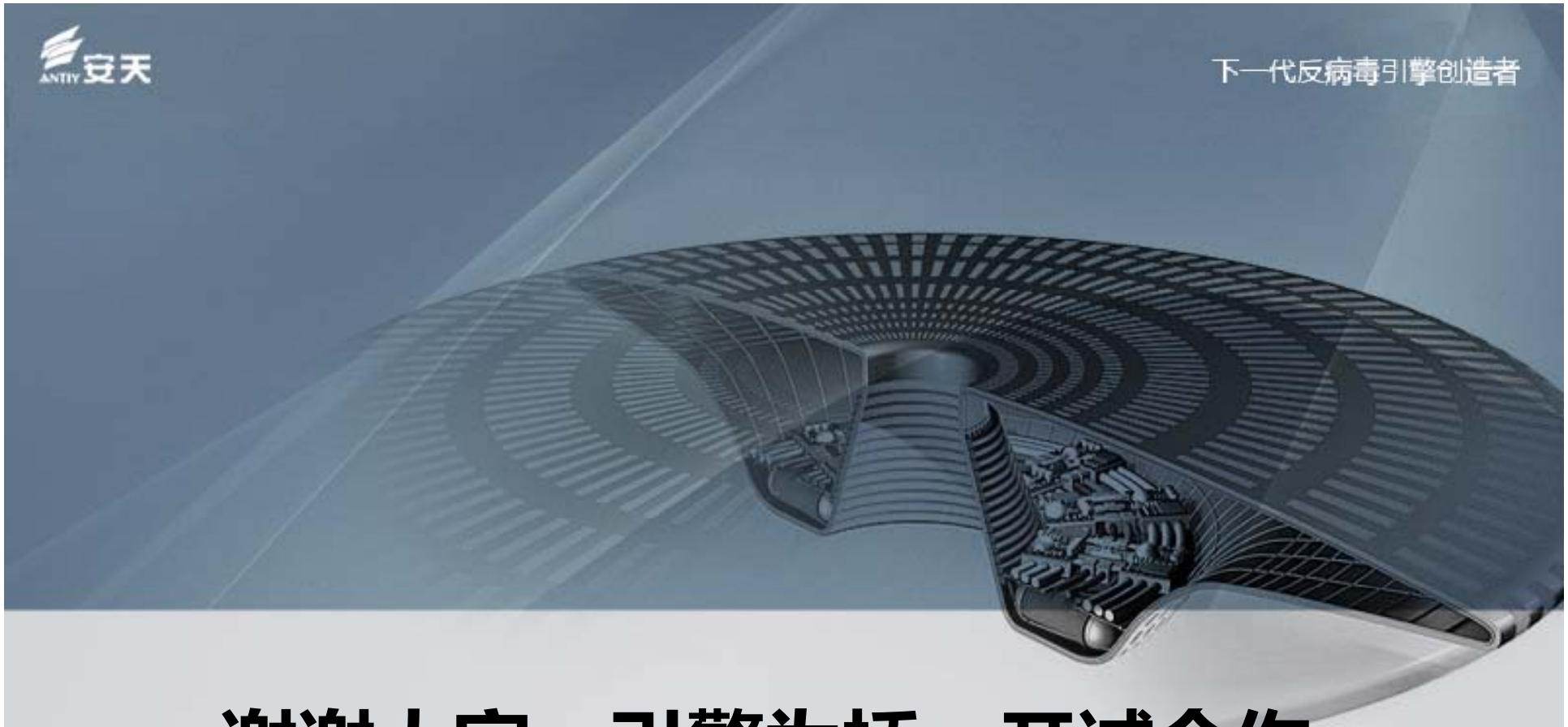
前瞻——移动安全防御链条不断延展和关联



前瞻——移动互联网威胁和安全的挑战



不是起跑线的问题，不能输在思想上



谢谢大家，引擎为桥，开诚合作

http://blog.avlyun.com
tompan@antiy.cn