

# Building a Test Environment for Android Anti-Malware Tests

**Hendrik Pilz**

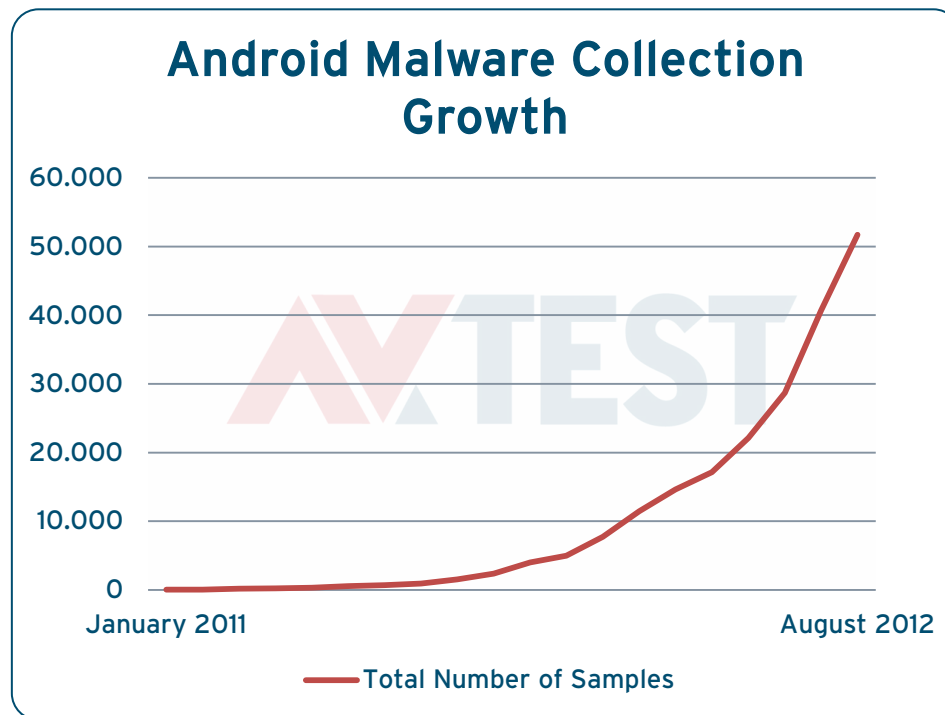
**Director Technical Lab / Mobile Security**

**[hpilz@av-test.de](mailto:hpilz@av-test.de)**

# Agenda

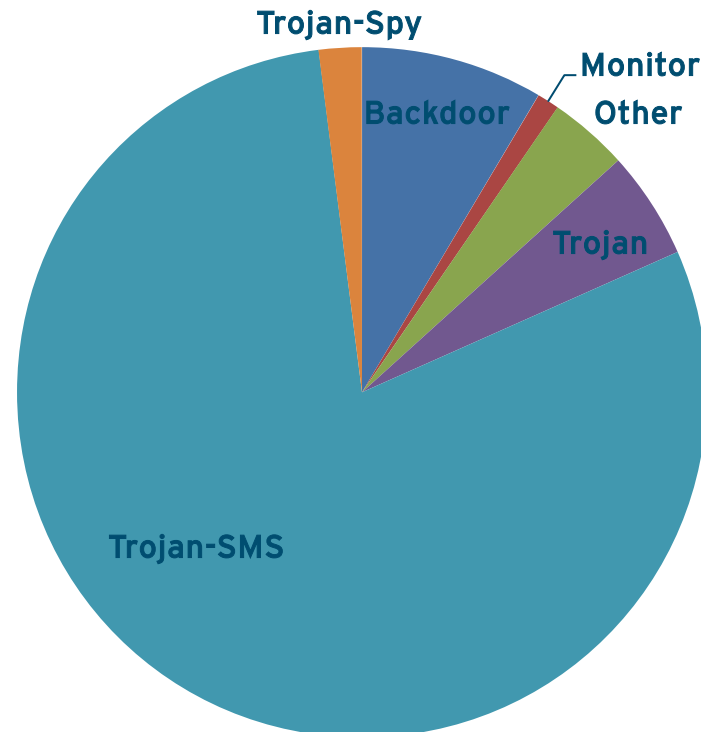
- **Android Malware Landscape**
- **Real Devices or Emulator?**
- **Preparation**
- **Test Scenarios**
- **Automation**
- **Problems**

# Android Malware Landscape



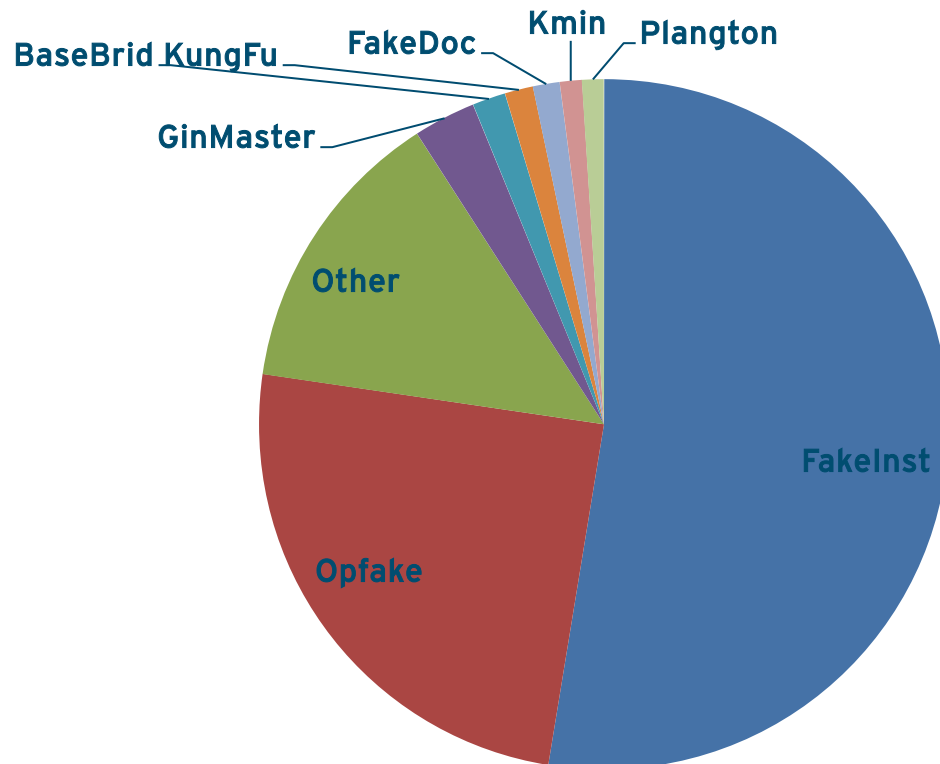
# Android Malware Landscape

Malware Categories August 2012



# Android Malware Landscape

Malware Families August 2012



# Real Devices or Emulator

## Device

- Real user experience
- App activation via SMS
- Real life environment

## Emulator

- Cost efficient, scalable
- Root privileges
- Multiple API versions and hardware configurations
- Snapshots

# Preparation

## System Requirements:

- **PC which is capable to run the Android SDK**
- **Android device, prepaid SIM**
- **USB cable**
- **WiFi-Internet for Android device**

# Preparation





# Preparation

- **Install Android SDK from [developer.android.com/sdk](https://developer.android.com/sdk)**
- **Choose Malware Samples according to AMTSO Guidelines**
- **Install Anti-Malware on test device, update signatures**

# Preparation

- **Connect device to PC**

- **Create device backup**

```
$: adb backup -f <file> -apk -shared -all -system
```

```
$: adb restore <file>
```

- **Take Screenshots**

```
$: android-sdk/tools/ddms
```

# Test Scenarios - On-Demand Scan

- **Copy samples to device**  
`$: adb push <source> /sdcard/samples`
- **Perform on-demand scan, delete all malicious files**
- **Save remaining files**  
`$: adb pull /sdcard/samples <dest>`
- **Save scan reports, if possible**

# Test Scenarios - On-Demand Scan

**Alternative to `adb push/pull`:**

**Copy files over WiFi from/to network share (e.g. with Astro File Manager)**

# **Test Scenarios - On-Demand Scan**

**Some Anti-Malware apps scan installed apps only!**

**An On-Access Test is always required to determine accurate detection rates!**

# Test Scenarios - On-Access

- **Install each sample one-by-one**  
`$: adb install <apk-file>`
- **Check warnings and messages of Mobile Security**
- **Remove or uninstall sample**  
`$: adb uninstall <package-name>`

# Test Scenarios - On-Access

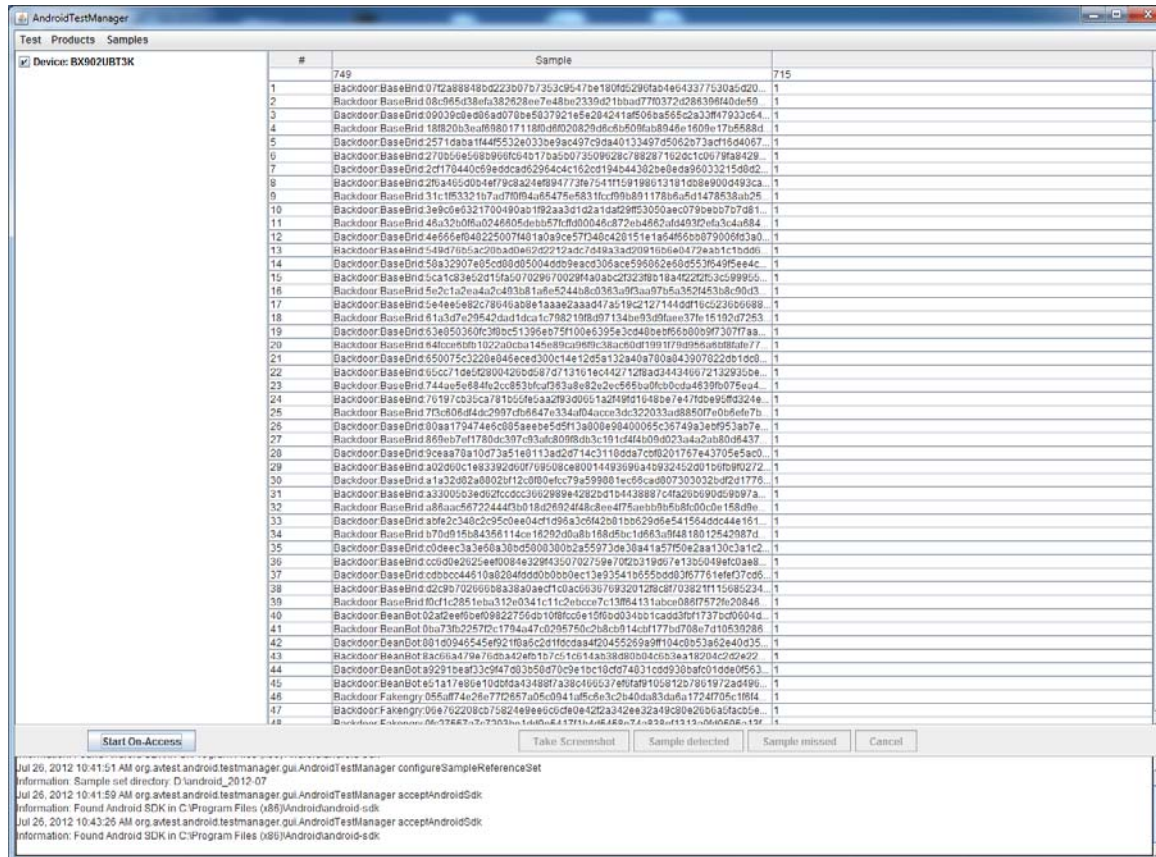
```
on-access.sh x
#!/bin/bash
c=1
# traverse sample directory
for i in `ls $1`
do
    sample="${1}/${i}"
    echo "${c}: Installing ${i}"
    # get android package name
    package=`adb dump badging ${sample} |\
grep package: | sed "s/package: name= '//" |\
sed "s/' versionCode.*$//"`
    echo "Package: ${package}"
    # install, enter result, uninstall
    adb install ${sample}
    echo "Was the sample detected? (1 - yes / 0 - no)"
    read result
    echo -e "${sample}\t${result}" >> "on_access_report.txt"
    adb uninstall $package
    let c=c+1
done
```

# Test Scenarios - On-Access

```
android@android-VirtualBox:~$ on-access.sh /media/android/  
1: Installing 00a9677cd438dd8b3b3320ad45562d409b929e33587ed7211356d40477725dfc.apk  
Package: com.keji.danti34  
* daemon not running. starting it now on port 5037 *  
* daemon started successfully *  
415 KB/s (923684 bytes in 2.168s)  
  pkg: /data/local/tmp/00a9677cd438dd8b3b3320ad45562d409b929e33587ed7211356d40477  
725dfc.apk  
Success  
Was the sample detected? (1 - yes / 0 - no)  
1  
Success  
2: Installing 0153e70bb3573e3fa701e307ae8a5b1b5d2ad72e2339b75acf5770cfda0c9a60.apk  
Package: com.keji.Graphisa  
422 KB/s (10997244 bytes in 25.399s)  
  pkg: /data/local/tmp/0153e70bb3573e3fa701e307ae8a5b1b5d2ad72e2339b75acf5770cfda  
0c9a60.apk  
Success  
Was the sample detected? (1 - yes / 0 - no)  
1
```



# Test Scenarios - On-Access



#	Sample	Count
749	Sample	715
1	Backdoor BaseBnd 072a98918b0d223b07b352c9547be180f5290ab44e43377530a5d20...	
2	Backdoor BaseBnd 08f6d53ba6382528e7e43b2233d21bbw7703372d8830840de59...	
3	Backdoor BaseBnd 09030cbe986ad070be5837921e5e204241af505ba565c2a32047933c64...	
4	Backdoor BaseBnd 18f820b3eaf688017118b0d020829d8dc680fab849e1608e17b5688d...	
5	Backdoor BaseBnd 2571daba1f44f532e033be9ac497c9da4013349750562b73ac1f64d067...	
6	Backdoor BaseBnd 270b6e56b96f0c4b17bac0073509028c7882817620c1c0678fa8429...	
7	Backdoor BaseBnd 2cf178440c59e9dca02964c4c162cd194b444302be8e9d990332154b9e2...	
8	Backdoor BaseBnd 2b64500546796324e894773675411f191980131028e900493ca...	
9	Backdoor BaseBnd 31c1f83301b7ad7f00f4b5476f6831fc09b081178ba6fd1478538ab05...	
10	Backdoor BaseBnd 3e9c0e321700490ab192aa31d2a1da2f9f30505ac079b9e707d81...	
11	Backdoor BaseBnd 48a320f0a0246605deb57cfd00046c872b4862af49302efac3ca884...	
12	Backdoor BaseBnd 4e565ef0462250074801a0a9ce57f346c426151e1a456eb079006f3a0...	
13	Backdoor BaseBnd 549d7b55ac20ba0a6e22212adc7d48a3a2d99105e6a479eab1c1b0d6...	
14	Backdoor BaseBnd 58a32907e3cc09095004d099ac308ace59052e505530649f5e4c...	
15	Backdoor BaseBnd 5ca163960d11a407029f70028f9a0ac0c3236918a4022f3c3c9995...	
16	Backdoor BaseBnd 5e2c1a2e402453b1a6e5244b0c383a0f3aa97b5a3501453b89d3...	
17	Backdoor BaseBnd 5e4e6582c78046ab0e1aae2aaad47a519c2127144dd16cc52360688...	
18	Backdoor BaseBnd 61a3d7e29542dad1dca1c7982108b0d7134be93d0fane37e15182d7253...	
19	Backdoor BaseBnd 63e950260c30bc51396eb75f100e6395e3cd40ebf656009f73077aa...	
20	Backdoor BaseBnd 640cc0b01022a0c3a14e89fca09f38ac60d1991f79d956a508de77...	
21	Backdoor BaseBnd 650075c220e048ce4300c74e12e5f13240a70b0439070220b16d...	
22	Backdoor BaseBnd 65cc718e5280042b0d587d713161ec427128ad241436672132935d...	
23	Backdoor BaseBnd 744ae5e884c2c83bfc1833a8e82ce565ba96b0cda483fb075ea4...	
24	Backdoor BaseBnd 76197cb3ca781b5569ca2930051a2d49d1048be7e47db9e9fd324e...	
25	Backdoor BaseBnd 73c806d84d2997cb6647e33a1d04acca3d322033a88807070befe7b...	
26	Backdoor BaseBnd 80aa179474e5c95aee5e5f13a00e9840005c36749a2eb953ab7e...	
27	Backdoor BaseBnd 8f6b7a11780ac307c8a6d08bd3c19144ab0b073a4a20a0b0437...	
28	Backdoor BaseBnd 9cea78a10d73a51e8113ae20714c31180da7c06201767443705e5ac0...	
29	Backdoor BaseBnd a02000c1e83392608769508ca80014192690a4092452d01b0f80272...	
30	Backdoor BaseBnd 1a32402e8002f12c0f0efc79e599001ec69ca80730032bdf2d1776...	
31	Backdoor BaseBnd a33005c3e0d2fccc0362989e4282b1d4438887c4fa20b690d9097a...	
32	Backdoor BaseBnd a8faaa567224443b018d26024f8c8ee475aebb9b6f8f000de158d9e...	
33	Backdoor BaseBnd a0fe2c340c2c50ee4d01d95a3c942b81b062995e5415540c44e161...	
34	Backdoor BaseBnd b70d1f5a435f114a162920ba18b65a16683a4818312542087d...	
35	Backdoor BaseBnd c0deec3a3e68a30b5f0003002a5f973de30a4185750e2aa10c31c2...	
36	Backdoor BaseBnd cc02e205ee0084e328f350702759e70f2b19907e11305049efc0a8...	
37	Backdoor BaseBnd cdbcc44610a8204d5d0b0b0cc13e93541b655b0d3967761efef37c05...	
38	Backdoor BaseBnd d2c9b702606b8a38a0aefc1c0ac063070932012828703821111685234...	
39	Backdoor BaseBnd fd1c2851eba312e0341c11c2ebcc7c13964131ab0e08f7572620848...	
40	Backdoor BaseBot 02a2eef0e0822750d1080c9e15960b31001ca03801137c0d0964...	
41	Backdoor BaseBot 0b77b2572c1776a47c039676cb0b01144f177d70871f1050386...	
42	Backdoor BaseBot 881d045545ef921f0a5c2d1f0ccaa4f2045525e99f104c0b53a5e4d035...	
43	Backdoor BaseBot 8ac08a479e76db424fb1b7c51c514a383808040c3aa18204c2d2e22...	
44	Backdoor BaseBot a9291beaf3c94f033e56870c9e1bc16cd74631cd9930baf01d5e0563...	
45	Backdoor BaseBot e51a17e80e19d0da43498f7a38c466537e9f0a9105812b7801972a499...	
46	Backdoor Fakegry 085aff74c26e772687a05c0941af5c5e3cb40da83d86a1724f05c1894...	
47	Backdoor Fakegry 09e10220c0168e0cc0e04c22a342ae32a4820a203645c0c8...	
48	Backdoor Fakegry 0e176e673710331440e417114e483e7a15381931310646a6e15f...	

Jul 26, 2012 10:41:51 AM org.avtest.android.testmanager.gui.AndroidTestManager configureSampleReferenceSet  
 Information: Sample set directory: D:\android\_2012-07  
 Jul 26, 2012 10:41:59 AM org.avtest.android.testmanager.gui.AndroidTestManager acceptAndroidSdk  
 Information: Found Android SDK in C:\Program Files (x86)\Android\android-sdk  
 Jul 26, 2012 10:43:26 AM org.avtest.android.testmanager.gui.AndroidTestManager acceptAndroidSdk  
 Information: Found Android SDK in C:\Program Files (x86)\Android\android-sdk

# Test Scenarios - False Positives

- **Combination of OA & OD**
- **Install clean apps via ADB**
- **Run an OD-scan afterwards**
- **Note all warnings and detections**

# Test Scenarios – False Positives

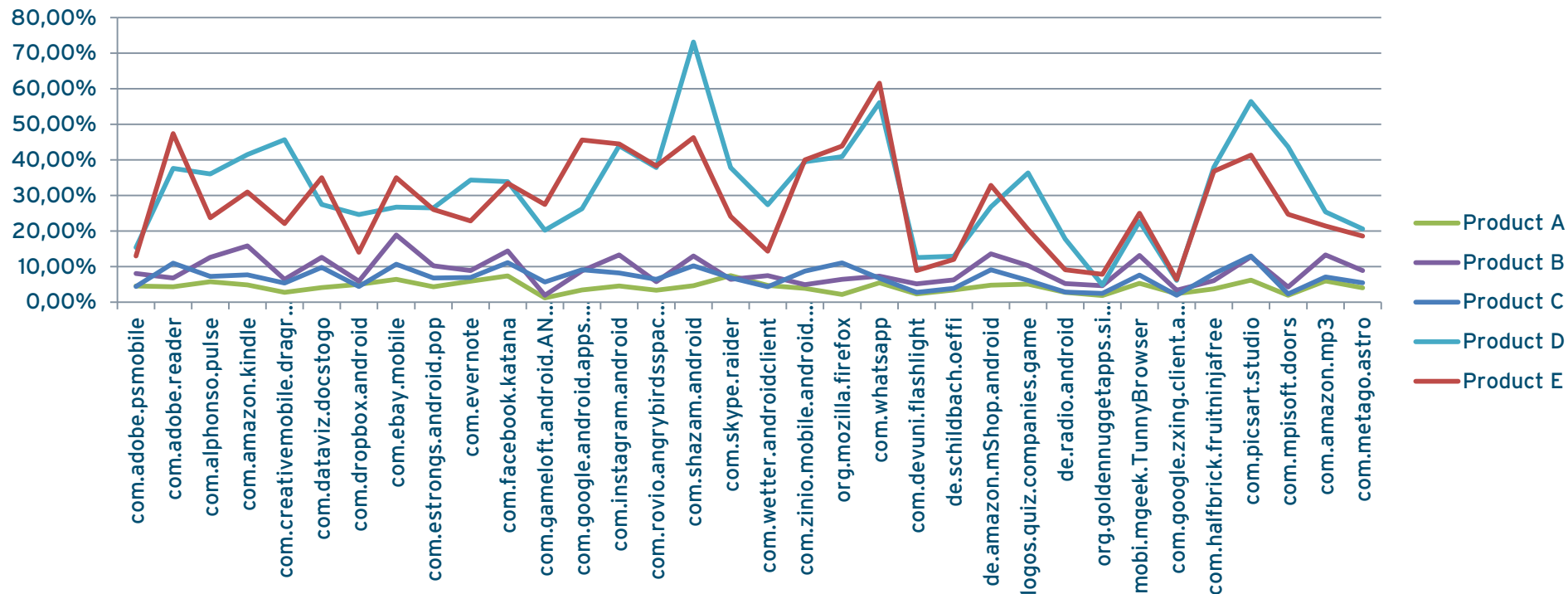
- **Be aware of greyware:**
  - Ad supported apps
  - Privacy risks

# Test Scenarios - Performance

- **Install clean apps from Google Play**
  - We can't use ADB here, because we can't disable USB charging
- **Monitor CPU-usage and battery discharge**
- **Repeat several times**

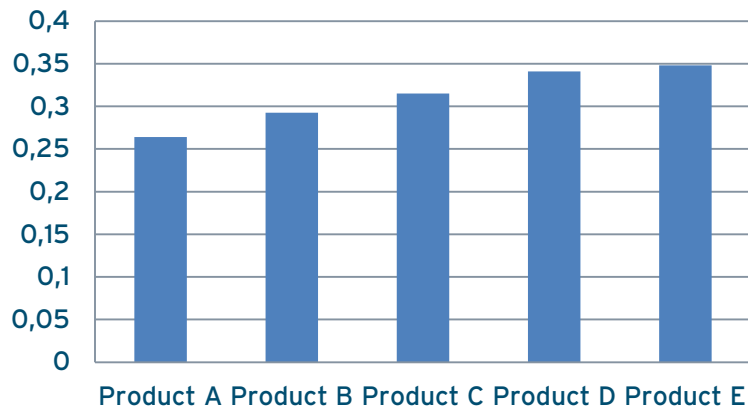
# Test Scenarios - Performance

## CPU usage

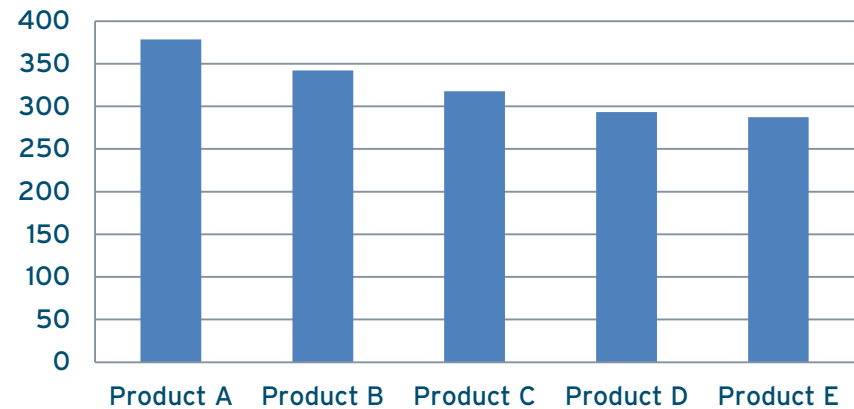


# Test Scenarios – Performance

**Discharge rate in  
% per minute**



**Estimated battery life in  
minutes**



# Test Scenarios – Performance

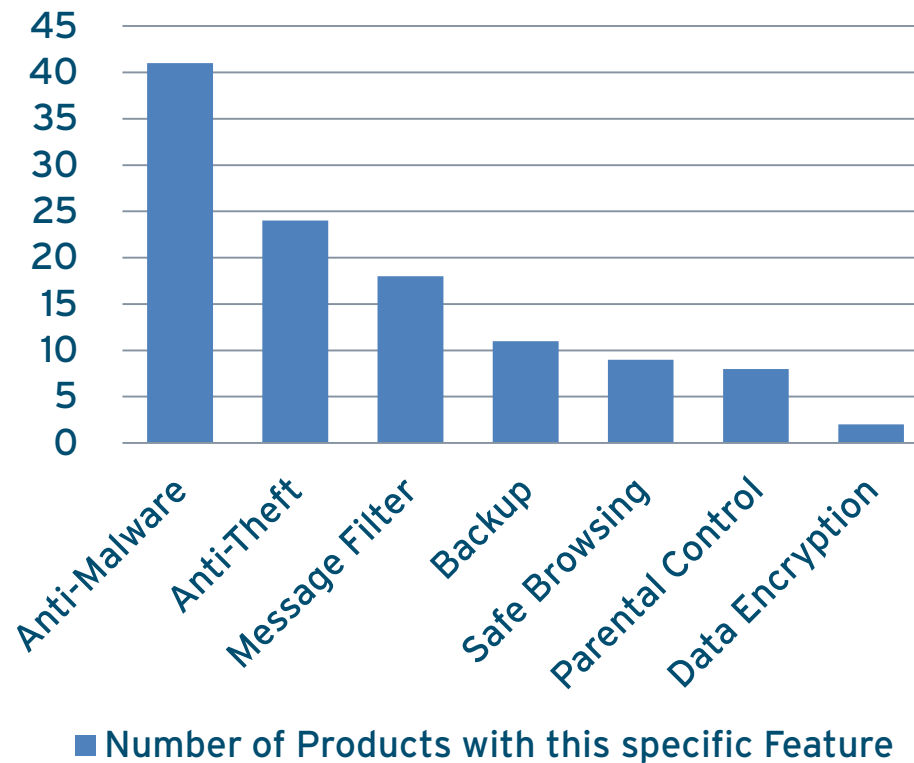
- **Measure impact on real-world usage**
  - **Loading websites**
  - **Sending/receiving messages**
  - **Opening apps**
  - **Playing media files**
  - ...

# Test Scenarios - Others?

- **Other functions are not common among all AV/mobile security products:**
  - **Anti-Theft**
  - **Backup, Encryption**
  - **Spam, Phishing**
  - ...



# Test Scenarios - Others?



# Automation

- **ADB-CLI**
- **ddmlib.jar (included in SDK)**
  - High Level API to control ADB
- **Robotium** <<http://code.google.com/p/robotium/>>
  - GUI automation of Android apps

# Problems

- **Not all apps support SD card scan**
- **No proper reporting**
- **No export of report files**

**Thank You!**

**Questions?**