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
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

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

Table 1: Framework availability
# Types of Introspection

<table>
<thead>
<tr>
<th>Framework</th>
<th>Implementation Details</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Android Version</td>
</tr>
<tr>
<td>AASandbox</td>
<td>—</td>
</tr>
<tr>
<td>AppIntent</td>
<td>2.3</td>
</tr>
<tr>
<td>ANANANS</td>
<td>2.3-4.2</td>
</tr>
<tr>
<td>Andrubis</td>
<td>2.3.4</td>
</tr>
<tr>
<td>AppsPlayground</td>
<td>—</td>
</tr>
<tr>
<td>CopperDroid</td>
<td>2.2.3</td>
</tr>
<tr>
<td>DroidBox</td>
<td>2.3-4.1</td>
</tr>
<tr>
<td>DroidScope</td>
<td>2.3</td>
</tr>
<tr>
<td>ForeSafe</td>
<td>?</td>
</tr>
<tr>
<td>Joe Sandbox Mobile</td>
<td>4.0.3 / 4.0.4</td>
</tr>
<tr>
<td>Mobile Sandbox</td>
<td>2.3.4</td>
</tr>
<tr>
<td>SandDroid</td>
<td>?</td>
</tr>
<tr>
<td>SmartDroid</td>
<td>2.3.3</td>
</tr>
<tr>
<td>TraceDroid</td>
<td>2.3.4</td>
</tr>
<tr>
<td>vetDroid</td>
<td>2.3</td>
</tr>
<tr>
<td>VisualThreat</td>
<td>?</td>
</tr>
</tbody>
</table>

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

## Table 2: Results. Part 2

<table>
<thead>
<tr>
<th>Framework</th>
<th>Static</th>
<th>Analysis Type</th>
<th>GUI Interactions</th>
<th>File</th>
<th>Network</th>
<th>Phone</th>
<th>Native Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>AASandbox</td>
<td>●</td>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>AppIntent</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANANANS</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Andrubis</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>AppsPlayground</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td>●</td>
</tr>
<tr>
<td>CopperDroid</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>DroidBox</td>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>DroidScope</td>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>ForeSafe</td>
<td>●</td>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Joe Sandbox Mobile</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Mobile Sandbox</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>SandDroid</td>
<td>●</td>
<td>●</td>
<td>?</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>?</td>
</tr>
<tr>
<td>SmartDroid</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>TraceDroid</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>vetDroid</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>VisualThreat</td>
<td>●</td>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>
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“
Interdependency?

• 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
Sample Selection

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

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)

<table>
<thead>
<tr>
<th>Framework</th>
<th>Obad</th>
<th>Geinimi</th>
<th>DroidKungFu</th>
<th>Basebridge/Nyleaker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andrubis</td>
<td>● / ●</td>
<td>● / ●</td>
<td>● / ●</td>
<td>● / ○</td>
</tr>
<tr>
<td>CopperDroid</td>
<td>- / -</td>
<td>● / -</td>
<td>- / ●</td>
<td>- / -</td>
</tr>
<tr>
<td>ForeSafe</td>
<td>● / ●</td>
<td>● / ●</td>
<td>● / ●</td>
<td>● / ●</td>
</tr>
<tr>
<td>Joe Sandbox Mobile</td>
<td>● / ●</td>
<td>● / ●</td>
<td>● / ●</td>
<td>● / ●</td>
</tr>
<tr>
<td>Mobile Sandbox</td>
<td>- / -</td>
<td>- / -</td>
<td>- / -</td>
<td>- / -</td>
</tr>
<tr>
<td>SandDroid</td>
<td>- / -</td>
<td>- / -</td>
<td>- / -</td>
<td>- / -</td>
</tr>
<tr>
<td>TraceDroid</td>
<td>● / ●</td>
<td>● / ●</td>
<td>● / ●</td>
<td>● / ●</td>
</tr>
<tr>
<td>VisualThreat</td>
<td>● / -</td>
<td>● / ●</td>
<td>● / ●</td>
<td>● / ●</td>
</tr>
</tbody>
</table>

Table 3: Evaluation results with malware. Two samples per family.
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
I has a question...