SECURE2013

ANDROTOTAL

A SCALABLE FRAMEWORK FOR ANDROID ANTIMALWARE TESTING

Federico Maggi, Andrea Valdi, Stefano Zanero

Politecnico di Milano, DEIB

fede@maggi.cc

ROADMAP

1. Android threats and protections
2. Limitations
3. Testing antimalware
4. AndroTotal
5. Status

1. ANDROID THREATS AND PROTECTIONS

2. LIMITATIONS

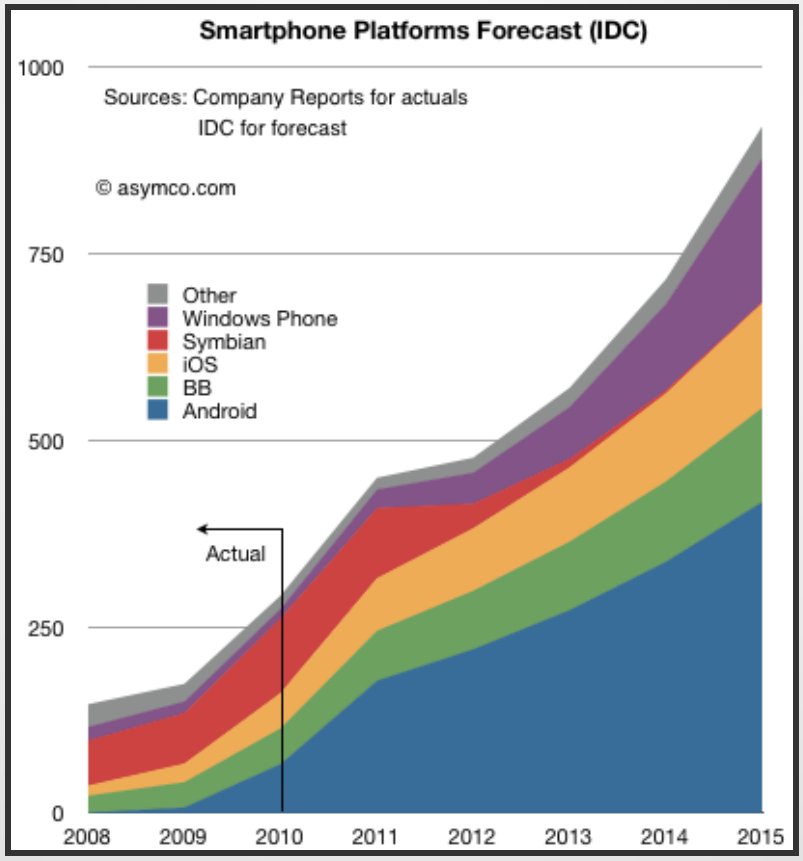
3. TESTING ANTIMALWARE

4. ANDROTOTAL

5. STATUS

ANDROID FACTS

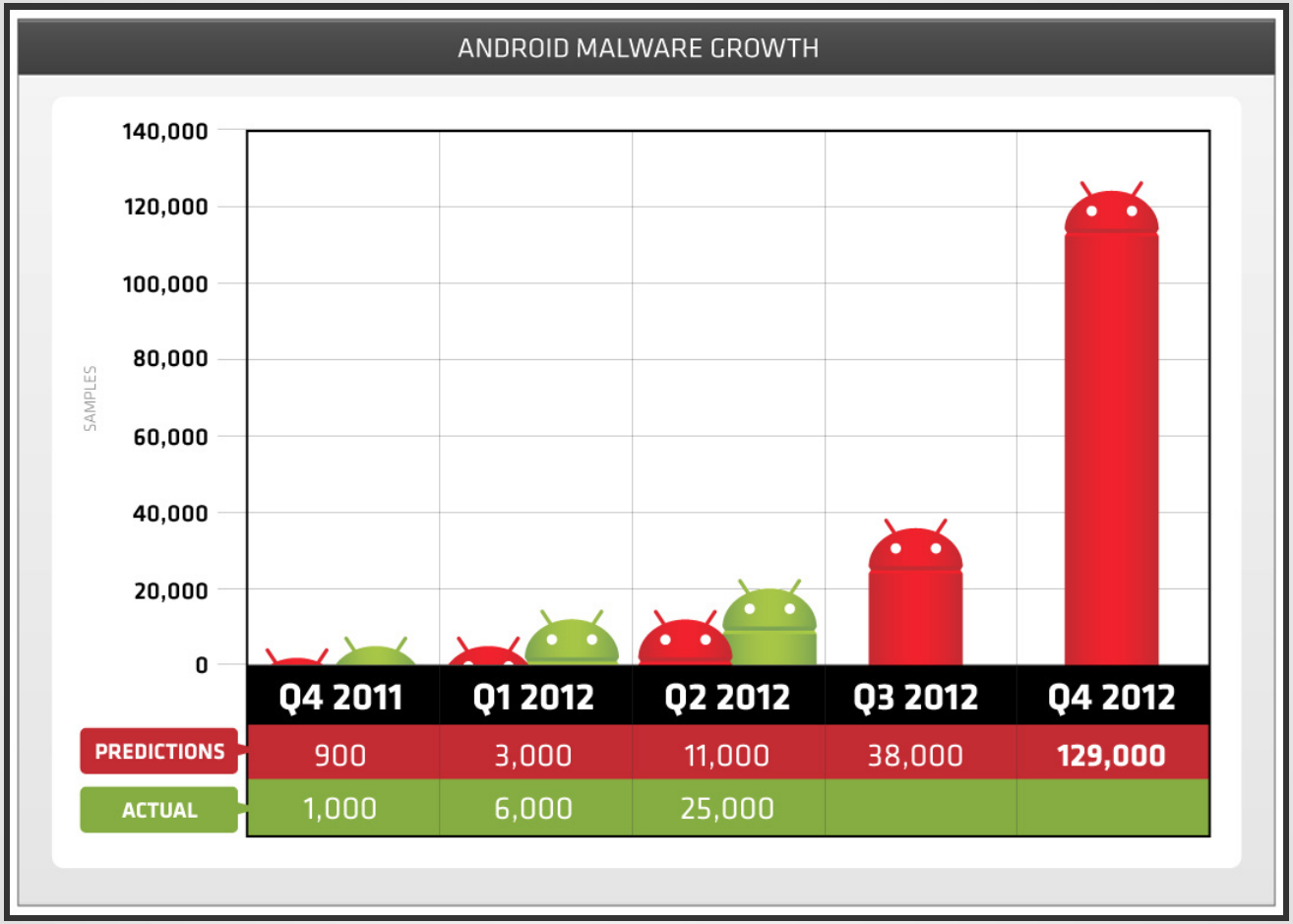
- Android is the most popular mobile platform (79%)
- Rich marketplaces stocked with apps
- Very attractive target for attackers



ATTACKERS GOALS

- Steal sensitive data (intercept texts or calls)
- Turn devices into bots (perform malicious actions)
- Financial gain (call or text premium numbers)

GROWTH OF MALICIOUS APPS (2011–2012)



http://blog.trendmicro.com/trendlabs-security-intelligence/byod-a-leap-of-faith-for-enterprise-users/

NUMBER OF MOBILE 'THREATS' (Q1 2013)

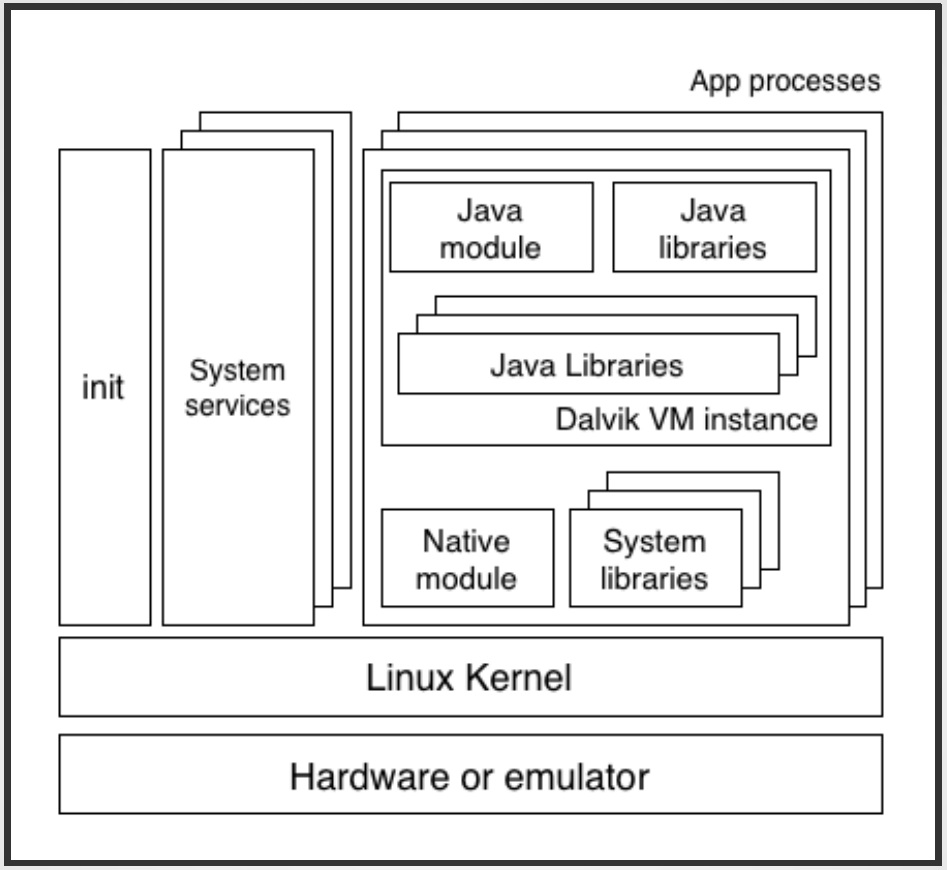
- Symantec: ~3,900
- McAfee: ~60,000
- TrendMicro: ~509,000

Google @ VB2013: Situation is vastly exaggerated

GOOGLE'S LAYERED SECURITY APPROACH

- Google Play vetting
- Install and permission confirmation
- SMS/call blacklisting and quota
- Runtime checks (?)
- App sandboxing

APP SANDBOXING



"Sensitive" operations require static permissions

1. THREATS AND PROTECTIONS

2. LIMITATIONS

3. TESTING ANTIMALWARE

4. ANDROTOTAL

5. STATUS

ANTIMALWARE LIMITATIONS

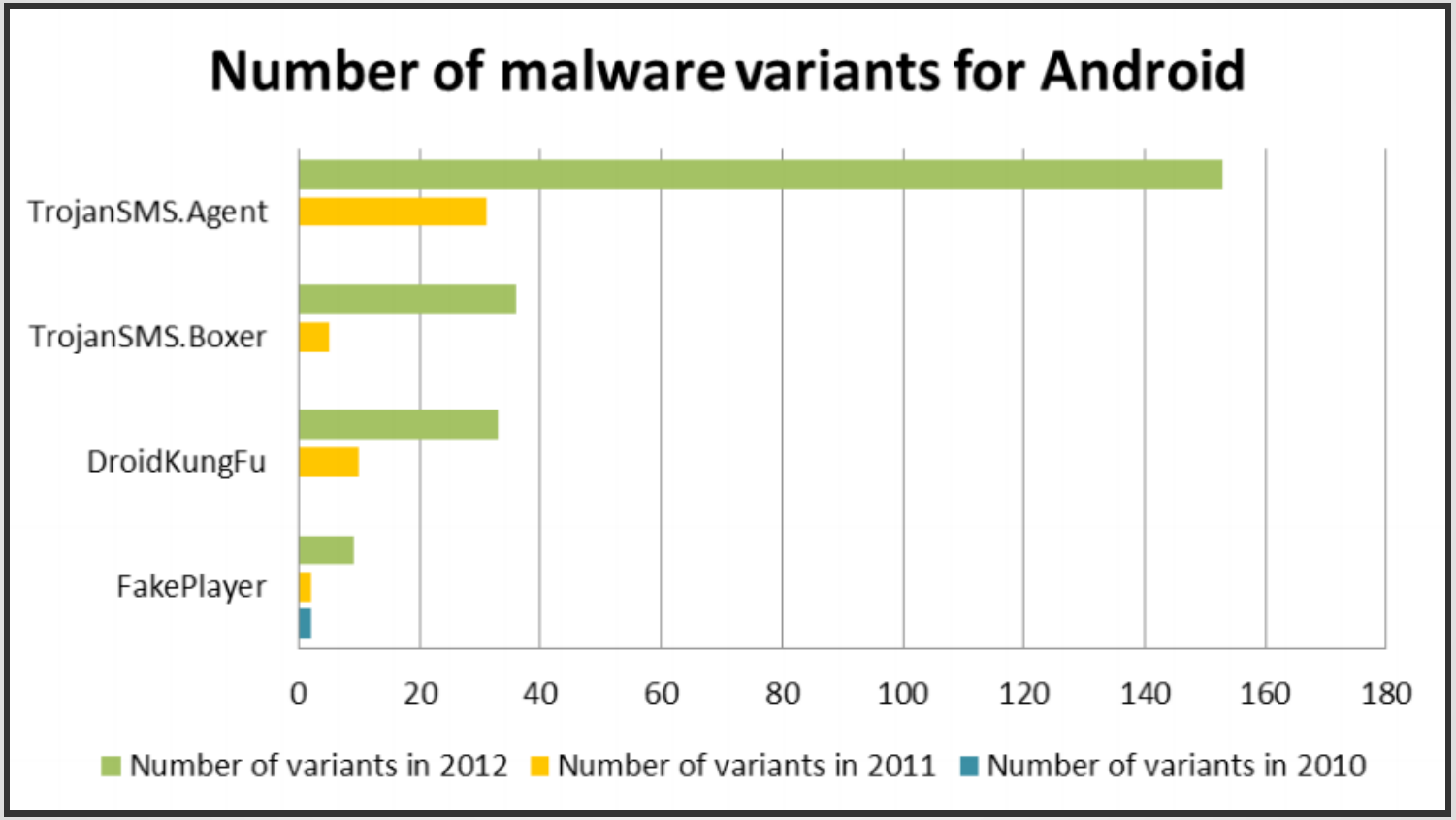
- No primitives for auditing running processes
- Workarounds:
  - Signature-based matching
  - Custom kernel (e.g., intercept syscalls)
  - Root the device and increase the antimalware's privileges

MALWARE LIMITATIONS

- Less freedom: a malware is an isolated app itself
- Workarounds:
  - Social engineering
  - **Signature evasion**

SIGNATURE EVASION

MORE VARIANTS THAN DISTINCT FAMILIES



http://go.eset.com/us/resources/white-papers/Trends_for_2013_preview

SIGNATURE EVASION

OBFUSCATION, ENCRYPTION, REPACKAGING

ADAM: An Automatic and Extensible Platform to Stress Test Android Anti-Virus Systems, DIVMA2013

DroidChameleon: Evaluating Android Anti-malware against Transformation Attacks, AsiaCCS2013

Based on this research we implemented 11 mutation scripts.

1. THREATS AND PROTECTIONS

2. LIMITATIONS

3. TESTING ANTIMALWARE

4. ANDROTOTAL

5. STATUS

ANTIMALWARE PRODUCTS

- About 100 (free) antimalware apps
- Extra features on rooted devices

HOW TO TEST THEM?

1. Obtain **M** samples of known malware
2. Apply **T** transformations to each sample
3. Analyze **M** × **T** variants with **P** antimalware apps
4. Repeat for each of the **A** Android versions

NUMBERS

- M = 1,000 (very conservative)
- T = 11
- P = 100
- A = 3 (2.3, 4.1, 4.2)

1,000 × 11 × 100 × 3 = 3,300,000 TESTS

LACK OF AUTOMATION TOOLS

VIRUSTOTAL.COM?

- Command-line, desktop-based AVs with signatures for Android
- Unclear whether the same signatures will work on the respective mobile products
- No versioning support

STATE OF THE ART

- H. Pilz, *"Building a test environment for Android anti-malware tests,"* Virus Bulletin Conference '12
  - Human oracle is needed
- M. Zheng, P. P. C. Lee, and J. C. S. Lui, *"ADAM: An Automatic and Extensible Platform to Stress Test Android Anti-Virus Systems,"* DIMVA'12
  - Focus on transformation
- V. Rastogi, Y. Chen, and X. Jiang, *"DroidChameleon: Evaluating Android Anti-malware against Transformation Attacks,"* AsiaCCS'13
  - Focus on transformation

TECHNICAL REQUIREMENTS

- Scalable architecture
- Android antimalware products are UI driven

1. THREATS AND PROTECTIONS

2. LIMITATIONS

3. TESTING ANTIMALWARE

4. ANDROTOTAL

5. STATUS

- SDK for writing UI tests/scrapers
- Pluggable adapters for each antimalware
- Parametric tests (e.g., version, platform)
- Task queues with distributed workers

CHARACTERISTICS

- Web frontend for humans
- JSON/REST API for machines
- Pluggable code-transformation modules
- Works on both emulators and physical devices

Scan application (advanced)

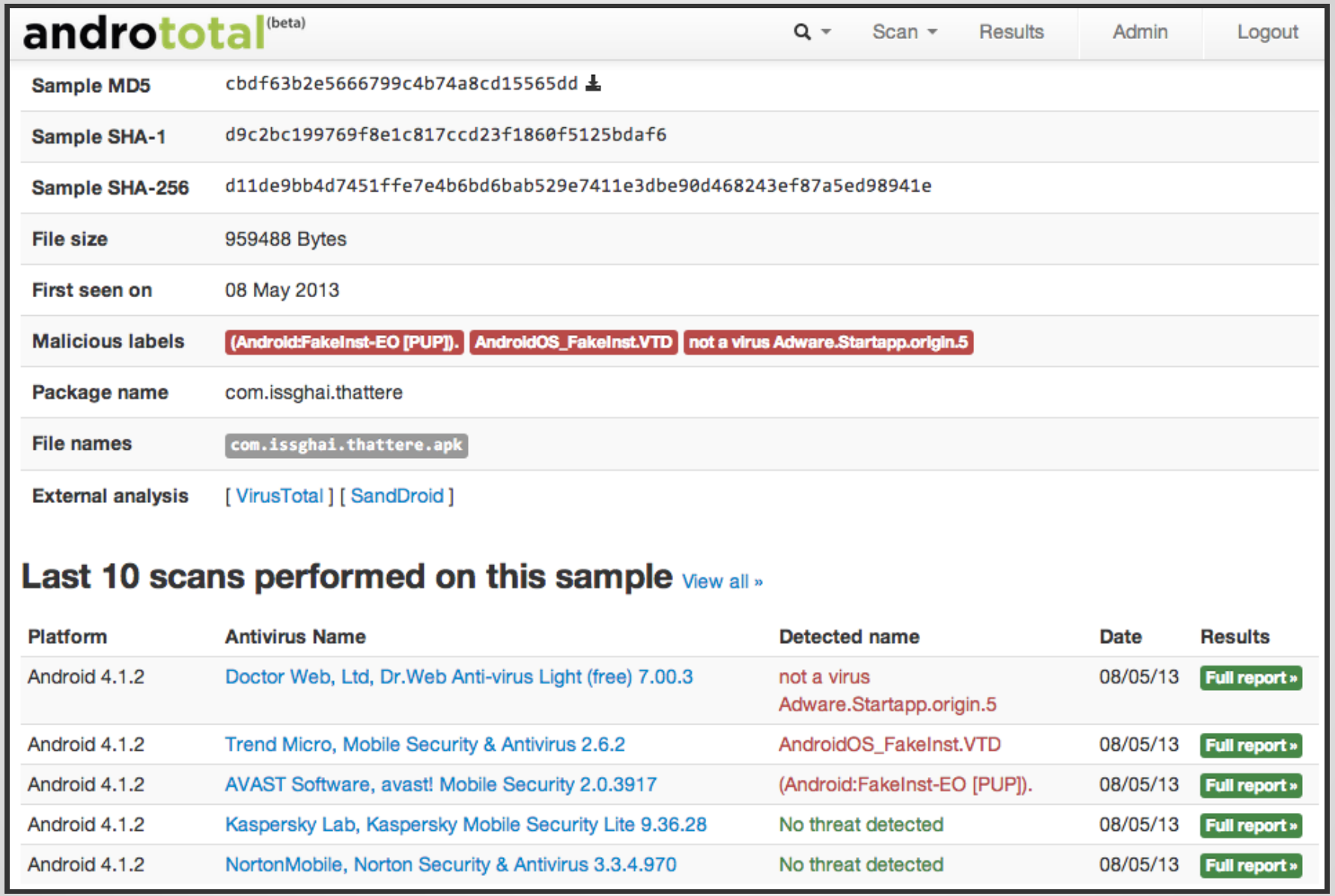
Sample File [Browse]

Is this sample a malware?
- ( ) Yes
- ( ) No
- ( ) I do not know

Force sample reanalysis [ ]

Are you human?

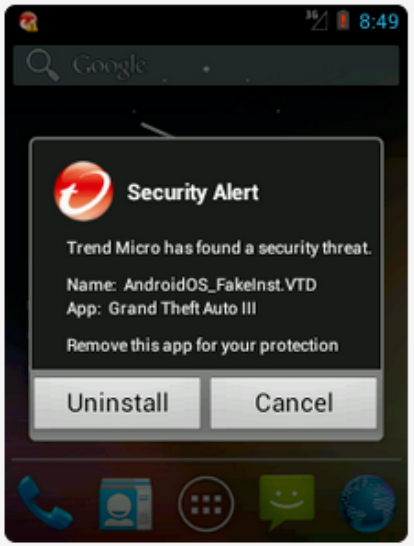
Antivirus name	Antivirus version	Android platform	Detection method	
Trend Micro, Mobile Security & Antivi	2.6.2	Android 4.1.2	On install	+
AVAST Software, avast! Mobile Security	2.0.3380	Android 4.1.2	On install	✖
AVAST Software, avast! Mobile Security	2.0.3380	Android 4.1.2	On demand	✖
AVAST Software, avast! Mobile Security	2.0.3917	Android 4.1.2	On install	✖


andrototal (beta)
Scan
Results
Admin
Logout
Sample MD5 cbdf63b2e5666799c4b74a8cd15565dd
Sample SHA-1 d9c2bc199769f8e1c817ccd23f1860f5125bdaf6
Sample SHA-256 d11de9bb4d7451ffe7e4b6bd6bab529e7411e3dbe90d468243ef87a5ed98941e
File size 959488 Bytes
First seen on 08 May 2013
Malicious labels (Android:FakeInst-EO [PUP]). AndroidOS_FakeInst.VTD not a virus Adware.Startapp.origin.5
Package name com.issghai.thattere
File names com.issghai.thattere.apk
External analysis [VirusTotal] [SandDroid]
Last 10 scans performed on this sample View all »
Platform Antivirus Name Detected name Date Results
Android 4.1.2 Doctor Web, Ltd, Dr.Web Anti-virus Light (free) 7.00.3 not a virus Adware.Startapp.origin.5 08/05/13 Full report »
Android 4.1.2 Trend Micro, Mobile Security & Antivirus 2.6.2 AndroidOS_FakeInst.VTD 08/05/13 Full report »
Android 4.1.2 AVAST Software, avast! Mobile Security 2.0.3917 (Android:FakeInst-EO [PUP]). 08/05/13 Full report »
Android 4.1.2 Kaspersky Lab, Kaspersky Mobile Security Lite 9.36.28 No threat detected 08/05/13 Full report »
Android 4.1.2 NortonMobile, Norton Security & Antivirus 3.3.4.970 No threat detected 08/05/13 Full report »

Trend Micro / Mobile Security & Antivirus / 2.6.2 / 2013-05-08 17:07:23 - cbdf63b2e5666799c4b74a8cd15565dd

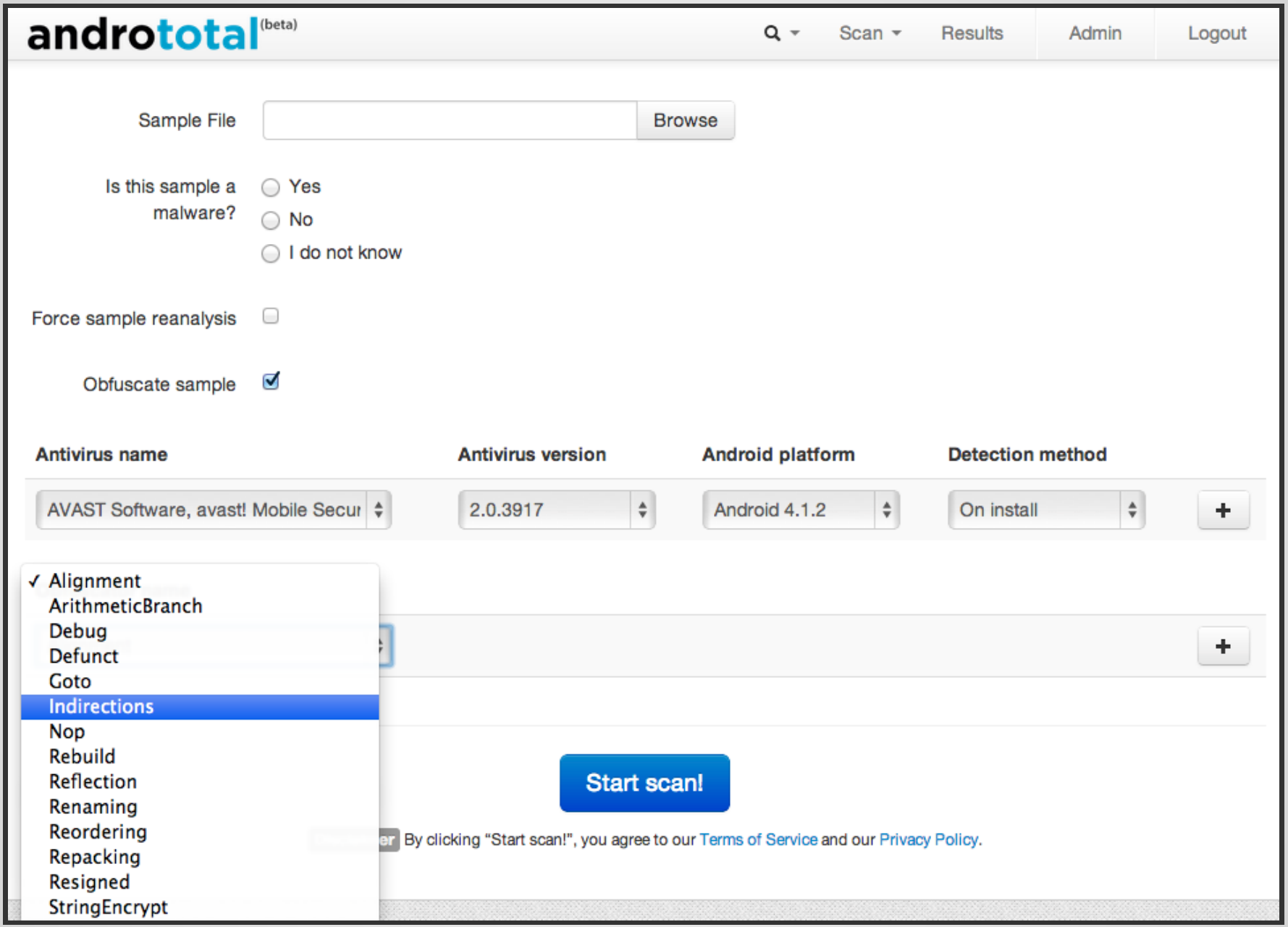
Mobile Security & Antivirus 2.6.2 scan for cbdf63b2e5666799c4b74a8cd15565dd

Task id	131bd4fe-3bcd-4a72-a207-683ed8eb79f1
Vendor name	Trend Micro
Antivirus name	Mobile Security & Antivirus
Engine version	2.6.2
Analysis started on	08/05/2013 at 17:05
Analysis completed on	08/05/2013 at 17:07 (took *91* seconds)
Detection method	On install
Analysis result	AndroidOS_FakeInst.VTD
Sample md5	`cbdf63b2e5666799c4b74a8cd15565dd`



Logcat dump (download)

```
99.  I/tmms-vsapi-jni(  674): VSReadVirusPattern OK. Action successful.
100. I/tmms-vsapi-jni(  674): OK. VSSetProcessAllFileInArcFlag. oldValue = ret = 0.
101. I/tmms-vsapi-jni(  674): OK. VSSetExpandLiteFlag. oldValue = ret = 1.
102. I/tmms-vsapi-jni(  674): OK. VSSetProcessAllFileFlag. oldValue = ret = 0.
103. I/tmms-vsapi-jni(  674): OK. VSSetCleanZipFlag. oldValue = ret = 0.
104. I/tmms-vsapi-jni(  674): OK. VSSetCleanBackupFlag. oldValue = ret = 0.
105. I/tmms-vsapi-jni(  674): VSGetDetectableVirusNumber virus in patter num = 3283
106. I/tmms-vsapi-jni(  674): filename = /data/data/com.trendmicro.tmmspersonal/Library/pattern/msvpnaos.457
107. I/tmms-vsapi-jni(  674): InternalVer = 145700, PtnVer = 457.
108. D/PrepareVSAPI4RTScan(  674): before tmmsAntiMalwareJni4RTScan.init()!
109. I/tmms-vsapi-jni(  674): VSInit OK!
110. D/PrepareVSAPI4RTScan(  674): after tmmsAntiMalwareJni4RTScan.init()!
111. I/tmms-vsapi-jni(  674): in vsSetPatternPath, vc = 711579352
112. I/tmms-vsapi-jni(  674): Current pattern path is : /etc/iscan
113. I/tmms-vsapi-jni(  674): Pattern path is set to : /data/data/com.trendmicro.tmmspersonal/Library/pattern
114. I/tmms-vsapi-jni(  674): Pattern file(s) successfully deleted.
115. I/tmms-vsapi-jni(  674): in vsLoadPattern, vc = 711579352, sharedVC = 708085592, scanType =
116. I/tmms-vsapi-jni(  674): vsLoadPattern patternPath = /data/data/com.trendmicro.tmmspersonal/Library/pattern.
```


andrototal (beta)
Scan
Results
Admin
Logout
Sample File
Browse
Is this sample a malware?
Yes
No
I do not know
Force sample reanalysis
Obfuscate sample
Antivirus name
Antivirus version
Android platform
Detection method
AVAST Software, avast! Mobile Secur
2.0.3917
Android 4.1.2
On install
Alignment
ArithmeticBranch
Debug
Defunct
Goto
Indirections
Nop
Rebuild
Reflection
Renaming
Reordering
Repacking
Resigned
StringEncrypt
Start scan!
By clicking "Start scan!", you agree to our Terms of Service and our Privacy Policy.

WRITING TESTS ~~IS~~ WAS TEDIOUS

We have abstracted away the low level details, so that we can focus on the important things: *extracting the results*.

ANDROPILOT

TEST RECIPE (ON-INSTALL DETECTION)

```
#andrototal-adapters/ComZonerAndroidAntivirus.py
class TestSuite(base.BaseTestSuite):
  def on_install_detection(self, sample_path):
    self.pilot.install_package(sample_path)

    if self.pilot.wait_for_activity(
        "com.zoner.android.antivirus_common.ActScanResults", 10):

      result = self.pilot.get_view_by_id("scaninfected_row_virus")
    else:
      result = False
```

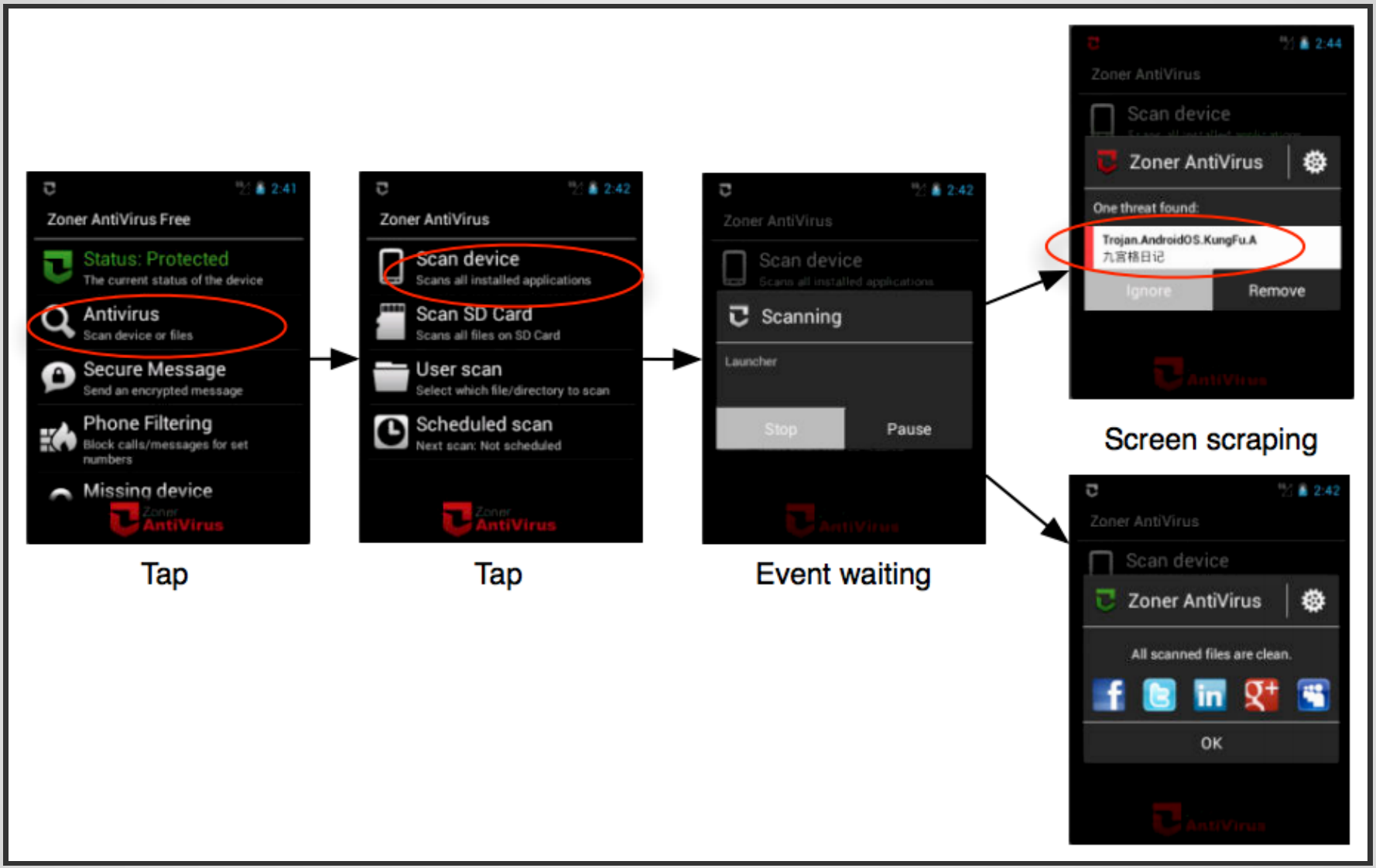
TEST RECIPE (ON-DEMAND DETECTION)

```
#...
def on_demand_detection(self, sample_path):
  self.pilot.install_package(sample_path)
  self.pilot.start_activity("com.zoner.android.antivirus", ".ActMain")
  self.pilot.wait_for_activity("com.zoner.android.antivirus.ActMain")

  self.pilot.tap_on_coordinates(120, 130)
  self.pilot.wait_for_activity("com.zoner.android.antivirus.ActMalware")

  # start scan
  self.pilot.tap_on_coordinates(120, 80)
  self.pilot.wait_for_activity(
    "com.zoner.android.antivirus_common.ActScanResults")

  self.pilot.refre dsh()
# ...
```


Zoner AntiVirus Free
Status: Protected
The current status of the device
Antivirus
Scan device or files
Secure Message
Send an encrypted message
Phone Filtering
Block calls/messages for set numbers
Missing device
Tap
Zoner AntiVirus
Scan device
Scans all installed applications
Scan SD Card
Scans all files on SD Card
User scan
Select which file/directory to scan
Scheduled scan
Next scan: Not scheduled
Tap
Scanning
Launcher
Stop
Pause
Event waiting
One threat found:
Trojan.AndroidOS.KungFu.A
九宫格日记
Ignore
Remove
Screen scraping
All scanned files are clean.
OK

WORKFLOW

1. Retrieve a suspicious APK
2. Choose parameters
   - Android version(s)
   - List of antimalware product and versions
   - Apply chain of mutations
3. Pull clean image(s) from repository
4. Instantiate one test per combination of
   - Android version
   - Product version
5. Enqueue test instances

ARCHITECTURE

- Web frontend
- Repository of clean Android images
- Asynchronous task dispatcher
- Distributed workers

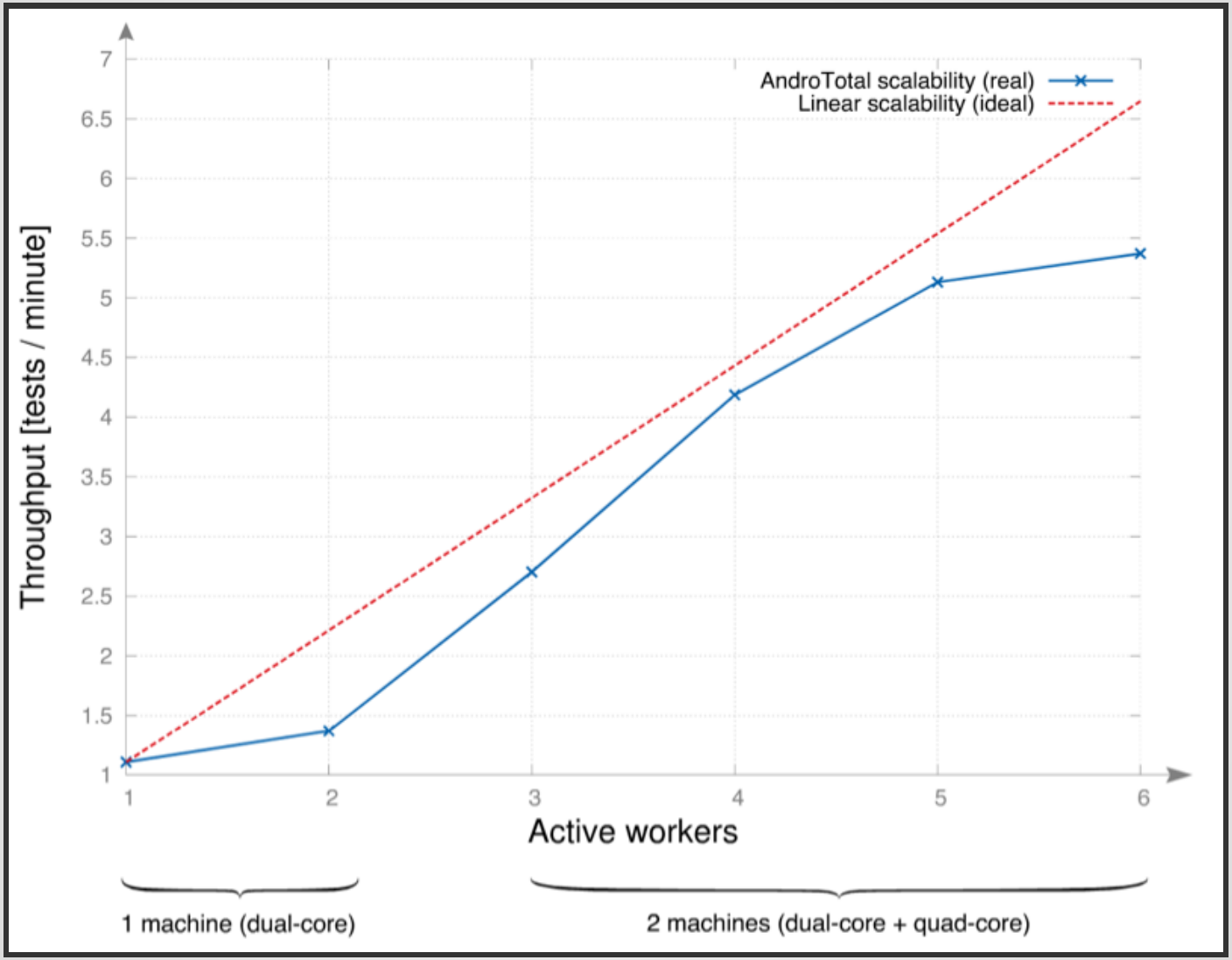
REST/JSON API AND CLIENT

- Push (public) and pull (invite only) samples
- Python client: https://bitbucket.org/andrototal/tools

```
$ python andrototal_cli.py -l DEBUG scan -at-key <...> -ms-key <...> path/to

Running command: scan
Uploading file sample.apk
Scan response: {"resource": "10a6f3efc8bc40c1922facde7d055208"}
Uploading file sample2.apk
Scan response: {"resource": "e870c6748ca3409f84c9c9e1a91daf3f"}
Uploading file 40156a176bb4554853f767bb6647fd0ac1925eac.apk
Scan response: {"resource": "21d6c7234a184db6b8e52f2bab523787"}
Uploading file samples-3.apk
Scan response: {"resource": "ec5b3c94ed624d6993b52a50d63153fa"}
```

SCALABILITY



1. THREATS AND PROTECTIONS

2. LIMITATIONS

3. TESTING ANTIMALWARE

4. ANDROTOTAL

5. STATUS

NUMBERS

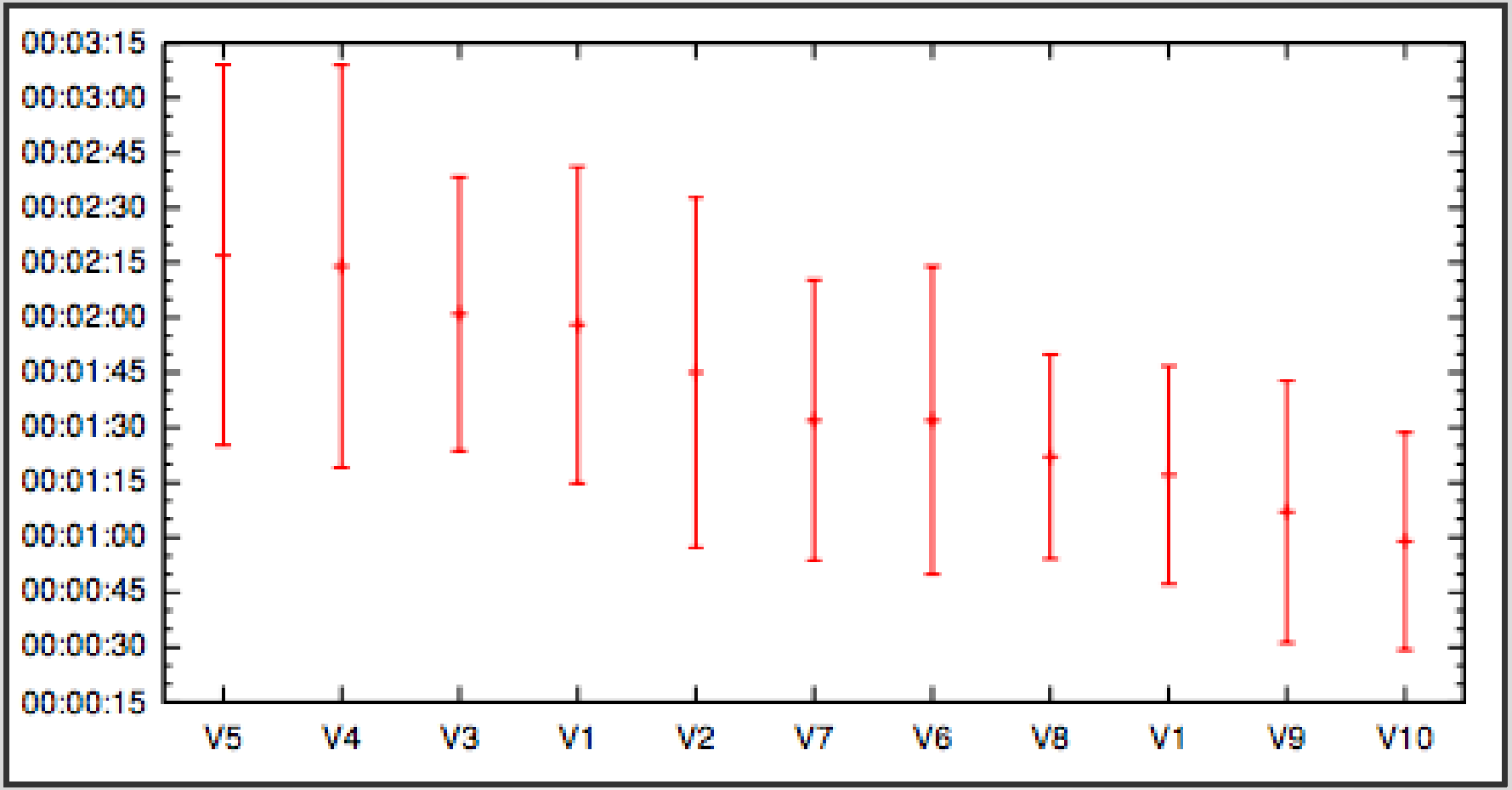
- 1,275 users subscribed
- 13 antimalware vendors supported (not all public)
- 16 products overall (not all public)
- 23,215 distinct APKs submitted and analyzed

SUPPORTED APPS (PUBLIC)

- ZONER, Inc. - Zoner AntiVirus Free 1.8.0
- ZONER, Inc. - Zoner AntiVirus Free 1.7.6
- AVAST Software - avast! Mobile Security 2.0.3917
- Doctor Web, Ltd - Dr.Web Anti-virus Light (free) 7.00.3
- Kaspersky Lab - Kaspersky Mobile Security Lite 9.36.28
- Kaspersky Lab - Kaspersky Mobile Security 10.4.41
- Trend Micro - Mobile Security & Antivirus 2.6.2
- Trend Micro - Mobile Security & Antivirus 3.1
- NortonMobile - Norton Security & Antivirus 3.2.0.769
- NortonMobile - Norton Security & Antivirus 3.3.4.970

Label	#
UDS:DangerousObject.Multi.Generic	3963
HEUR:Trojan-SMS.AndroidOS.Opfake.bo	1252
not a virus Adware.Airpush.origin.7	701
AndroidOS Opfake.CTD	700
HEUR:Trojan-SMS.AndroidOS.Opfake.a	628
Android.SmsSend.origin.281	620
Android:FakeNotify-A [Trj]	620
HEUR:Trojan-SMS.AndroidOS.FakeInst.a	512
Android.SmsSend.origin.315	485
HEUR:Backdoor.AndroidOS.KungFu.a	466
Android.SmsSend.origin.585	462
Android.SmsSend.origin.629	461
Adware.AndroidOS.Airpush-Gen	432
HEUR:Backdoor.AndroidOS.BaseBrid.a	390
AndroidOS Opfake.CTC	386

AVERAGE SPEED: NO MAJOR WINNER



FUTURE WORK

- Add more cores and scale
- Compare labels and detection results with VirusTotal.com
- Deploy on ARM boards and monitor power consumption
- Open malware repository and API: **anyone interested?**

GRAB A STICKER!

QUESTIONS?

http://andrototal.org

@andrototal_org

fede@maggi.cc