Graphical User Interface for Virtualized Mobile Handsets

Janis Danisevskis, Michael Peter, Jan Nordholz, Matthias Petschick, Julian Vetter

Security in Telecommunications
Technische Universität Berlin

MoST San José
May 21st, 2015
Bring You Own Device

Business Phone Policy (possibly)

- Restricted set of apps
- Restricted internet access (VPN/Firewall)
- Remote provisioning
Bring You Own Device

Private Phone Policy (likely)
This is my phone, so I do whatever I want. And, don’t meddle with my stuff.
Our approach on BYOD
Our approach on BYOD

Hypervisor/Microkernel
Our approach on BYOD

- Secure GUI (Trusted Path)
- Secure Virtual GPU

Speaker: Janis Danisevskis
Our approach on BYOD

- Secure GUI (Trusted Path)
- Secure Virtual GPU

Speaker: Janis Danisevskis
Challenges addressed by this work

Threat Model

Private side is under the control of an attacker
- Impersonation attacks
- Eavesdropping attacks
- Evasion of isolation

Corporate Login
Username:
Password:
Challenges addressed by this work

Threat Model

Private side is under the control of an attacker

- Impersonation attacks
- Eavesdropping attacks
- Evasion of isolation

Corporate Email App

From: Your Boss
Subject: New Acquisition
Transfer $gazillion
to account no: xxxevilxxxx

Your Boss
Challenges addressed by this work

Threat Model

Private side is under the control of an attacker
- Impersonation attacks
- Eavesdropping attacks
- Evasion of isolation

- Keylogging/
  Logging of touch events
- Spying on screen output
Challenges addressed by this work

Threat Model

Private side is under the control of an attacker
- Impersonation attacks
- Eavesdropping attacks
- Evasion of isolation

DMA devices can threaten isolation


Challenges addressed by this work

threat Model
Private side is under the control of an attacker
- Impersonation attacks
- Eavesdropping attacks
- Evasion of isolation

Design Goals
- High graphics performance
- Low impact on CPU load
- Low impact on the TCB
Challenges addressed by this work

**Threat Model**
- Private side is under the control of an attacker
  - Impersonation attacks
  - Eavesdropping attacks
  - Evasion of isolation

**Design Goals**
- High graphics performance
- Low impact on CPU load
- Low impact on the TCB

**Design and Implementation**
- Secure GUI (Trusted path)
- Secure Mobile GPU Virtualization
Secure GUI (Trusted Path)
Secure Virtual GPU
Evaluation
Conclusion

Output label

Private
Business
Screen is split into label region and client region

Speaker: Janis Danisevskis
Client VMs have private framebuffers
Label controlled by the switcher indicates output routing
Zero copy and composition in hardware
Motivation
Secure GUI (Trusted Path)
Secure Virtual GPU
Evaluation
Conclusion

Graphical User Interface for Virtualized Mobile Handsets

display controller
driver
framebuffer
switch
input driver
input switch
event == !

client 1 VM
client 2 VM

policy master
decision maker

vsync interrupt input events output data

Speaker: Janis Danisevskis
Summary: Secure GUI

- Unforgeable labels
  → prevents impersonation
- Private framebuffers and exclusive input routing
  → prevent eavesdropping
- Zero copy with hardware overlays
  → low CPU load and low complexity
Motivation
Secure GUI (Trusted Path)
Secure Virtual GPU
Evaluation
Conclusion

Mobile GPU Driver Stack

- **User-space driver**
  - Provides: OpenGL/EGL abstraction
  - Comprises: shader compiler, linker, ...

- **Kernel-space driver**
  - Schedules rendering tasks
  - Protects memory
**Mobile GPU Driver Stack**

- **User-space driver**
  - Provides: OpenGL/EGL abstraction
  - Comprises: shader compiler, linker, ... 

- **Kernel-space driver**
  - Schedules rendering tasks
  - Protects memory

---

Speaker: Janis Danisevskis

Graphical User Interface for Virtualized Mobile Handsets 11/20
Mobile GPU Driver Stack

- User-space driver
  - Provides: OpenGL/EGL abstraction
  - Comprises: shader compiler, linker, ...

- Kernel-space driver
  - Schedules rendering tasks
  - Protects memory

Speaker: Janis Danisevskis

Graphical User Interface for Virtualized Mobile Handsets 11/20
User-space driver unmodified

- User-kernel interface unmodified
- Custom protocol between GPU driver stub and GPU server
  - No forwarding of high bandwidth data, such as textures, attribute lists, or shader programs
  - Forwards job requests to the GPU server (and job completion notifications to the client)
  - Forwards mapping requests to the GPU server
User-space driver unmodified
User-kernel interface unmodified
Custom protocol between GPU driver stub and GPU server
- No forwarding of high bandwidth data, such as textures, attribute lists, or shader programs
- Forwards job requests to the GPU server (and job completion notifications to the client)
- Forwards mapping requests to the GPU server
Mobile GPU Driver Stack (paravirtualized)

- User-space driver unmodified
- User-kernel interface unmodified
- Custom protocol between GPU driver stub and GPU server
  - No forwarding of high bandwidth data, such as textures, attribute lists, or shader programs
  - Forwards job requests to the GPU server (and job completion notifications to the client)
  - Forwards mapping requests to the GPU server
Mobile GPU Driver Stack (paravirtualized)
Prototype

Hardware
Samsung Galaxy SIII
- Exynos4412 SoC
- $4 \times$ ARM Cortex A9 @ 1.4 GHz
- ARM Mali 400 MP4 GPU

Software
- Fiasco.OC (based on rev. 38)
- L4Re (based on rev. 38)
- L4Linux (based on Linux 3.0.101)
- Cyanogenmod CM-10.1.3
## TCB impact

<table>
<thead>
<tr>
<th>Module</th>
<th>SLOC¹</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU-RG²</td>
<td>2,679</td>
</tr>
<tr>
<td>display driver</td>
<td>2,382</td>
</tr>
<tr>
<td>framebuffer switch</td>
<td>548</td>
</tr>
<tr>
<td>input driver</td>
<td>710</td>
</tr>
<tr>
<td>input switch</td>
<td>539</td>
</tr>
<tr>
<td>total</td>
<td>6,858</td>
</tr>
</tbody>
</table>

¹Source lines of code measured with David A. Wheeler’s “SLOCCount”
²GPU-RG: Name of our GPU-server (RG is for resource governor)
Performance evaluation — experiments

Native
Cyanogenmod on Linux on bare metal

Pass-through
Cyanogenmod on L4Linux on Fiasco.OC
GPU driven by the guest kernel

GPU-RG
Cyanogenmod on L4Linux on Fiasco.OC
GPU driven by GPU-RG
Performance evaluation — benchmarks

Cube, Blending, Fog, and Teapot are part of the 0xbench [1] benchmark suite. Quake III is the FOUR.DM_68 demo of QuakeIII Arena run with QIIIA4A [2].
Performance evaluation — benchmarks

![Graph showing performance evaluation benchmarks for Cube unsynced.]
Job Submission and Notification cost

<table>
<thead>
<tr>
<th>experiment</th>
<th>GP(^1) [(\mu\text{s})]</th>
<th>PP(^1) [(\mu\text{s})]</th>
</tr>
</thead>
<tbody>
<tr>
<td>native submit</td>
<td>15.0</td>
<td>25.2</td>
</tr>
<tr>
<td>pass-through submit</td>
<td>22.1</td>
<td>34.9</td>
</tr>
<tr>
<td>notify</td>
<td>3.6</td>
<td>3.2</td>
</tr>
<tr>
<td>GPU-RG submit</td>
<td>47.3</td>
<td>67.5</td>
</tr>
<tr>
<td>notify</td>
<td>52.8</td>
<td>49.7</td>
</tr>
</tbody>
</table>

**Takeaway:**
To meet a job submission rate of 60 Hz, an additional 2.3 % of CPU utilization is incurred on one CPU core.

\(^1\)The ARM Mali 400 MP4 GPU has a geometry processor (GP) and 4 pixel presenters (PP)
Secure GUI (Trusted Path) addresses:
- Impersonation attacks
- Eavesdropping attacks
- Impact on CPU load and TCB

Secure GPU virtualization addresses:
- Enforced isolation of GPU jobs
- Low overhead for GPU jobs
- Low impact on TCB
Questions?
References I

[1] 0xbench.
https://code.google.com/p/0xbench/.


http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0972,01 1014.

Dark side of the shader: Mobile gpu-aided malware delivery.

Cloudburst.
Black Hat USA June, 2009.