KASPERSKY lab

It's Time for Heavy Weapons: Behavioral Detection on Android

Yury Slobodyanyuk@kaspersky.com
Roman Unuchek@kaspersky.com

Introduction

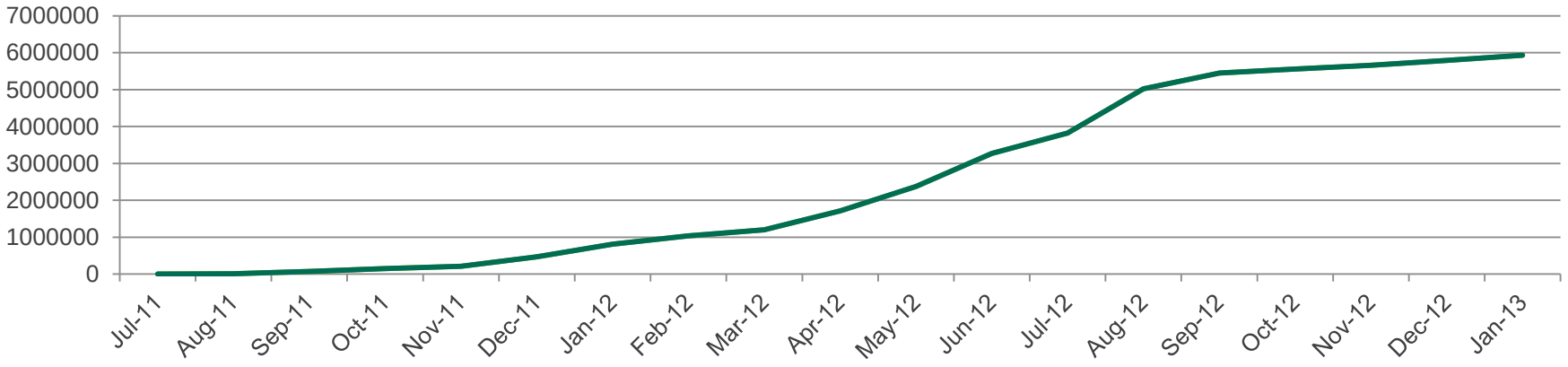
- Android popularity
- Current AV engines for android
  - Signature
  - Heuristics
  - Behavior
- Behavioral detection
  - Why we made it
  - How it works
- Behavior detection In action – working with real malware
  - Obfuscated malware detection
  - Decryption of encrypted files
  - Personal data leaks prevention

Total Number of Droid Phones Sold Worldwide

295,000,000[1]

- Android Phones activated each day 700,000
- Percent of market held by Droid Phones 36.7 %[1]

Total number of malware applications



1. http://www.statisticbrain.com/android-phone-statistics

KASPERSKY lab

Android AV engines

- **Signature:**

```
00000000: 18 E7 A0 B4 E8 A7 A3 E8 A1 A5 E4 B8 81 28 E6 BF ;  ↑з ґи§JиЎґдёЃ(жї
00000010: 80 E6 B4 BB E3 80 81 E5 8E BB E5 B9 BF E5 91 8A ;  Ђжґ»гЂЃеЋ»е№їе‘Љ
00000020: 29 E3 80 82 5B E6 B8 B8 E6 88 8F E5 85 94 E5 AD ;  )гЂ‚[жёёжЂЏе…”е-
00000030: 90 40 E6 88 91 E7 9A 84 E6 B8 B8 E6 88 8F 5D 00 ;  ђ@жЂ‘зљ„жёёжЂЏ]
00000040:                                                 ;
```

- **Heuristic:**

```
'DIRECTORY_DOWNLOADS' && 'fileURL' && 'softID'
&&      ('HAUzHAQDAgklAgoD' || 'GxkOExUURwkVHA5cCRUc')
&&      ('AA0fGDgFAQktHwc' || 'Gx4JPhMbFhUdPhsOGw')

->      DETECT("HEUR:Backdoor.AndroidOS.GinMaster.a")
```

- **Behavioral:**

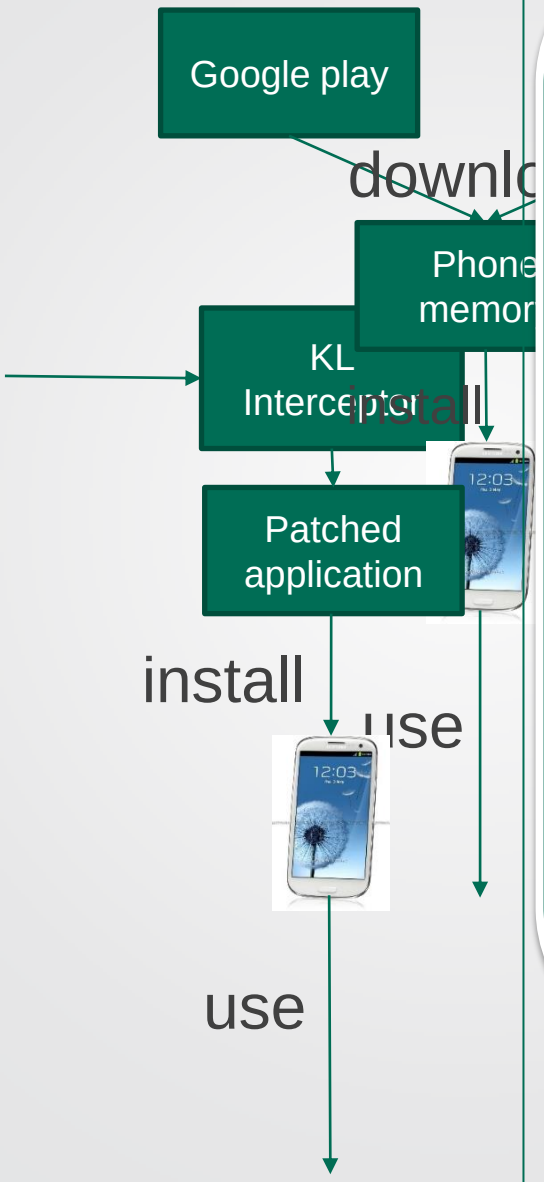
```
'getDeviceId' &&'getSubscriberId'&&'getLine1Number'
&& 'isConnectedOrConnecting' &&'getMethod("sendTextMessage")'
&& 'android.telephony.SmsManager.sendTextMessage'

➔ DETECT("BSS: Trojan-SMS.AndroidOS.Opfake.bo")
```

Why behavioral?

- We can stop executing application at any moment.
- Immune to obfuscation and reflection
- Detect new, never seen before threats
- Prevent apps from using “dangerous” functions
- Can stop exploiting vulnerabilities

How does it work?



Behavioral Log

android.app.Application.getSystemService("window")
com.kms.sandbox.SBActivity1.getSystemService("connectivity")
android.app.Application.getSystemService("connectivity")
android.app.Application.getSystemService("phone")
android.telephony.TelephonyManager.getSimOperator()
com.kms.sandbox.SBActivity1.getClass()
com.kms.sandbox.SBActivity1.getSystemService("connectivity")
android.app.Application.getSystemService("activity")
android.telephony.TelephonyManager.getDeviceId()
android.telephony.TelephonyManager.getNetworkCountryIso()
org.apache.http.message.BasicStatusLine.toString()
android.app.Application.getSystemService("location")
java.io.File.toString()
java.io.File.toString()
java.io.File.exists()
java.io.File.exists()
java.io.File.createNewFile()
android.app.ContextImpl$ApplicationPackageManager.getInstalledPackages(0)
java.io.FileOutputStream.write([B@406258b8)
com.kms.sandbox.SBActivity1.startActivity(Intent { act=android.service.wallpaper.LIVE_WALLPAPER_CHOOSER (has extras) })
com.kms.sandbox.SBActivity1.startActivity(Intent { cmp=com.livegame.wallpaperxingqiumj/com.kms.sandbox.SBActivity2 })`

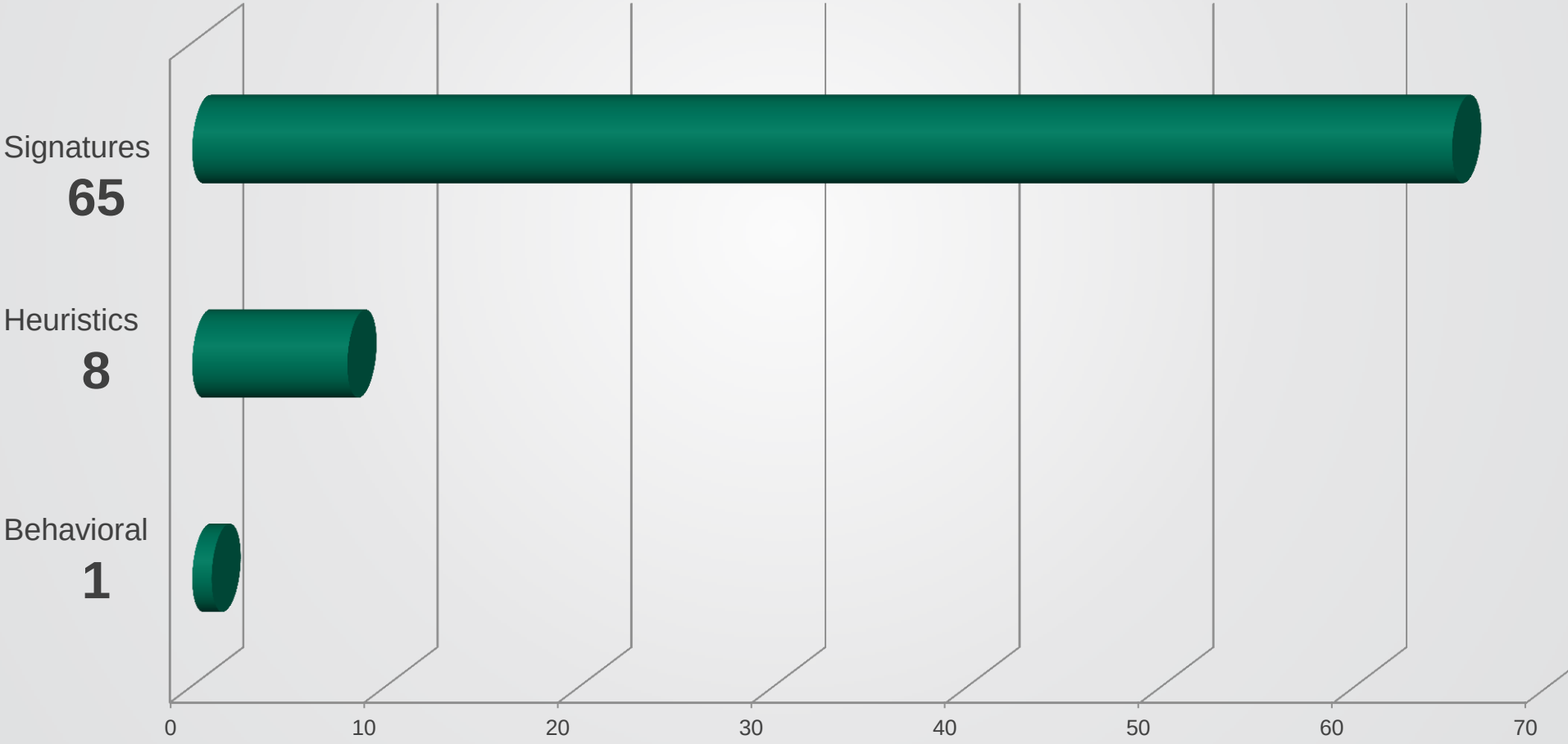
KASPERSKY lab

Proactive detection in action

Trojan-SMS.AndroidOS.Opfake

4% of malware installations (more than 2 500 in one month)

Number of records:



KASPERSKY lab

Trojan-SMS.AndroidOS.Opfake

Dex file before and after decompiling

```
String str2 = cogst.cogst.cogst("0oiFvqiwvPo");
Class[] arrayOfClass2 = new Class[1];
arrayOfClass2[0] = ckmoaxh.cogst;
Method localMethod1 = localClass2.getMethod(str2, arrayOfClass2);
cogst = "3";
cogst = "8";
cogst = "6";
cogst = "1";
String str3 = cogst.cogst.cogst("@v@i");
Class[] arrayOfClass3 = new Class[2];
arrayOfClass3[0] = ckmoaxh.ycbblfb;
arrayOfClass3[1] = Class.forName(cogst.cogst.cogst("TwywuqoP]E@iJujoJ"));
Method localMethod2 = localClass2.getMethod(str3, arrayOfClass3);
cogst = "7";
cogst = "3";
cogst = "7";
cogst = "6";
Method localMethod3 = localClass2.getMethod(cogst.cogst.cogst("5/R@vwt"), new Class[] { localClass3 });
cogst = "4";
cogst = "5";
```

Behavioral log:

```
...
java.lang.Class.getMethod("codePointAt", [Ljava.lang.Class;@405771a8)
java.lang.Class.getMethod("size", [Ljava.lang.Class;@4050b9c8)
java.lang.Class.getMethod("setData", [Ljava.lang.Class;@40724860)
...
```

Trojan-SMS.AndroidOS.Opfake (Detection)

```
java.lang.Class.getMethod("getSystem
java.lang.Class.getMethod("getDevice
android.app.Application.getSystemSer
android.telephony.TelephonyManager.getDeviceId()
java.lang.Class.getMethod("getSystem
java.lang.Class.getMethod("getSubscr
android.app.Application.getSystemSer
android.tel
java.lang.C
java.lang.C
android.app
android.tel
java.lang.C
com.kms.san
java.lang.C
java.lang.C
android.net.N
java.lang.Class.getMethod("addAction
java.lang.Class.getMethod("addDataAu
java.lang.Class.getMethod("addDataScheme", [Ljava.lang.Class;@4063eb18)
java.lang.Class.getMethod("registerR
java.lang.Class.getMethod("elementAt
java.lang.Class.getMethod("e", [Ljav
java.lang.Class.getMethod("getDefault", [Ljava.lang.Class;@405b6ec8)
java.lang.Class.getMethod("sendTextMessage", [Ljava.lang.Class;@40761c28)
android.telephony.SmsManager.sendTextMessage("770325", "null", "7223994", null, null)
java.lang.Class.getMethod("e", [Ljava.lang.Class;@406fff88)
java.lang.Class.getMethod("e", [Ljava.lang.Class;@4076ead8)
java.lang.Class.getMethod("sleep", [Ljava.lang.Class;@405b6528)
java.lang.Class.getMethod("getSystemService", [Ljava.lang.Class;@407256a8)
com.kms.sandbox.SBActivity1.getSystemService("connectivity")
java.lang.Class.getMethod("getActiveNetworkInfo", [Ljava.lang.Class;@405a59d8)
```

android.telephony.TelephonyManager.getDeviceId()

android.telephony.TelephonyManager.getSubscriberId()

tLine1Number()

rConnecting()

java.lang.Class.getMethod(“sendTextMessage”

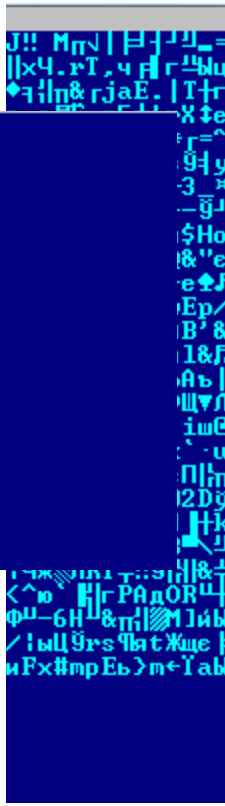
android.telephony.SmsManager.sendTextMessage

DETECTED:

BSS:*Trojan-SMS.AndroidOS.Opfake*

Trojan-SMS.AndroidOS.Opfake DECRYPTION

```
java.io.FileOutputStream.write(99)
java.io.FileOutputStream.write(101)
java.io.FileOutputStream.write(105)
java.io.FileOutputStream.write(118)
java.io.FileOutputStream.write(101)
receiveSMSNumber=770056
receiveSMSText=
appId=76
apiKey=mapk01_20f
firstStart=false
shortcutIdList=17301583
nextTime=1355921354808
data=1355917937466,1;1355917937594,1;1355917937787,1;1355919689684,1;1355919689893,1;1355919690143,1;
receiveSMSNumber=770056
receiveSMSText=
appId=76
apiKey=mapk01_20f
firstStart=false
shortcutIdList=17301583
nextTime=1355921354808
data=1355917937466,1;1355917937594,1;1355917937787,1;1355919689684,1;1355919689893,1;1355919690143,1;
java.io.FileOutputStream.write(55)
java.io.FileOutputStream.write(48)
java.io.FileOutputStream.write(48)
java.io.FileOutputStream.write(53)
java.io.FileOutputStream.write(54)
java.io.FileOutputStream.write(10)
java.io.FileOutputStream.write(114)
java.io.FileOutputStream.write(101)
java.io.FileOutputStream.write(99)
java.io.FileOutputStream.write(101)
java.io.FileOutputStream.write(105)
java.io.FileOutputStream.write(118)
java.io.FileOutputStream.write(101)
java.io.FileOutputStream.write(83)
```

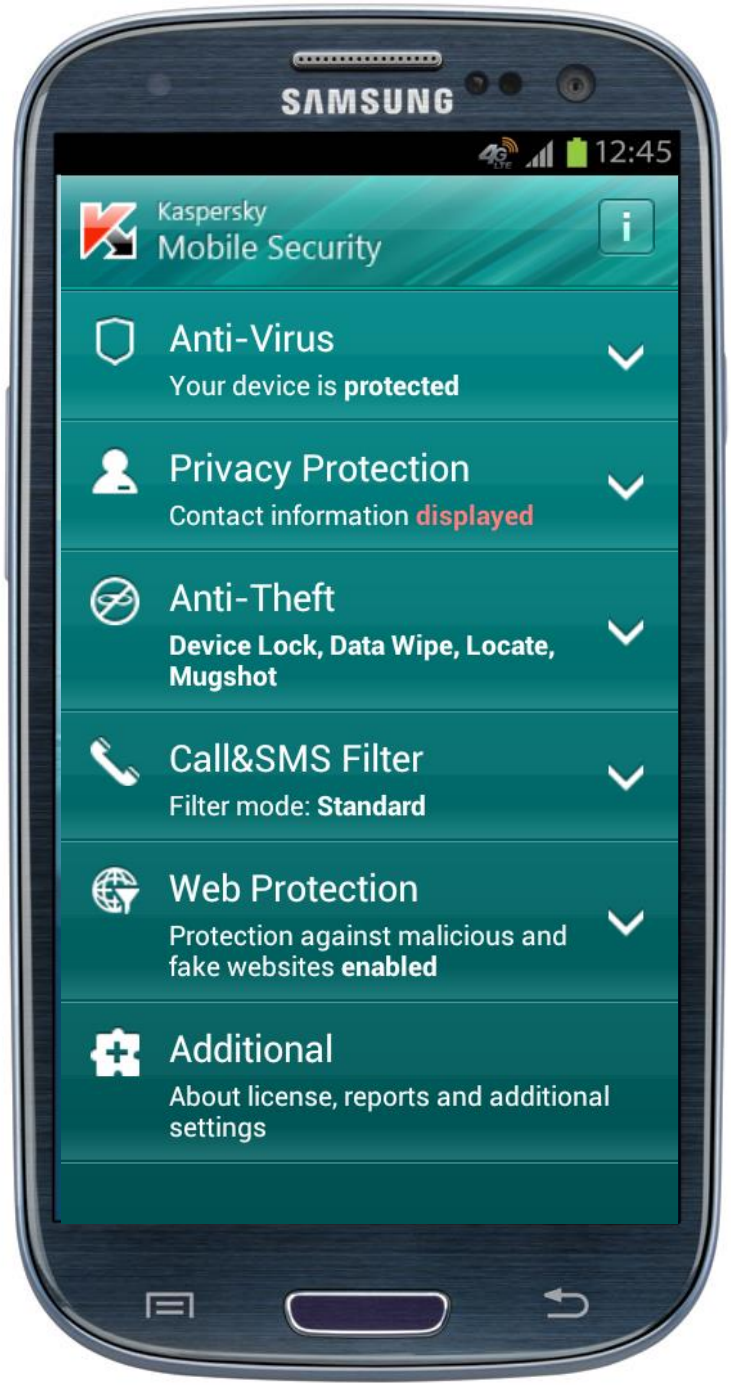


Clean app - personal data leak prevention

Application behavior: **Behavioral engine:**

Behavior log:

```
javax.crypto.Cipher.getInstance("AES/CBC/PKCS5Padding")
javax.crypto.Cipher.init(1, javax.crypto.spec.SecretKeySpec@3ac)
javax.crypto.Cipher.getParameters()
javax.crypto.Cipher.doFinal([B@40792ba0)
javax.crypto.Cipher.getInstance("AES/CBC/PKCS5Padding")
javax.crypto.Cipher.init(2, javax.crypto.spec.SecretKeySpec@3ac, javax.crypto.spec.IvParameterSpec@407a2
javax.crypto.Cipher.doFinal([B@407a0d70)
java.io.File.toString()
java.io.File.exists()
com.kms.sandbox.SBActivity4.getSystemService("phone")
com.kms.sandbox.SBActivity1.getSystemService("phone")
android.telephony.TelephonyManager.getDeviceId()
android.telephony.TelephonyManager.getSubscriberId()
android.telephony.TelephonyManager.getSimSerialNumber()
android.telephony.TelephonyManager.getLine1Number()
android.telephony.TelephonyManager.getDeviceId()
android.telephony.TelephonyManager.getSubscriberId()
android.telephony.TelephonyManager.getSimSerialNumber()
android.telephony.TelephonyManager.getLine1Number()
java.lang.Class.getMethod("get", [Ljava.lang.Class;@407ce028)
java.lang.Class.getMethod("get", [Ljava.lang.Class;@407cdbe8)
com.kms.sandbox.SBActivity4.getSystemService("connectivity")
com.kms.sandbox.SBActivity1.getSystemService("connectivity")
```


SAMSUNG
12:45
Kaspersky
Mobile Security
Anti-Virus
Your device is protected
Privacy Protection
Contact information displayed
Anti-Theft
Device Lock, Data Wipe, Locate, Mugshot
Call&SMS Filter
Filter mode: Standard
Web Protection
Protection against malicious and fake websites enabled
Additional
About license, reports and additional settings

Thank You!

Yury.Slobodyanyuk@kaspersky.com
Roman.Unuchek@kaspersky.com