Boxify: Full-fledged App Sandboxing for Stock Android

Michael Backes, Sven Bugiel, Christian Hammer, Oliver Schranz, Philipp von Styp-Rekowsky

24th USENIX Security Symposium
Motivation

- FlaskDroid [SEC'13]
- TaintDroid [OSDI'10]
- L4Android [SPSM'11]
- I-ARM-Droid [MoST'12]
- Aurasium [SEC'12]
- Cells [SOSP'11]
- AppGuard [TACAS'13]
- CRePE [ISC'10]
- MOSES [SACMAT'12]
- DroidForce [ARES'14]
- RetroSkeleton [MobiSys'13]
- AirBag [NDSS'14]
- Apex [ASIACCS'10]
- Dr. Android & Mr. Hide [SPSM'12]
- ASM [SEC'15]
- TrustDroid [SPSM'11]
- CRePE [ISC'10]
Motivation

FlaskDroid  
[SIC’13]

Cells  
[SOSP’11]

TaintDroid  
[OSDI’10]

L4Android  
[SPSM’11]

Apex  
[ASIACCS’10]

AppGuard  
[TACAS’13]

CRePE  
[ISC’10]

I-ARM-Droid  
[MoST’12]

MOSES  
[SACMAT’12]

AirBag  
[NDSS’14]

RetroSkeleton  
[MobiSys’13]

TrustDroid  
[SPSM’11]

ASM  
[SEC’15]

DroidForce  
[ARES’14]

Dr. Android & Mr. Hide  
[SPSM’12]

Aurasium  
[SEC’12]
Motivation

OS Extensions

- FlaskDroid [SEC'13]
- TaintDroid [OSDI'10]
- CRePE [ISC'10]
- MOSES [SACMAT'12]
- TrustDroid [SPSM'11]

Cells

- [SOSP'11]
- Apex [ASIACCS'10]

L4Android

- [SPSM'11]

Application Layer Solutions

- AppGuard [TACAS'13]
- I-ARM-Droid [MoST'12]

- Dr. Android & Mr. Hide [SPSM'12]

- RetroSkeleton [MobiSys'13]

- Aurasium [SEC'12]

- MOSES [SACMAT'12]
- AirBag [NDSS'14]
- ASM [SEC'15]

- DroidForce [ARES'14]
Android OS Extensions

- System Services
- Binder IPC
- Syscall API
- Linux Kernel
- App
- Process Boundary
- Kernel Boundary
Android OS Extensions

- ✔ Strong security
- ✖ Hard to deploy
Application Layer Solutions

- Easy to deploy
- No app monitoring possible

![Diagram showing layers of application systems with Linux Kernel, Binder IPC, and Syscall API]
Inlined Reference Monitor

- ✔ Easy to deploy
- ❌ Weak security

![Diagram of Inlined Reference Monitor]

- System Services
- App Monitor
- Binder IPC
- Syscall API
- Linux Kernel
- App
Goal

OS Extensions

❌ Hard to deploy

✔ Strong security

Application Layer Solutions

✔ Easy to deploy

❌ Weak security
Goal

**OS Extensions**
- ✗ Hard to deploy
- ✔ Strong security

**Our Goal**
- ✔ Easy to deploy
- ✔ Strong security

**Application Layer Solutions**
- ✔ Easy to deploy
- ✗ Weak security
Objectives

Monitor and constrain untrusted applications

✔ Easy to deploy
  - No firmware modification / root
  - No application modification

✔ Strong security
  - Protected reference monitor
  - Fail-safe defaults
Approach (1)

**Objective:** No firmware modification / root

**Solution:** Regular user-space application
Approach (2)

Objective: No application modification

Solution: Application virtualization
Approach (3)

Objective: Protected reference monitor

Solution: Separate process
Objective: Fail-safe defaults
Approach (4)

Objective: Fail-safe defaults

System Services

Monitor

Shim

App

Binder IPC

Syscall API

Linux Kernel
Objective: Fail-safe defaults
Approach (4)

Objective: Fail-safe defaults

Solution: Isolated process
Isolated Process

- Allows service components to run isolated from the rest of the application

- Isolated processes
  - Have zero permissions
  - Have no access to system services
  - Run with a distinct, transient UID
  - Cannot write to the filesystem
Architecture

- **Boxify**
  - **Target**
    - **App**
  - **Broker**
    - **System Services**
    - **Binder IPC**
    - **Syscall API**
  - **Linux Kernel**
  - **App**
Target

Broker

Target

App

IPC Shim

Sandbox Service

Syscall Shim
Target

- App
  - IPC Shim
  - Sandbox Service
  - Syscall Shim

Broker

Divert Binder IPC to Broker
Target

Divert Syscalls to Broker
Control channel for loading/terminating apps
Loading an app

Broker

Context.bindService()

Binder SandboxService

SandboxService.prepare()

Binder ApplicationThread

ApplicationThread.bindApplication()

Target

Isolated process is created

Shims are set up

App is started
API Layer

Establish compatibility across Android versions
Core Logic Layer

Baseline enforcement & virtual system services
Virtualization Layer

Translate between Boxify and Android system
Virtualization Layer

startActivity(Activity A)

startActivity(Activity 1)

scheduleLaunchActivity(Activity A)

scheduleLaunchActivity(Activity 1)
System Integration

- Launching apps
  - Dedicated Activity

- Installing/Updating apps
System Integration

- Launching apps
  - Dedicated Activity
  - Shortcuts on Home Screen

- Installing/Updating apps
System Integration

- Launching apps
  - Dedicated Activity
  - Shortcuts on Home Screen
  - Virtualized Launcher

- Installing/Updating apps
System Integration

- Launching apps
  - Dedicated Activity
  - Shortcuts on Home Screen
  - Virtualized Launcher

- Installing/Updating apps
  - Directly via App Stores
# Performance

## Middleware Microbenchmark

<table>
<thead>
<tr>
<th>API Call</th>
<th>Native</th>
<th>On Boxify</th>
<th>Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Camera</td>
<td>103.24 ms</td>
<td>104.48 ms</td>
<td>1.24 ms (1.2%)</td>
</tr>
<tr>
<td>Query Contacts</td>
<td>7.63 ms</td>
<td>8.55 ms</td>
<td>0.92 ms (12.0%)</td>
</tr>
<tr>
<td>Insert Contacts</td>
<td>66.49 ms</td>
<td>67.51 ms</td>
<td>1.02 ms (1.5%)</td>
</tr>
<tr>
<td>Delete Contacts</td>
<td>75.86 ms</td>
<td>76.81 ms</td>
<td>0.95 ms (0.9%)</td>
</tr>
</tbody>
</table>
# Performance

## Syscall Microbenchmark

<table>
<thead>
<tr>
<th>System Call</th>
<th>Native</th>
<th>On Boxify</th>
<th>Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>create</td>
<td>47.2 µs</td>
<td>162.4 µs</td>
<td>115.2 µs</td>
</tr>
<tr>
<td>open</td>
<td>9.5 µs</td>
<td>122.7 µs</td>
<td>113.2 µs</td>
</tr>
<tr>
<td>remove</td>
<td>49.5 µs</td>
<td>159.6 µs</td>
<td>110.1 µs</td>
</tr>
<tr>
<td>mkdir</td>
<td>88.4 µs</td>
<td>199.4 µs</td>
<td>111.0 µs</td>
</tr>
</tbody>
</table>
## Performance

### Benchmark Tools

<table>
<thead>
<tr>
<th>Tool</th>
<th>Native</th>
<th>On Boxify</th>
<th>Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>CF Bench</td>
<td>16082 Pts</td>
<td>15376 Pts</td>
<td>4.3%</td>
</tr>
<tr>
<td>Geekbench</td>
<td>1649 Pts</td>
<td>1621 Pts</td>
<td>1.6%</td>
</tr>
<tr>
<td>PassMark</td>
<td>3674 Pts</td>
<td>3497 Pts</td>
<td>4.8%</td>
</tr>
<tr>
<td>Quadrant</td>
<td>7820 Pts</td>
<td>7532 Pts</td>
<td>3.6%</td>
</tr>
</tbody>
</table>
Discussion & Limitations

- Cancels Android’s own access control checks
- Violates Principle of Least Privilege
- Full kernel attack surface available
- Presence of Boxify detectable
Use Cases

- Instantiate OS extensions at application layer
  - Fine-grained access control
  - Information flow control
  - Dual-persona, BYOD
  - Dynamic analysis
  - Automated testing
  - Xposed
  - ...
Conclusion

- Lightweight application virtualization for stock Android
- No root or app modification required
- Low runtime performance overhead
- Wide range of applications

Thank you!