

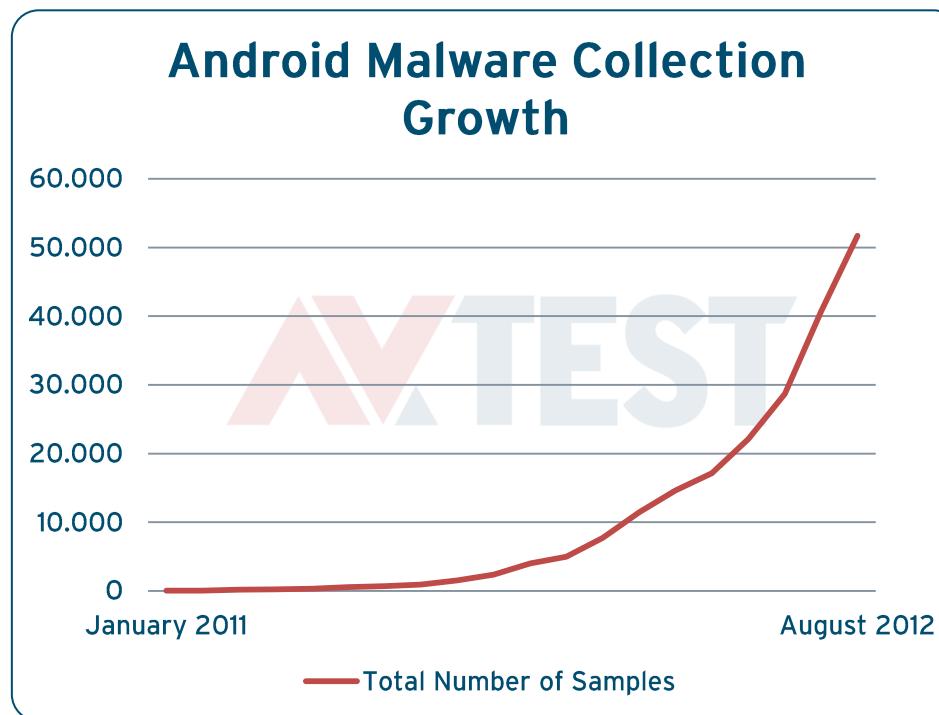
Building a Test Environment for Android Anti-Malware Tests

Hendrik Pilz
Director Technical Lab / Mobile Security
hpilz@av-test.de

Agenda

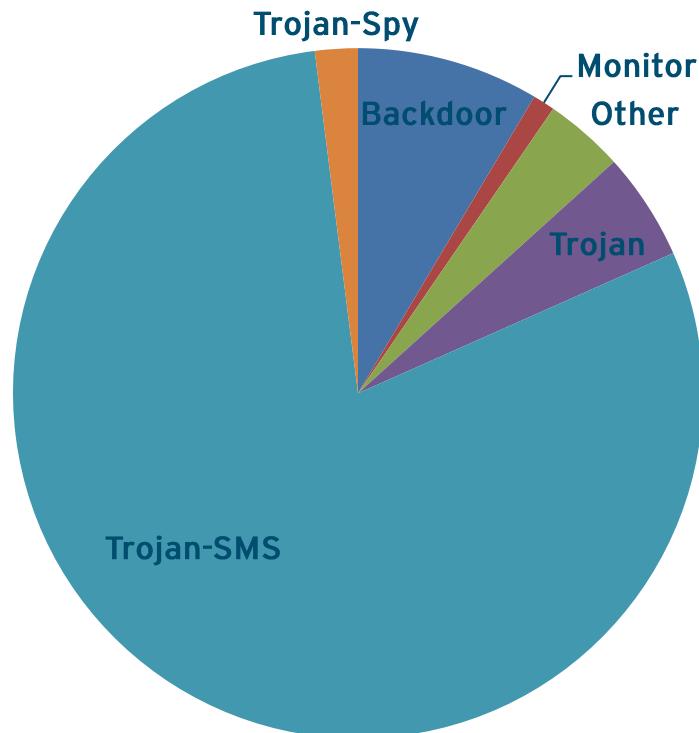
- **Android Malware Landscape**
- **Real Devices or Emulator?**
- **Preparation**
- **Test Scenarios**
- **Automation**
- **Problems**

Android Malware Landscape



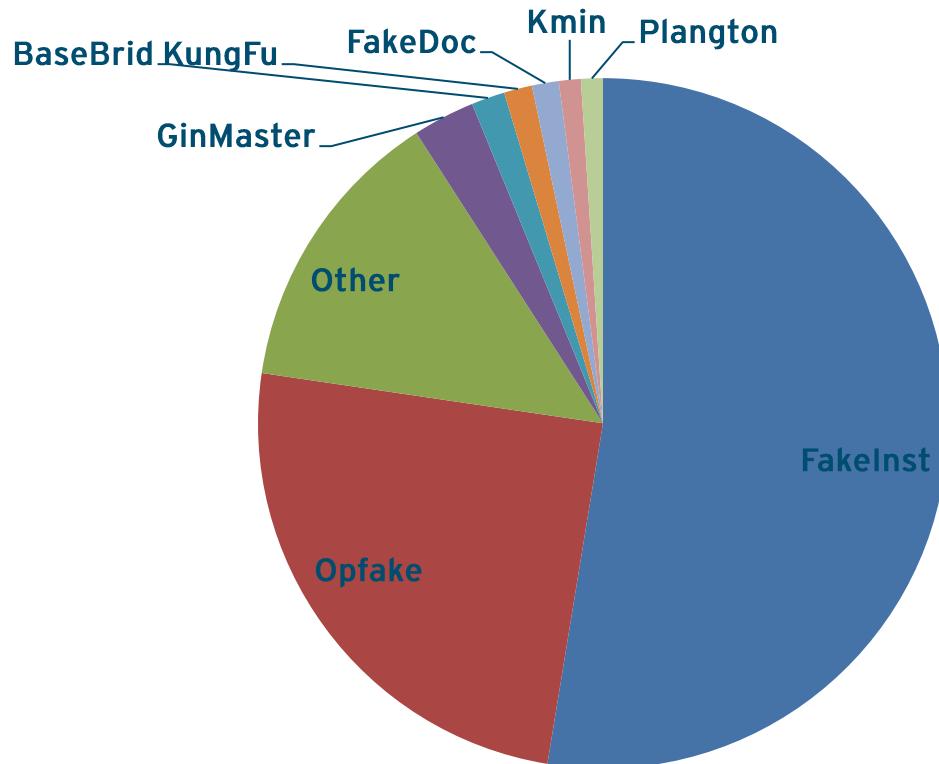
Android Malware Landscape

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**
- **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

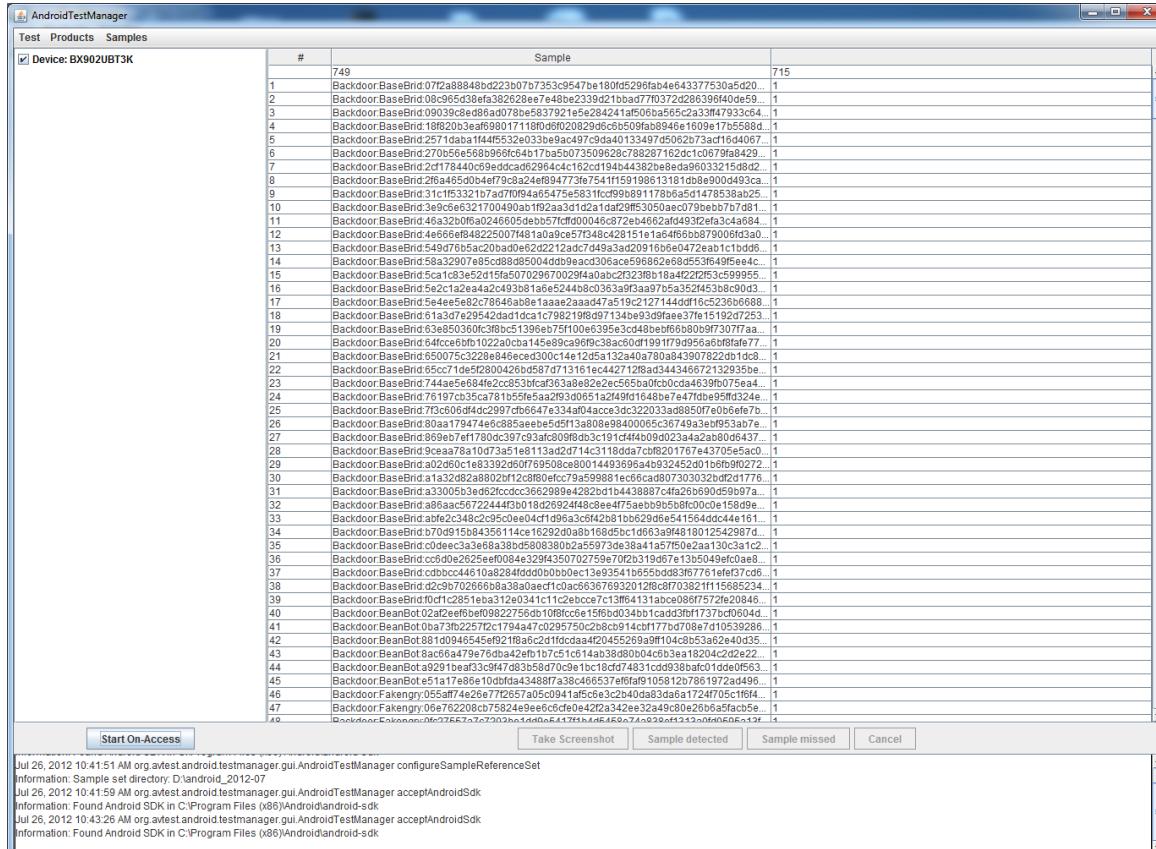
```
on-access.sh ✘
#!/bin/bash

c=1
# traverse sample directory
for i in `ls $1`
do
    sample="${1}/${i}"
    echo "${c}: Installing ${i}"
    # get android package name
    package=`aapt 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



The screenshot shows the AndroidTestManager application window. At the top, there's a menu bar with 'Test', 'Products', and 'Samples'. Below the menu is a toolbar with buttons for 'Start On-Access', 'Take Screenshot', 'Sample detected', 'Sample missed', and 'Cancel'. The main area is a table with columns for '#', 'Sample', and 'Result'. The table lists 40 samples, each with a unique ID and a brief description. Most samples show a result of '1' (detected). The last few samples show a result of '-1' (missed). At the bottom of the table, there's a message: 'Jul 26, 2012 10:41:51 AM org.avtest.android.testmanager.gui.AndroidTestManager configureSampleReferenceSet Information: Sample set directory: D:\Android_2012-07'. Below this, there are several 'Information:' messages indicating the location of the Android SDK and Java environment.

#	Sample	
1	Backdoor/BaseBrid 07f2a88848bd223b07b7353c9547be180fd296fab4e643377530a5d20...	1
2	Backdoor/BaseBrid 09e5638ef382628ee748be2339d21ba7d7f0372d286396f40de59...	1
3	Backdoor/BaseBrid 09039c8ed86ad078be5837921e5e28421a506ba565c3af7933c6...	1
4	Backdoor/BaseBrid 1882003ea9698011190f09020829d6c5059fa946e1609e175588d...	1
5	Backdoor/BaseBrid 2571dab14445532e033be9979d4a0133497505273a19d4067...	1
6	Backdoor/BaseBrid 2705e56b599fc64b17ba5b073509626788287162c10679fa429...	1
7	Backdoor/BaseBrid 2d17844059e6dcad62964c6162cd194a44382b159603321508d...	1
8	Backdoor/BaseBrid 28a45050097934246773f754d196213181969399393ca...	1
9	Backdoor/BaseBrid 29a45050097934246773f754d196213181969399393ca...	1
10	Backdoor/BaseBrid 3e09ca6321700400ab192a3d1d2a1d2985e0503ea0799eb77...	1
11	Backdoor/BaseBrid 46a320f6a0246f09debb57ff00046e0729b4662fa4d3fe03a4...	1
12	Backdoor/BaseBrid 4e885a94923507491e9e2957549420151e646fb307900f63e...	1
13	Backdoor/BaseBrid 549d765a20ba0d62d212ad74943a3ad2016b6e9472e01c101dd...	1
14	Backdoor/BaseBrid 58323907e85c2e8d95040d9e9ac3016ace59862e8a425536495e4c...	1
15	Backdoor/BaseBrid 5ca1c3e5d15fa5070296700294fa0abc2f238b18a4220f5c5995...	1
16	Backdoor/BaseBrid 5e2c12ae4a2c43391a6b02363a97b5352453bb90d3...	1
17	Backdoor/BaseBrid 5e44e56827085a6a9e1aae2aaa4475a19c2127144d1615c23b6688...	1
18	Backdoor/BaseBrid 61aa3762942d2da1c17821969d7134b9e37e37e15192d7253...	1
19	Backdoor/BaseBrid 63e8f50350f51396e75f100e639563c448beb6b680973077fa...	1
20	Backdoor/BaseBrid 64fccc6b0102a0cb1a14589ca969a60d199179956a8bfafaf77...	1
21	Backdoor/BaseBrid 6500753228e846eece303914e12d5a132a040780a43907822201d...	1
22	Backdoor/BaseBrid 65c71de5f2800426b587d713161e4247128ad344346672132935b...	1
23	Backdoor/BaseBrid 744e5684fe2cc053bca363a9e2e2ec565b9a0fc0cd4639b07954...	1
24	Backdoor/BaseBrid 76197cb35ca718b5f6e5a293d051a24961648be7e470be95f324...	1
25	Backdoor/BaseBrid 773c06f4d2397f664743a044ce3dc32203a3d8050f70b6e67b...	1
26	Backdoor/BaseBrid 80aa179474e8c8585d65f113a08e9940065e36749a3953a07e...	1
27	Backdoor/BaseBrid 8586976f1780dc397c93af80969db3c1914f4a09023a4a2b80d437...	1
28	Backdoor/BaseBrid 9cea178307073a51d13a2d7f14a118dd0c801767e43705e9a...	1
29	Backdoor/BaseBrid a0290c1e83392d69f769508ce80014493959a4932452d01b6f90272...	1
30	Backdoor/BaseBrid a132e82b8892f12c80fcfc79a59961e0c56e9ad8073032bdf2776...	1
31	Backdoor/BaseBrid a33095b3e952f0cc036529989e2829d1b4438874c65b699059597...	1
32	Backdoor/BaseBrid a881d1324863a9e2e2ec565b9a0fc0cd4639b07954...	1
33	Backdoor/BaseBrid b010915b94389114cc1629240a91904bc1693a9491801254297d...	1
34	Backdoor/BaseBrid c299704626698a39a0ed10a8636769320128e8f70382111568523...	1
35	Backdoor/BaseBrid c03e3a68a3b765008280725973d39a15675022a130-3a...	1
36	Backdoor/BaseBrid c5d4262eef0084e33943507027597d702b31967e13b5049e0d4e9...	1
37	Backdoor/BaseBrid cbcc484510824958d40b0b0c13923841b855d983577614f0d32d...	1
38	Backdoor/BaseBrid d299704626698a39a0ed10a8636769320128e8f70382111568523...	1
39	Backdoor/BaseBrid f011c2851e3a312e341c112ebcc7e13ff61131a...	1
40	Backdoor/BaseBot 0242eebf09822756b106fc03156bd034b41cad39f1737bd0604...	1
41	Backdoor/BaseBot 0ba73fb2572c1794a47c152502bcb914c177b...	1
42	Backdoor/BaseBot 881d094545e9f216a6c211fcd9aa42045526939f1046b53a2e40d35...	1
43	Backdoor/BaseBot 8ac6647976da42ef1b7c1s14ab38d804b3e18204...	1
44	Backdoor/BaseBot 92916bea3c9497d83b58709e1b1c0d74331cd938ba9c01d0e0563...	1
45	Backdoor/BaseBot e517a7e86e10dbd4a434887a38c4665376f6f9105...	1
46	Backdoor/FakeNotify:055af742e6677f657a0590941af5c6e3c2b40d93da8172470516f...	1
47	Backdoor/FakeNotify:06e762208cb758249e9e6c6fe042123a42e3249c80e28b65facb5...	1
48	Backdoor/FakeNotify:0e726637a7c73032a1440a4171b14d450274...	-1

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

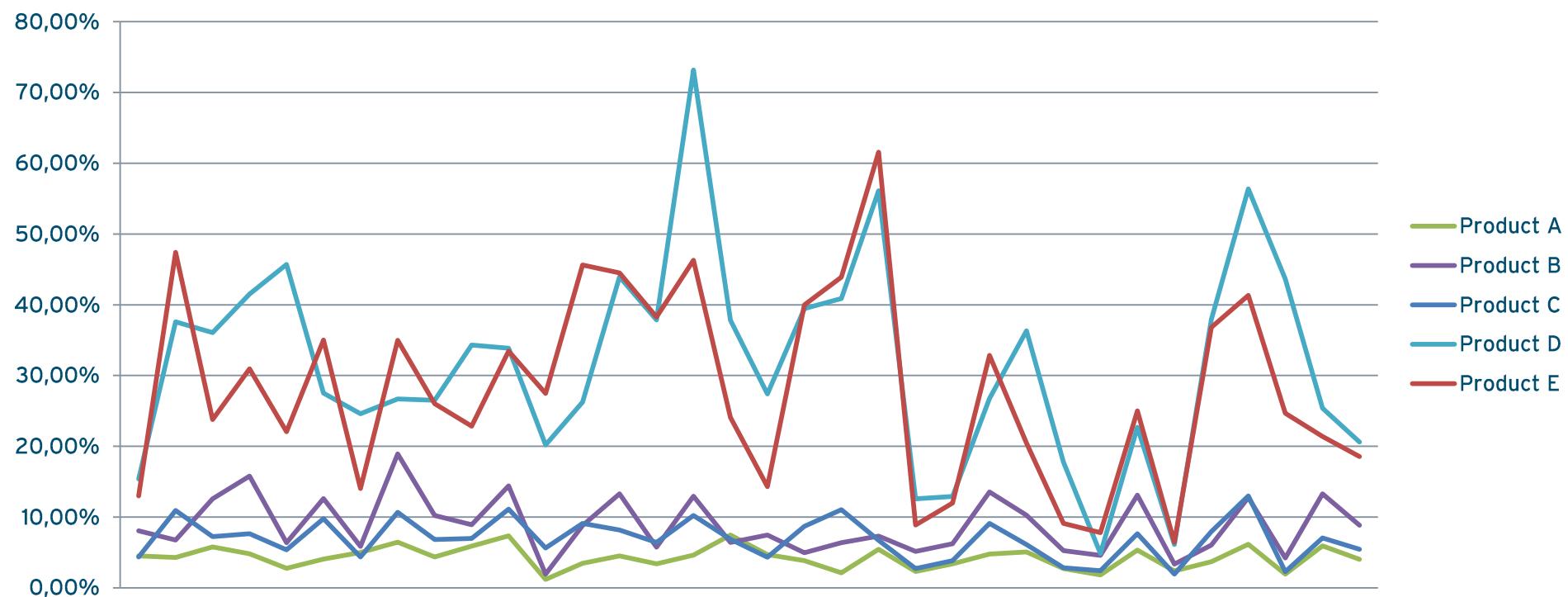
- Be aware of greyware:
 - Ad supported apps
 - Privacy risks

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**

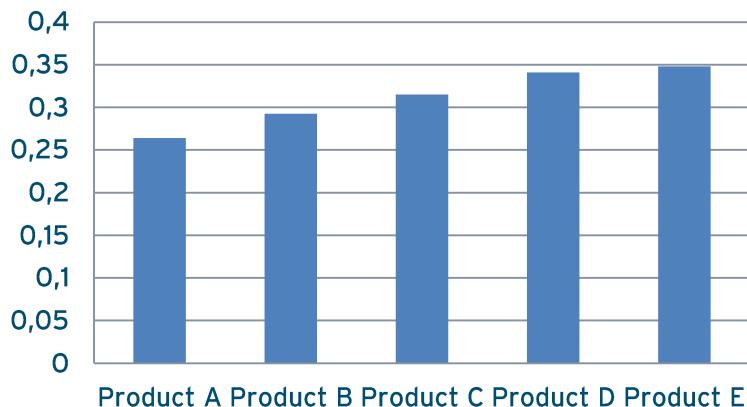
Test Scenarios - Performance

CPU usage

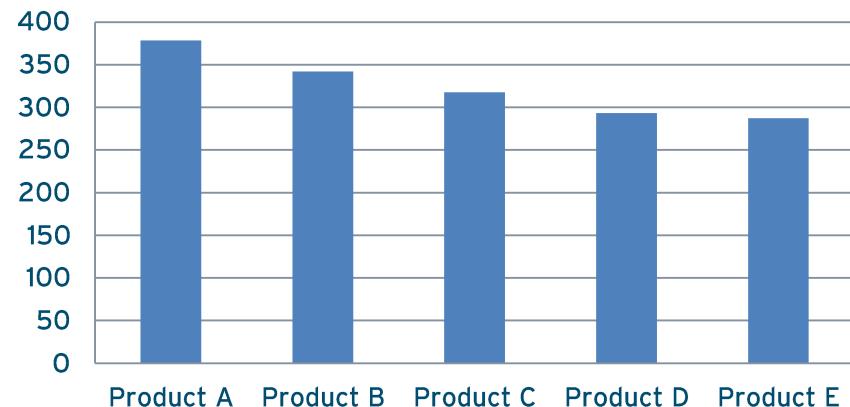


Test Scenarios - Performance

Discharge rate in
% per minute



Estimated battery life in
minutes



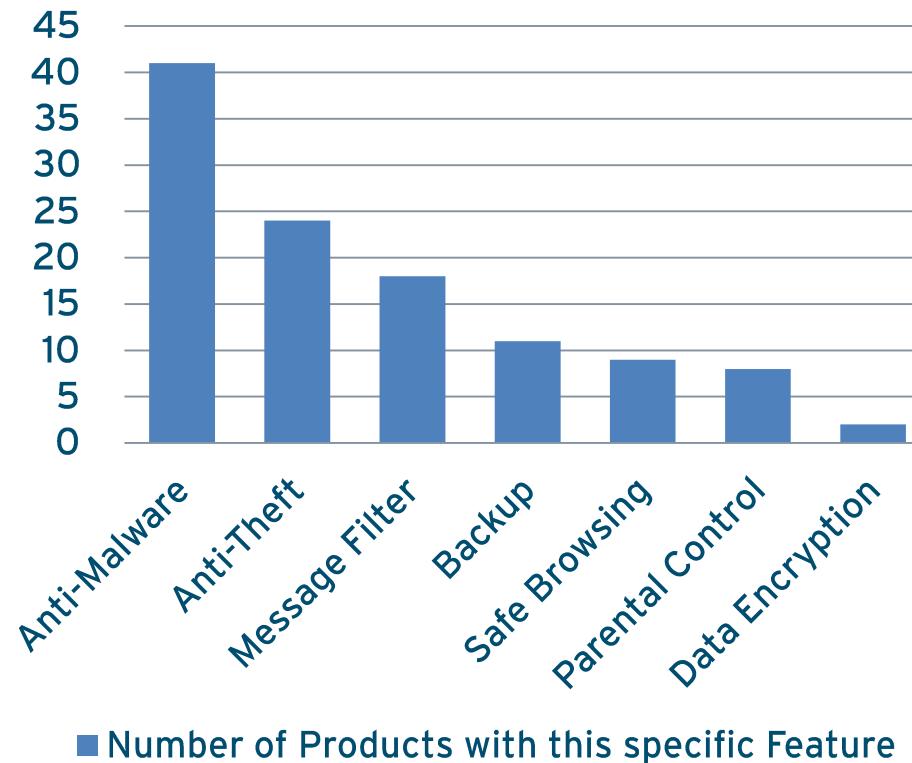
Test Scenarios – Performance

- **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?