DAI-Labor
TU Berlin

Smartphone Malware Evolution Revisited

Aubrey-Derrick Schmidt

CC SEC
Security

14.09.2009 SIDAR Spring 2009 - Stuttgart

Agenda

1. Introduction
2. Background
3. Smartphone Malware Statistics
4. Countermeasures
5. Conclusion

1. Introduction

- Motivation
  - Smartphones get **increasingly popular**
  - Moore‘s law constantly leads to „stronger“ devices
    - Device got attractive to malware writers
  - Smartphones faced **wide range** of malware attacks
  - Most work **end 2006**
    - Continuous information needed for researchers
    - Public data inconsistent

Agenda

1. Introduction
2. Background
3. Smartphone Malware Statistics
4. Countermeasures
5. Conclusion

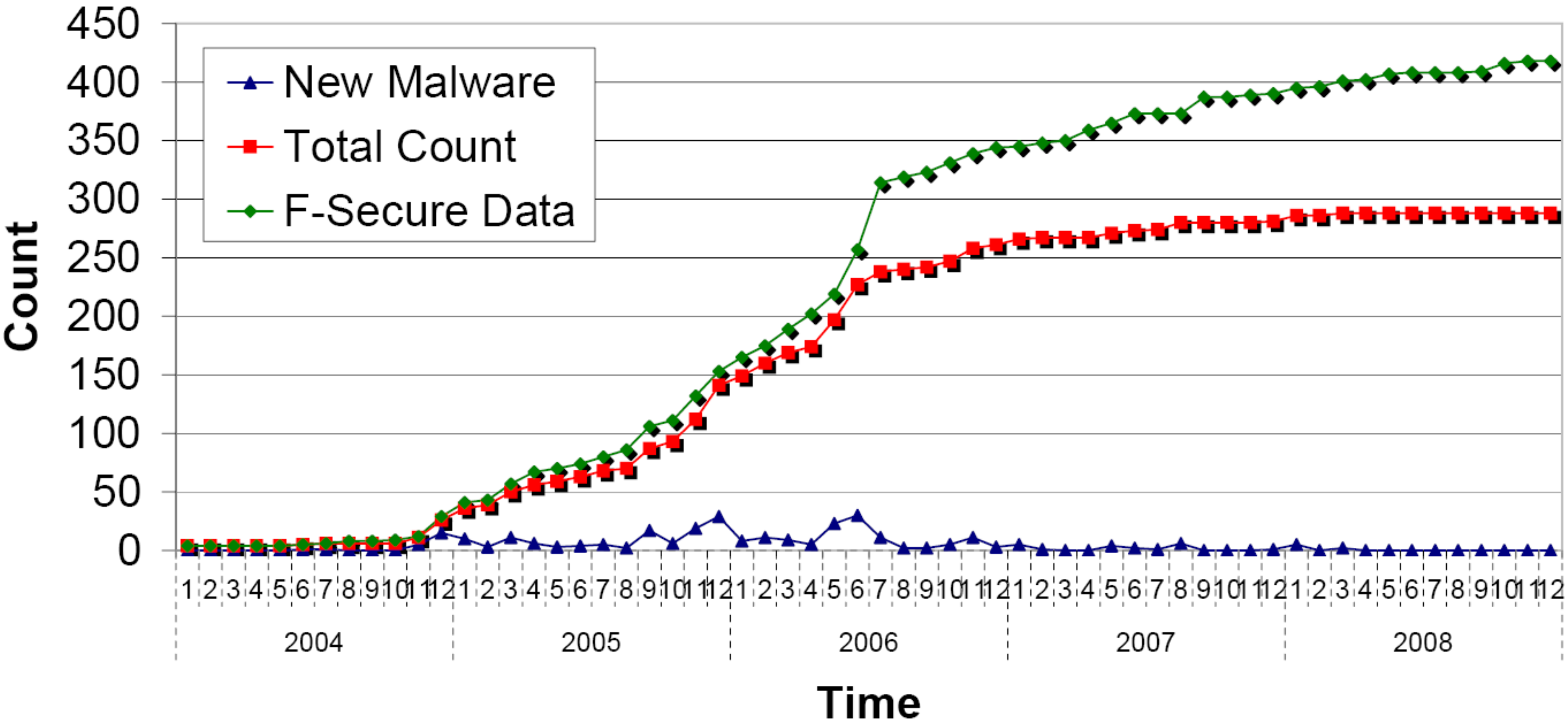
2. Background

- Malicious Software (Malware)
  - Software with malicious intentions
  - Major categories:
    - **Virus:** Hosting file can be virus itself, mostly needs **user interaction** for propagation
    - **Worm:** malicious code does not necessarily need a hosting file, normally, **no user interaction** needed for propagation
    - **Trojan horse: disguises malicious intention**, user interaction normally needed for propagation (user install)

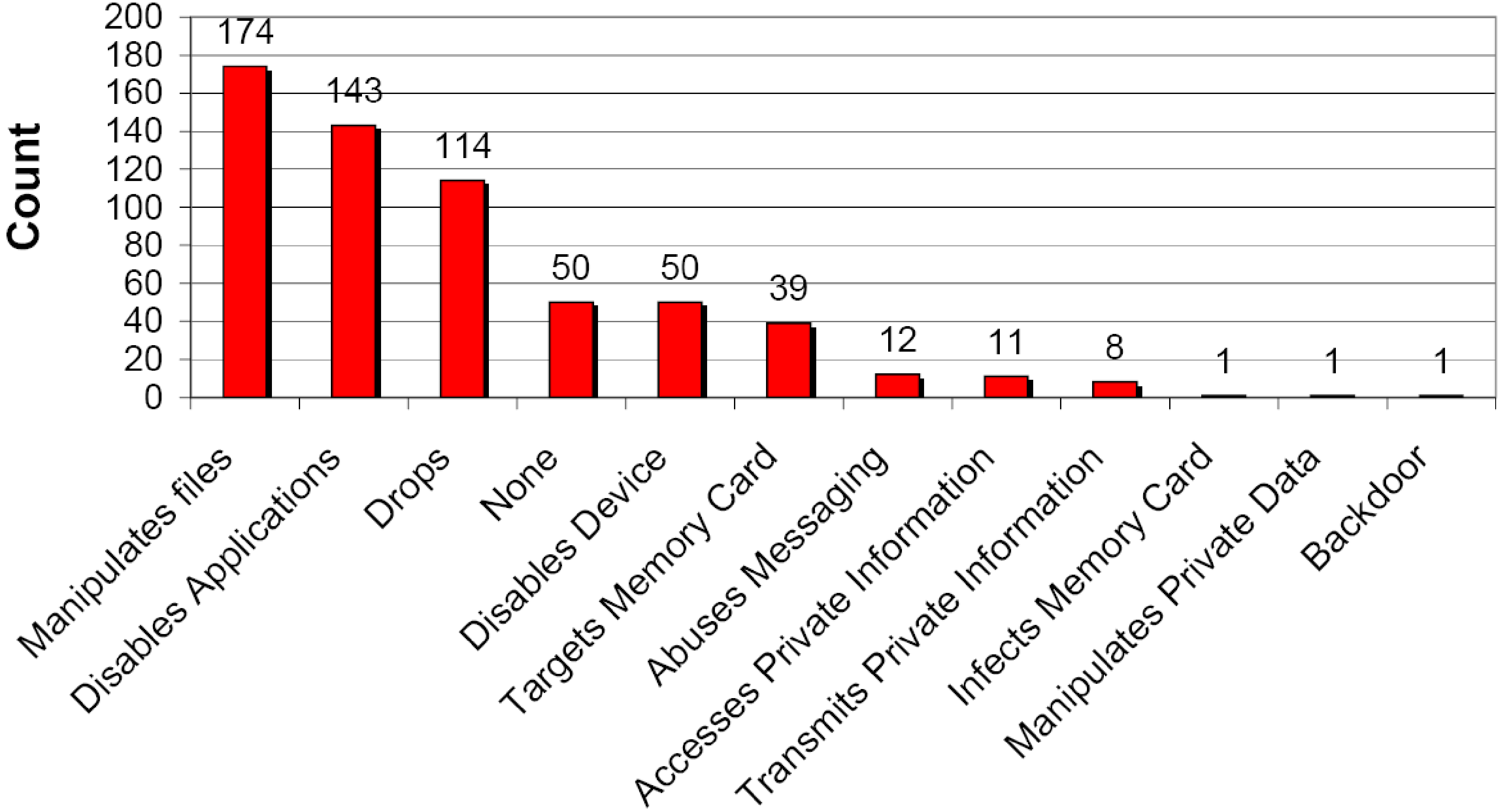
Agenda

1. Introduction
2. Background
3. Smartphone Malware Statistics
4. Countermeasures
5. Conclusion

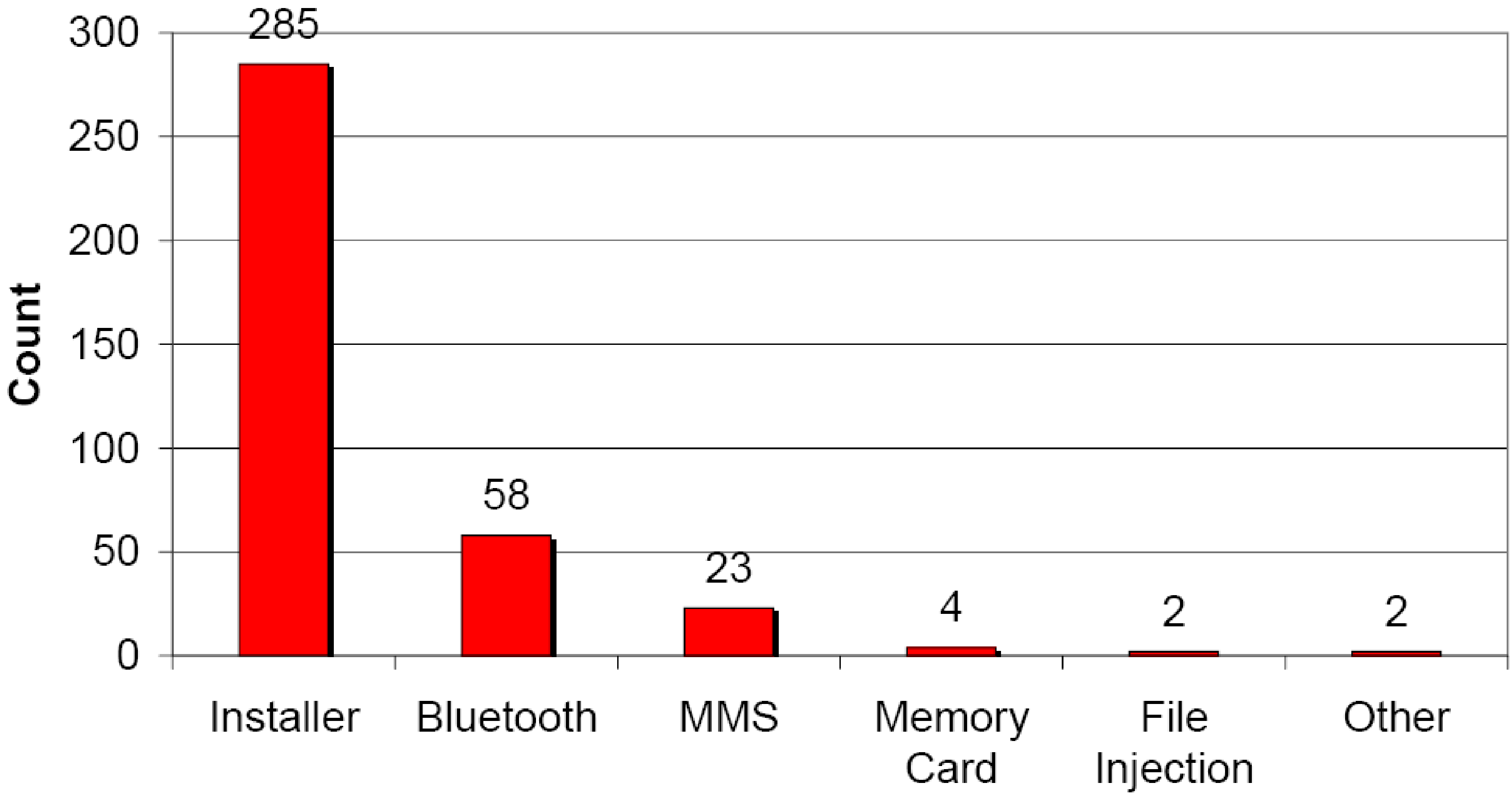
3. Smartphone Malware Statistics Smartphone Malware Appearance



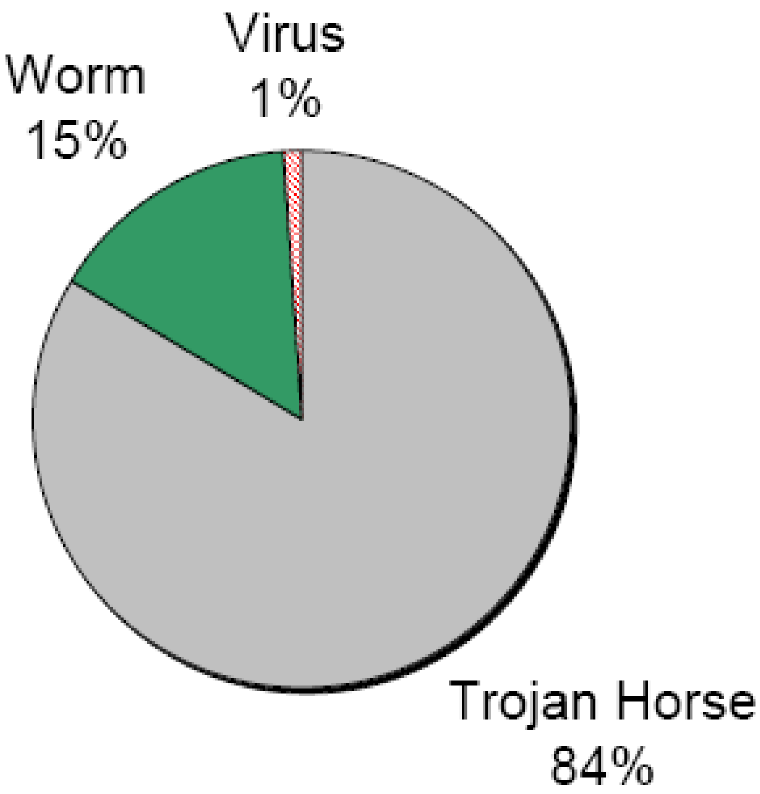
3. Smartphone Malware Statistics
Smartphone Malware Effects

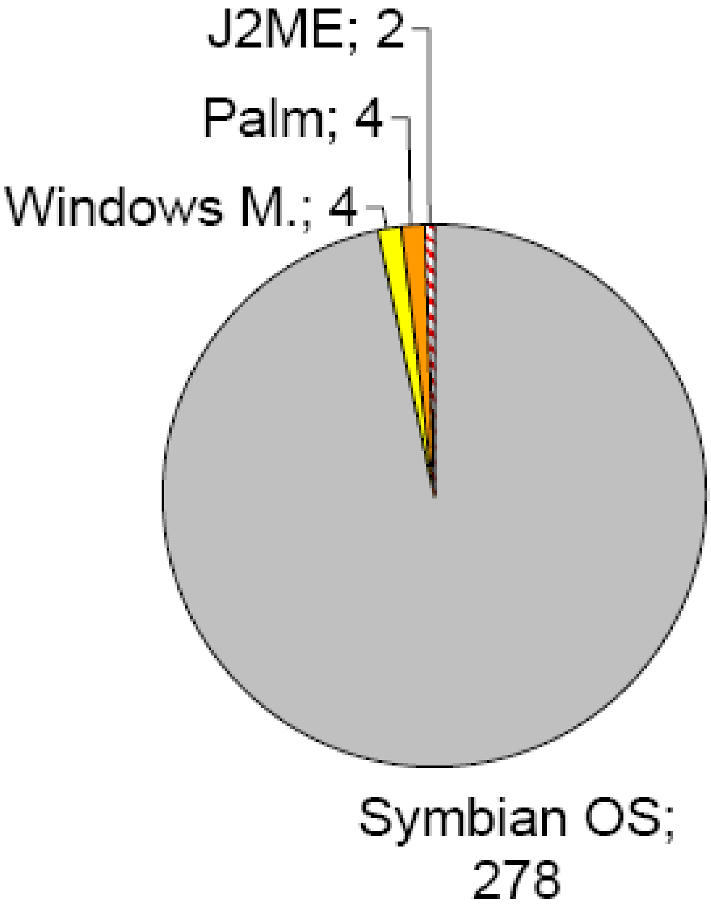


3. Smartphone Malware Statistics
Smartphone Malware Propagation



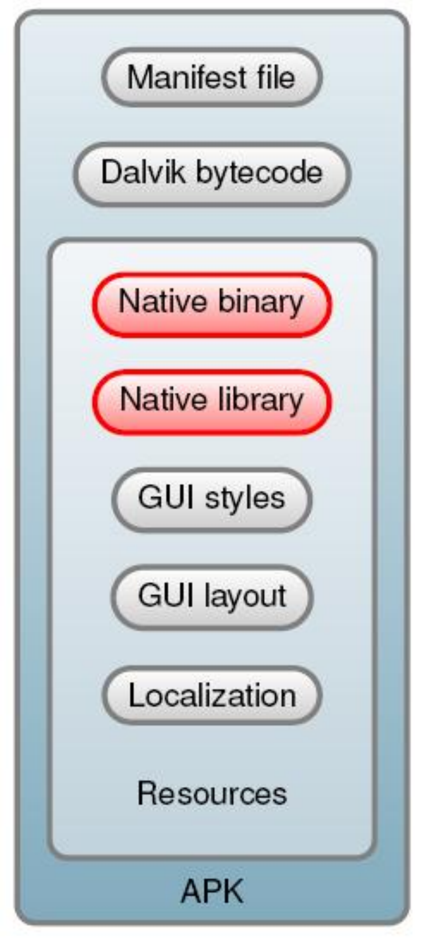
3. Smartphone Malware Statistics Additional Information





Android: Next Target?

- Android
  - Main parts are set open source
- Malware for Android
  - bypass permission system
  - Linux for malicious payload
  - Can reboot a „rooted“ device
    - „Loop of death“ through daemon
- Additionally: always keep an an eye on <u>Collin Mulliner’s</u> work



Agenda

1. Introduction
2. Background
3. Smartphone Malware Statistics
4. Countermeasures
5. Conclusion

4. Countermeasures - AV

- Antivirus software
  - Startet with simple pattern matching
    - 16 Byte were enough
  - Was extended by Wildcard approach
  - Was extended by Mismatch approach
- Smartphone antivirus **limited to signature-based** approaches

4. Countermeasures – AV Improvements: 1st Gen.

DAI-Labor
TU Berlin

- Hashing
  - Increase speed of comparison
- Generic detection
  - (in most cases simple single string detecting variants)
- Bookmarks
  - Distance start of virus body to matching string
- Top and Tail Scanning
  - Scan first or last bytes ->early viruses mod. these areas
- Entry-point and Fixed-point scanning
  - Start scanning in seperate areas
- Hyperfast scanning
  - Access hd via bios bypassing OS-level API

4. Countermeasures – AV Improvements: 2nd Gen.

- Smart Scanning
  - Ignores nop
- Skeleton Scanning
  - Checks makros line by line for ignoring useless instructions
- Nearly exact identification
  - Two strings to match instead of one
- Exact identification
  - As many as necessary (static) ranges

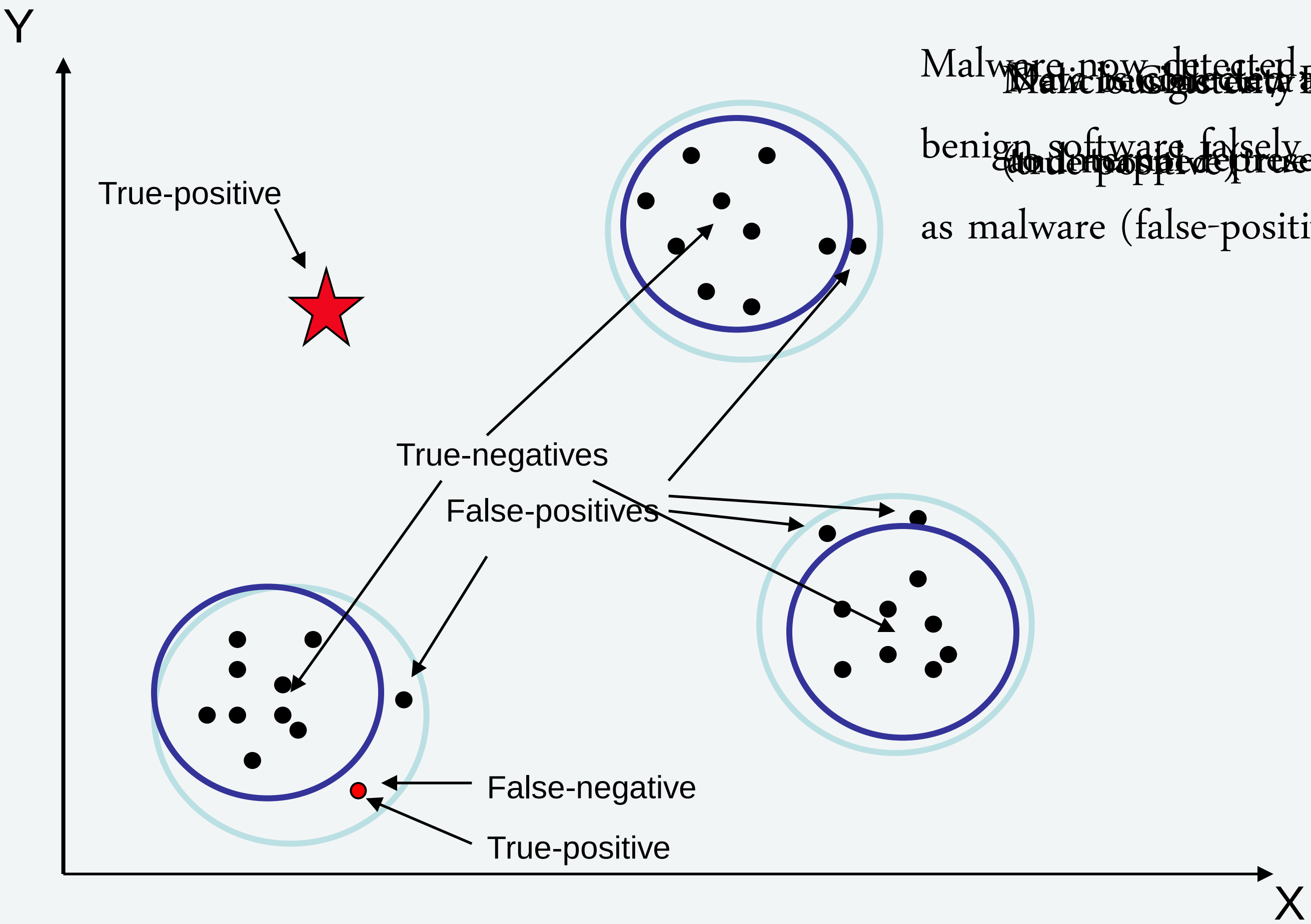
4. Countermeasures – AV Improvements: Further 1/2

- Algorithmic Scanning
  - If standard alg. fails, propietary algorithm is used
  - Formery were hard-coded detection routines
  - Filtering
  - Static decryptor detection
  - Cryptographic detection

4. Countermeasures – AV Improvements: Further 2/2

- Code Emulation
  - Geometric Detection
    - Checks for alteration in file system
  - Heuristic Analysis
    - Basically – behavior-based
- Disassembling
- Heuristic Analysis using Neural Networks
  - Basically applied AI, feature-based

4. Countermeasures – Anomaly Detection



Malware now detected, but new benign software falsely detected as malware (false-positive) [illegible]

4. Countermeasures – Static Function Call Analysis 1/2

- Current solutions use **signatures**
  - Vulnerable to **new/unknown** malware
  - Vulnerable to **old** malware
- **Function call** analysis can be valuable extension
  - Check **similarity** to benign applications
  - **Light-weight** algorithms
  - **High** detection rates

4. Countermeasures – Static Function Call Analysis 2/2

1. Function calls are extracted
   - From common benign software
   - From installed application
2. Function calls are compared
   - Simple string matching for occurrences
3. Occurences are checked for
   - Clusters
   - Statistics

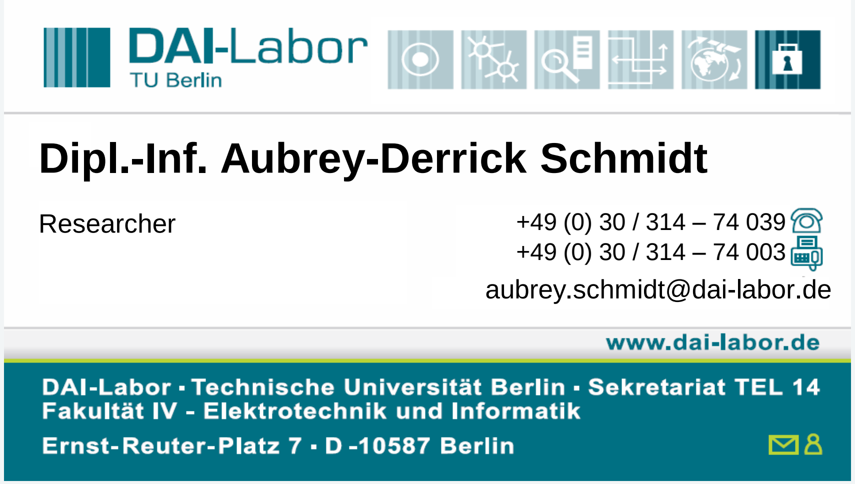
Agenda

1. Introduction
2. Background
3. Smartphone Malware Statistics
4. Countermeasures
5. Conclusion

5. Conclusion

- Smartphone malware evolution
  - Main target Symbian
  - New platforms still waiting for malware „in the wild“
  - Countermeasures on smartphones currently limited to signature-based approaches
    - Our research shows that static analysis might be an interesting addition

Kontakt



DAI-Labor
TU Berlin

Dipl.-Inf. Aubrey-Derrick Schmidt

Researcher

+49 (0) 30 / 314 – 74 039 ☎
+49 (0) 30 / 314 – 74 003 📠
aubrey.schmidt@dai-labor.de

www.dai-labor.de

DAI-Labor · Technische Universität Berlin · Sekretariat TEL 14
Fakultät IV - Elektrotechnik und Informatik
Ernst-Reuter-Platz 7 · D -10587 Berlin ✉