Enter Sandbox: Android Sandbox Comparison

Sebastian Neuner, Victor van der Veen,

Martina Lindorfer, Markus Huber,

Georg Merzdovnik, Martin Mulazzani and Edgar Weippl

Overview

- In a nutshell
  - Static analysis
  - Dynamic analysis
  - Combined approach
- Motivation
- Contributions
  - Evaluated sandboxes
  - Interdependency
  - Sandbox effectiveness
- Summary

Analysis in a Nutshell - Static

- Static Analysis
  - Check code against rules
    - Source is available or
    - Application is disassembled
  - Pros
    - Fast
    - No execution, no risk
  - Con
    - Does not detect runtime specifics

Analysis in a Nutshell - Dynamic

- Dynamic analysis
  - Execute target application
    - Analyse behaviour
    - Observe environment
  - Pro
    - Find runtime specifics (e.g. temporal infos)
  - Cons
    - Complex
    - Risky
    - Code coverage

Combined Approach

- More effective analysis
  - Static + dynamic (hybrid)
  - Example:
    - Static analysis of suspicious sample
    - Build callgraph
    - Detect GUI elements

    →Trigger GUI elements (not randomly but targeted)

    →Taint analysis on base of callgraph

Combined Approach

- More effective analysis
  - Static + dynamic (hybrid)
  - Example:
    - Static analysis of suspicious sample
    - Build callgraph
    - Detect GUI elements

    →Trigger GUI elements (not randomly but targeted)

    →Taint analysis on base of callgraph

secure sba-research.org

Combined Approach

- More effective analysis
  - Static + dynamic (hybrid)
  - Example:
    - Static analysis of suspicious sample
    - Build callgraph
    - Detect GUI elements

    →Trigger GUI elements (not randomly but targeted)

    →Taint analysis on base of callgraph

Sandbox

- Analysis environment for unknown software
  - Virtualized
  - Mostly hybrid
  - Watch network traffic, syscalls and other activities
  - Possible harms in case of malware (for host and guest system)

Motivation

- 1 billion Android devices expected in 2017
- SMSZombie: 500.000 infections (China)
- Too many sandboxes out there
  - Not enough coverage
  - No comparison

Why Compare?

- A lot of sandboxes
  - Which work and are available
  - How are they reused -> **Interdependency**
- Some sandboxes provide novel features
- **No Swiss-Army-Knife**

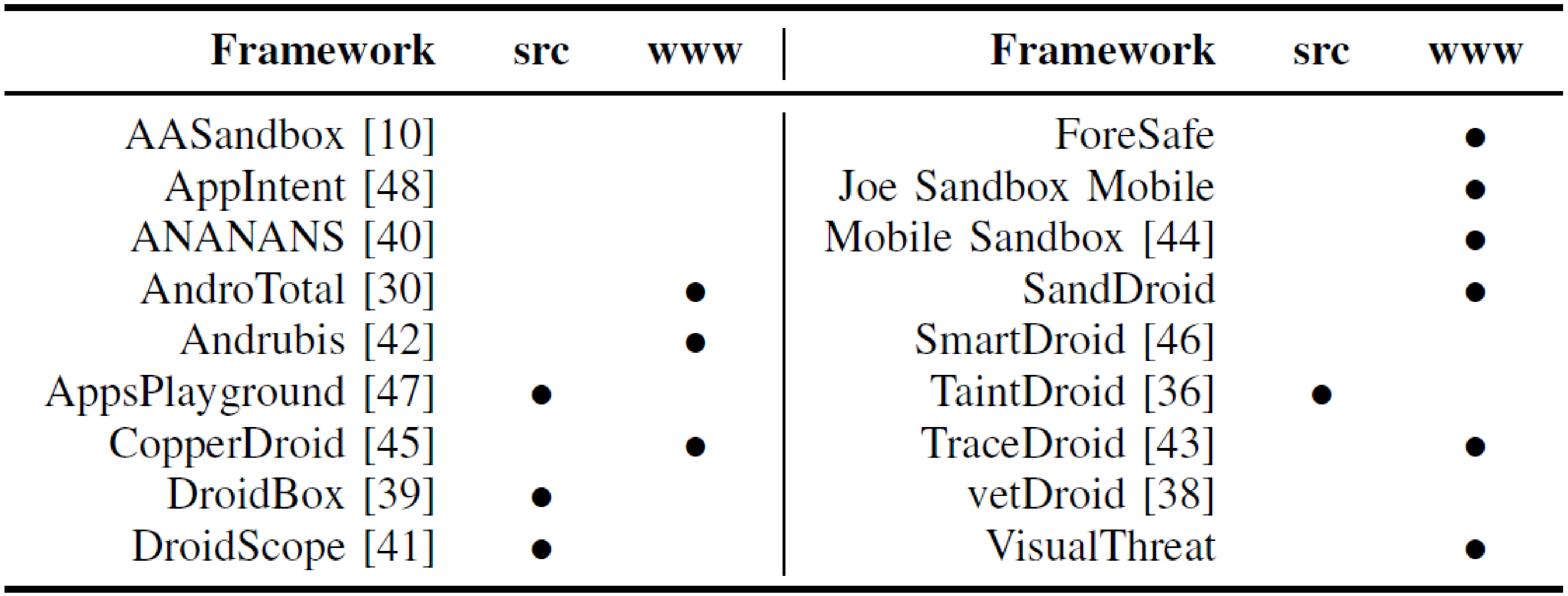
Contributions

- Comparison of 16 available sandboxes
  - Level of introspection
  - Functionality
  - Interdependency
- Discussion of methods to detect and probe dynamic analysis frameworks

Contributions

- Effectiveness of 8 sandboxes
  - Just online (no source downloaded and run)
  - Public malware
  - Master Key vulnerabilities

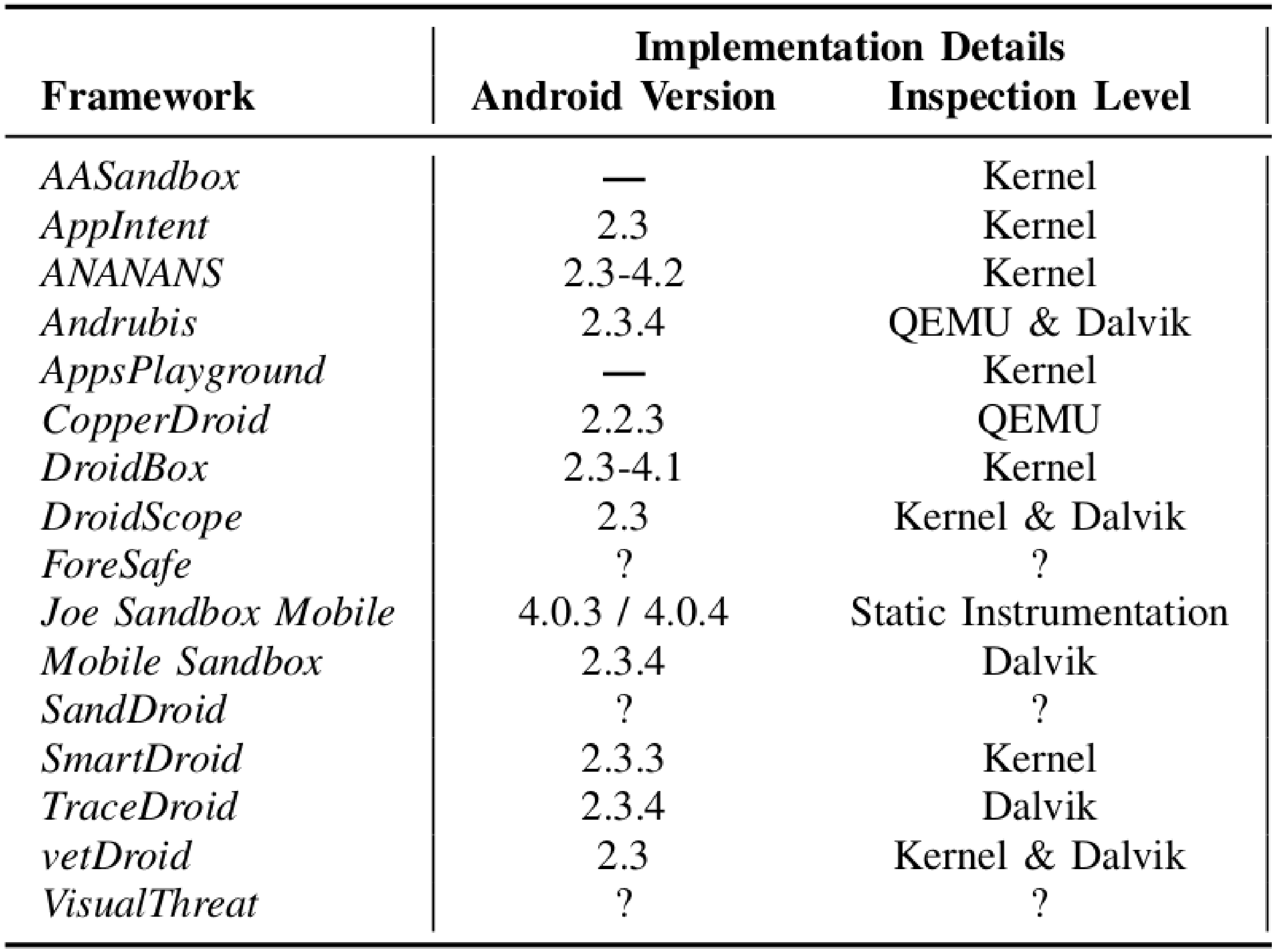
16 Sandboxes



Framework	src	www
AASandbox [10]		
AppIntent [48]		
ANANANS [40]		
AndroTotal [30]		●
Andrubis [42]		●
AppsPlayground [47]	●	
CopperDroid [45]		●
DroidBox [39]	●	
DroidScope [41]	●	
ForeSafe		●
Joe Sandbox Mobile		●
Mobile Sandbox [44]		●
SandDroid		●
SmartDroid [46]		
TaintDroid [36]	●	
TraceDroid [43]		●
vetDroid [38]		
VisualThreat		●

Table 1: Framework availability

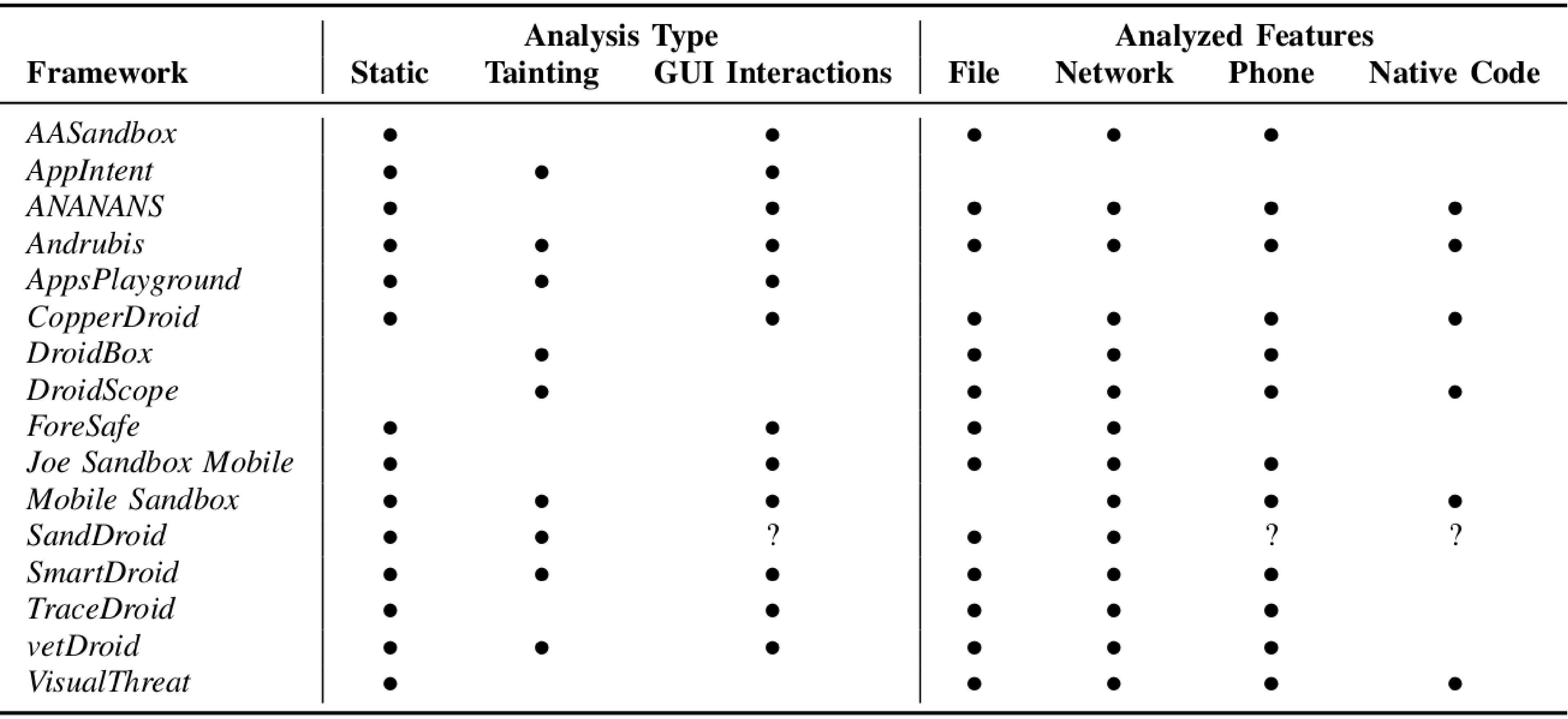
Types of Introspection



	Implementation Details	
Framework	**Android Version**	**Inspection Level**
AASandbox	—	Kernel
AppIntent	2.3	Kernel
ANANANS	2.3-4.2	Kernel
Andrubis	2.3.4	QEMU & Dalvik
AppsPlayground	—	Kernel
CopperDroid	2.2.3	QEMU
DroidBox	2.3-4.1	Kernel
DroidScope	2.3	Kernel & Dalvik
ForeSafe	?	?
Joe Sandbox Mobile	4.0.3 / 4.0.4	Static Instrumentation
Mobile Sandbox	2.3.4	Dalvik
SandDroid	?	?
SmartDroid	2.3.3	Kernel
TraceDroid	2.3.4	Dalvik
vetDroid	2.3	Kernel & Dalvik
VisualThreat	?	?

Table 2: Results. Part 1. „---“ installable on any Android version. „?“: Not possible to determine

Analysis Features



Framework	Analysis Type			Analyzed Features			
	Static	Tainting	GUI Interactions	File	Network	Phone	Native Code
AASandbox	●		●	●	●	●	
AppIntent	●	●	●				
ANANANS	●		●	●	●	●	●
Andrubis	●	●	●	●	●	●	●
AppsPlayground	●	●	●				
CopperDroid	●		●	●	●	●	●
DroidBox		●		●	●	●	
DroidScope		●		●	●	●	●
ForeSafe	●		●	●	●		
Joe Sandbox Mobile	●		●	●	●	●	
Mobile Sandbox	●	●	●		●	●	●
SandDroid	●	●	?	●	●	?	?
SmartDroid	●	●	●	●	●	●	
TraceDroid	●		●	●	●	●	
vetDroid	●	●	●	●	●	●	
VisualThreat	●			●	●	●	●

Table 2: Results. Part 2

Probing

- Benign.apk
  - Unpack with apktool
  - Change min and target SDK version (5, 9, 11, 14, 19, 25)
  - Repackage with apktool
  - Verify new SDKVersion
    - A: android:minSdkVersion(0x0101020c)=(type 0x10)0x19
    - A: android:targetSdkVersion(0x01010270)=(type 0x10)0x19

Sandboxes leaking API level

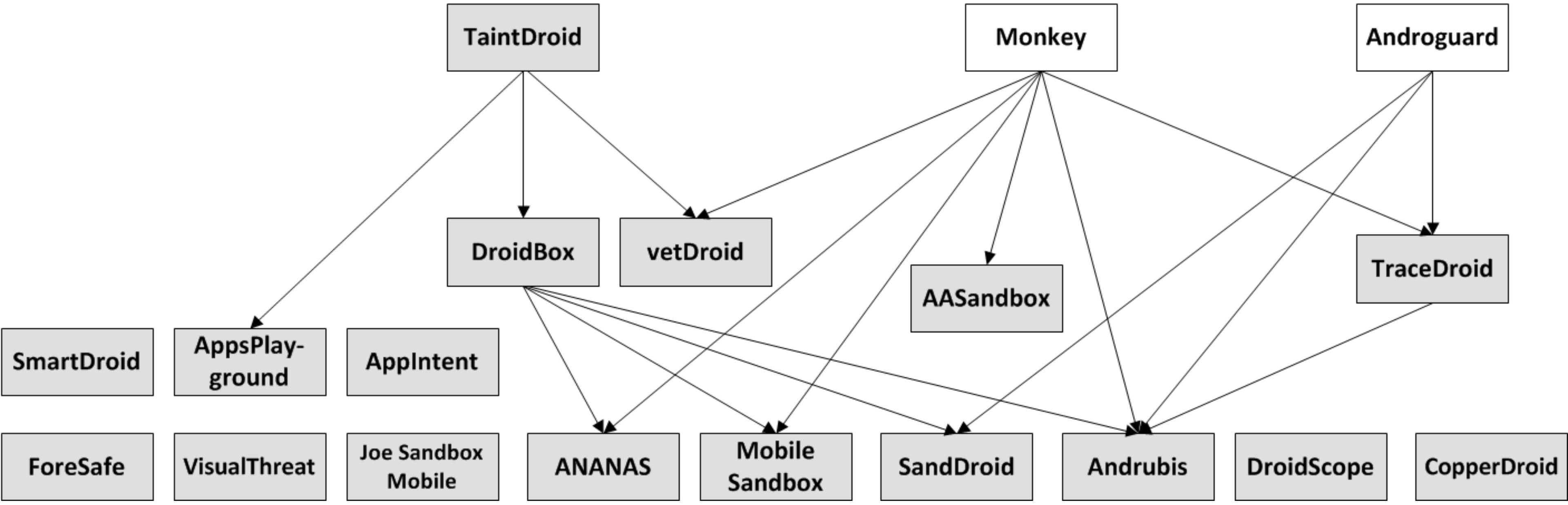
E.g.

„Errors: Setup command ‚_JBInstallAPK' failed: Installation failed: device is running API Level 15, but APK requires 19"

Interdependecy?

- Read documentations
- Read papers
- Emailed with authors
- Uploaded specific samples to see if something crashes :-D

Interdependency!



Effectiveness

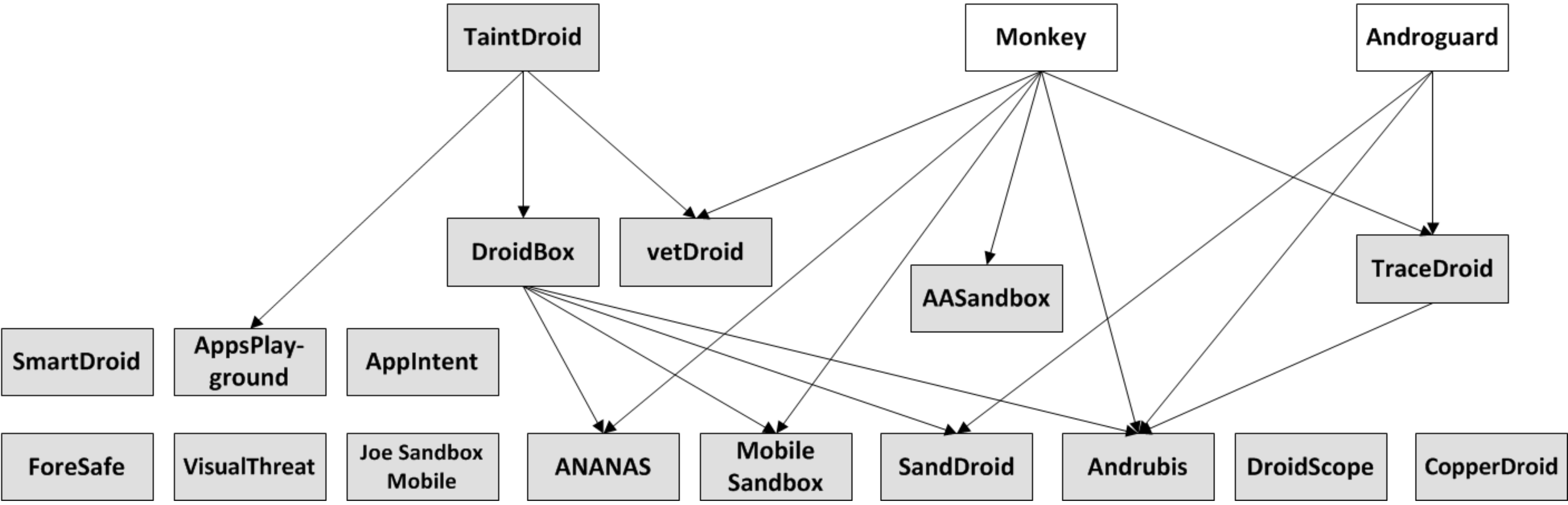
- Chosen malware
  - Public available malware sets:
    - Contagio Mobile
    - Android Malware Genome Project
  - Master Key vulnerabilities
    - Weaknesses in ZIP fileformat handling within Android (→ APK)
  - Python bug for specific zeros in ZIP header

Master Key

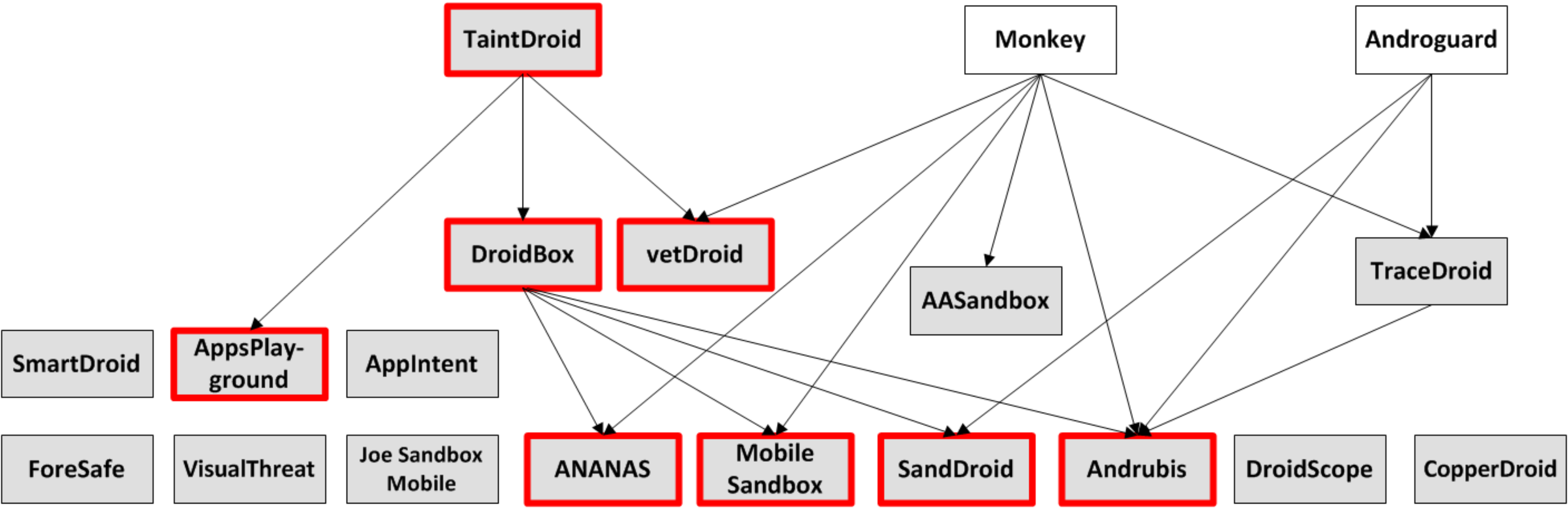
- How these weaknesses influence interdependency?
  - Wrong handling in massive used software

    → Would affect every edge in contact

So this would become…



…this


TaintDroid
Monkey
Androguard
DroidBox
vetDroid
AASandbox
TraceDroid
SmartDroid
AppsPlay-ground
AppIntent
ForeSafe
VisualThreat
Joe Sandbox Mobile
ANANAS
Mobile Sandbox
SandDroid
Andrubis
DroidScope
CopperDroid

Sample Selection

- Coverage (regarding table V in [1]):
  - Remote control
  - Financial charges
  - Personal information stealing

[1] … Y. Zhou and X. Jiang, “Dissecting Android Malware: Characterization and Evolution,” in Proceedings of the 33rd Annual IEEE Symposium on Security and Privacy (S&P), 2012.

Sample Origin

- 6 samples from Malware Genome Project
- 2 sample from private contact
- 4 crafted helloWorld apps

Malware Samples

- Obad
  - Kaspersky Labs: „[...] one of the most sophisticated mobile trojans to date [...]“
  - Part of botnet
  - 24 requested permissions
    - Send SMS
    - Send/receive data over network
    - ...
  - (Out of date) anti-emulation techniques
  - From: Malware Genome Project

Malware Samples

- Geinimi
  - Sending SMS
  - Phone calls
  - Total remote control

  - From: Malware Genome Project

Malware Samples

- DroidKungFu
  - Various privilege escalation techniques
    - RageAgainstTheCage
  - Reads IMEI and other sensitive data
  - Send data over network

  - From: Malware Genome Project

Malware Samples

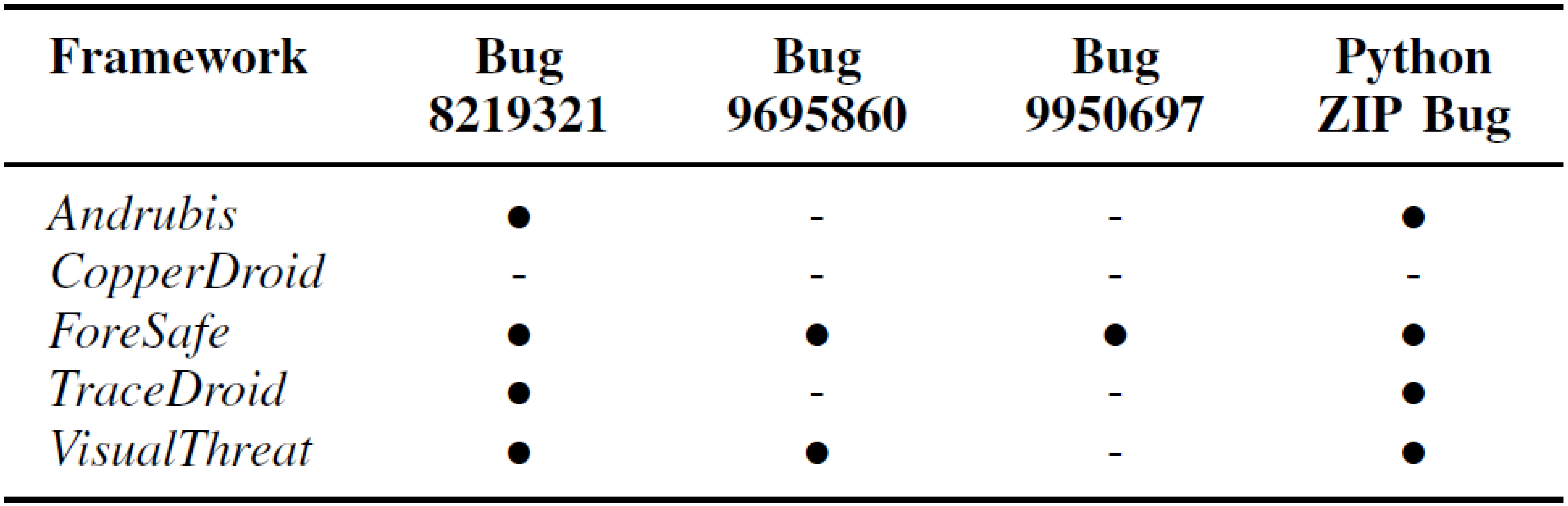
- Basebridge/Nyleaker
  - Invalid APK Manifest to evade Androguard
    - Successfully launched against a sandbox
  - From: Andrubis

Results (Again Tables)

Framework	**Obad**	**Geinimi**	**DroidKungFu**	**Basebridge/ Nyleaker**
Andrubis	● / ●	● / ●	● / ●	● / ○
CopperDroid	- / -	● / -	- / ●	- / -
ForeSafe	● / ●	● / ●	● / ●	● / ●
Joe Sandbox Mobile	● / ●	● / ●	● / ●	● / ●
Mobile Sandbox	- / -	- / -	- / -	- / -
SandDroid	- / -	- / -	- / -	- / -
TraceDroid	● / ●	● / ●	● / ●	● / ●
VisualThreat	● / -	● / ●	● / ●	● / ●

Table 3: Evaluation results with malware. Two samples per family

Tables, Tables, Tables...



Framework	Bug 8219321	Bug 9695860	Bug 9950697	Python ZIP Bug
Andrubis	●	-	-	●
CopperDroid	-	-	-	-
ForeSafe	●	●	●	●
TraceDroid	●	-	-	●
VisualThreat	●	●	-	●

Table 4: Evaluation results with Master Key vulnerabilities and the Python ZIP bug

Consequences

- Sandbox authors notified
  - Appreciated by authors
  - A lot of interesting discussions

Summary

1. Some sandboxes are hardly maintained or totally abandoned
2. Some sandboxes do not recognize even well-known malware
3. Interdependency and code reuse could lead to serious problems

Suggestions

- Not feasible
  - Do a qualified code review of every sandbox
  - Share reports to see if sandbox detects well-known malware
  - Build the analysis Swiss-Army-Knife
- Feasible
  - Build a meta-engine that submits a sample to every known sandbox

Thanks for your Time

- Sebastian Neuner
- SBA Research
  - https://www.sba-research.org/
- sneuner@sba-research.org
  - PGP: 0xDE76C43A

secure
sba-research.org