Lecture 4

L4Android: A Generic Operating System Framework for Secure Smartphones

Matthias Lange, Adam Lackorzynski et al.

Operating Systems Practical

23 October, 2013

Contents

Context

Proposed solution

Fiasco.OC

L4Re

L4Android

Evaluation

Keywords

Questions

Outline

Context

Proposed solution

Fiasco.OC

L4Re

L4Android

Evaluation

Keywords

Questions

Context

- Ubiquity of smartphones
- Need for secure apps
  - Near Field Communication
  - SIM cards
- Inherent lack of security in smartphone software

Security issues: delayed updates

- Mainline Android development: done by Google
- Phone vendors deploy customized Android versions
- “Maintenance nightmare”:
  - Provide periodic updates that fix vulnerabilities
  - Or no updates at all (that would cost too much)

Security issues: Linux kernel

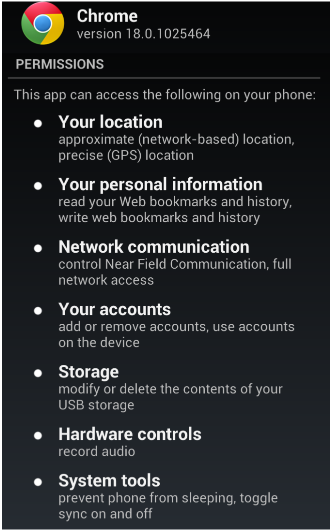
- Monolithic kernels are difficult to verify
- Device drivers run with full privileges
- Kernel components aren't isolated
- Device manufacturers develop custom (often proprietary) drivers

Security issues: rooted phones

- Root privileges allow full access to:
  - all the user data
  - manufacturer settings
  - the kernel
- "Rooted" phones are more vulnerable
  - Android phones don't allow root access by default
- Root access can be obtained
  - manually by the user
  - by malicious software (via exploits)

Security issues: Android permission system

- Permissions in Android
  - based on Mandatory Access Control (MAC)
    - “all or nothing” paradigm
  - too coarse-grained
    - e.g.: grant access to Internet and Address Book
    - → software can send user Address Book to any remote location



Chrome
version 18.0.1025464

PERMISSIONS

This app can access the following on your phone:

- **Your location**
  approximate (network-based) location, precise (GPS) location
- **Your personal information**
  read your Web bookmarks and history, write web bookmarks and history
- **Network communication**
  control Near Field Communication, full network access
- **Your accounts**
  add or remove accounts, use accounts on the device
- **Storage**
  modify or delete the contents of your USB storage
- **Hardware controls**
  record audio
- **System tools**
  prevent phone from sleeping, toggle sync on and off

Outline

Context

Proposed solution

Fiasco.OC

L4Re

L4Android

Evaluation

Keywords

Questions

A solution

- Isolate OS inside a **virtual machine**
- Run secure apps outside the OS
- Use a **microkernel**-based framework
  - ”extended hardware”
  - small Trusted Computing Base (TCB)
  - drivers as userspace services

A solution: L^4Android

- Framework for developing secure smartphone apps
- Components:
  - microkernel: Fiasco.OC μkernel
  - services: L4Re runtime environment
  - kernel: L^4Android
  - userspace: Android libraries, apps, ...

Outline

Context

Proposed solution

Fiasco.OC

L4Re

L4Android

Evaluation

Keywords

Questions

Fiasco.OC: Overview

- Based on Jochen Liedtke's L4 microkernel
- Implements basic OS primitives
  - Address Spaces
  - Threads
  - Scheduling
  - Inter-Process Communication
  - Interrupt Delivery (via Asynchronous IPC)

Fiasco.OC: Protection Domains

- Protection Domains:
  - equivalent to Linux namespaces/containers
  - run as tasks on top of the microkernel
  - provide isolation
    - among virtual machines
    - between VMs and the TCB

Fiasco.OC: Capabilities

- Capabilities provide access to:
  - kernel objects
    - address spaces
    - threads
    - communication channels
  - interrupts
- Fine-grained control over resources

Fiasco.OC: Verification

- Microkernel exposes minimal interface
  - small number of system calls
- Code base is small (~20,000 lines of code)
- Kernel is formally verifiable

Outline

Context

Proposed solution

Fiasco.OC

L4Re

L4Android

Evaluation

Keywords

Questions

L4 Runtime Environment

- Software layer on top of the microkernel
- Simplifies development in microkernel userspace
- Consists of:
  - basic functionality: allocators, data structures etc.
  - user libraries: C, C++, pthread etc.
  - servers providing access to I/O devices

Outline

Context

Proposed solution

Fiasco.OC

L4Re

L4Android

Evaluation

Keywords

Questions

L^4Android

- L^4Linux: Linux kernel modified to run paravirtualized
  - on top of Fiasco.OC + L4Re
  - with fine-grained access to devices via I/O servers
    - an L^4Linux instance can run without any access to peripherals
    - or it can be used as a driver provider
- L^4Android Kernel
  - based on L^4Linux
  - contains Android patches (wakelocks, binder etc.)
  - therefore it is able to run the Android user stack

Outline

Context

Proposed solution

Fiasco.OC

L4Re

L4Android

Evaluation

Keywords

Questions

Evaluation: Overview

- Four proposed scenarios
  - Software Smartcard
  - Mobile Rootkit Detection
  - Hardware Abstraction Layer
  - Unified Corporate and Private Phone
- Last scenario implemented as a demo
- Runnable on ARM and x86 architectures
  - Freescale iMX.51 (Cortex-A8)
  - Aava Mobile developer phone (Moorestown)

Evaluation: Software Smartcard

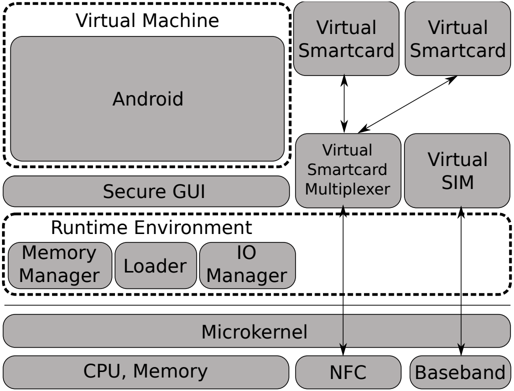
- Smartcard:
  - processor and memory integrated on a plastic card
  - cryptographic coprocessor smarcards for:
    - mobile phones (SIM, NFC)
    - credit cards
    - USB tokens
- "Software smartcard":
  - performing the same computations in software
  - cheaper and more flexible than a physical smartcard
  - usually unfeasible due to high security demands

Evaluation: Software Smartcard (2)

- the Fiasco.OC provides a secure computing base
  - the smartcard operations run on top of the microkernel
  - L4Re and microkernel syscalls offer a trusted interface
  - isolation from the L^4Android domain is achieved
- timing attacks are deflected by secure scheduling
- vendors can implement various virtual smartcard configurations

Evaluation: Software Smartcard

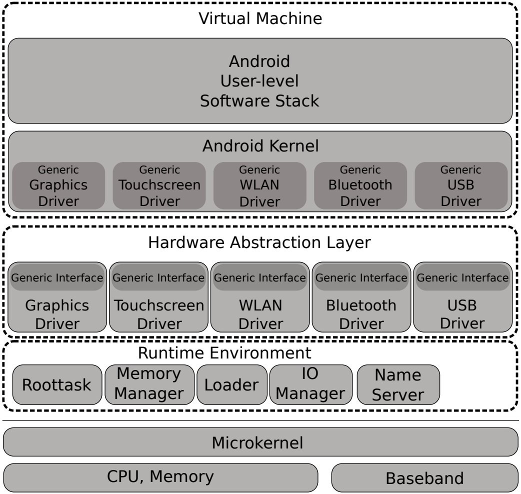
Possible Smartcard setup:



Evaluation: Hardware Abstraction Layer

- HAL: proposed L4-based development model for Linux drivers
- move driver logic to a layer between L4Re and the guest kernel
- develop generic driver stub in the guest OS
    - easier to port drivers to new kernel versions
        - by updating the Linux-HAL interface
    - driver faults are isolated from the rest of the system

Evaluation: Hardware Abstraction Layer



Evaluation: Dual Android Phone

- Corporate smartphones contain sensitive information
- Employees routinely carry two smartphones:
  - a company-provided smartphone configured according to the company's security policy
  - a personal, unrestricted phone

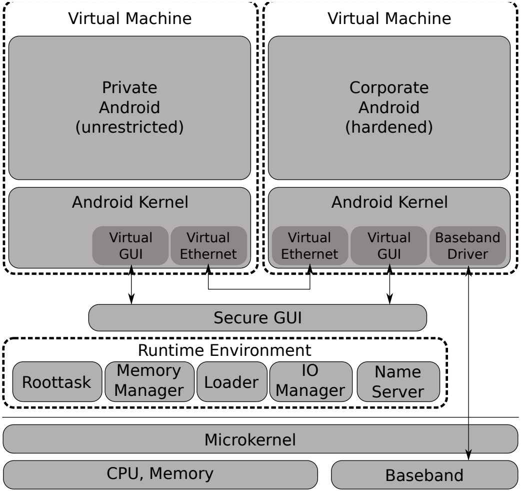
Evaluation: Dual Android Phone (2)

- Solution: a single phone running two Android virtual machines
  - private Android: can even be rooted
  - secure Android: implements corporate security policies
- User can easily switch between instances at runtime

Evaluation: Dual Android Phone (3)

- Access to devices is multiplexed between instances
  - Stub drivers in the guest kernels
  - Driver servers in the L4 Runtime Environment
- Virtualization requirements:
  - secure GUI server
  - virtual Ethernet interfaces
  - mobile telephony, hardware graphics/sound acceleration
    - drivers are binaries in the Linux kernel or Android userspace
    - difficult to virtualize

Evaluation: Dual Android Phone



Outline

Context

Proposed solution

Fiasco.OC

L4Re

L4Android

Evaluation

Keywords

Questions

Keywords

- smartphones
- operating system security
- Mandatory Access Control
- protection domain
- capability
- Trusted Computing Base
- paravirtualization
- microkernel
- L4
- I/O server

- http://l4android.org
- http://l4linux.org
- http://os.inf.tu-dresden.de/L4/
- http://users.sec.t-labs.tu-berlin.de/~steffen/papers/spsm03-lange.pdf
- Jochen Lietdke: On μ-Kernel Construction

Context

Proposed solution

Fiasco.OC

L4Re

L4Android

Evaluation

Keywords

Questions

Questions

?