The Open-Source seL4 Kernel
Military-Grade Security Through Mathematics

Gernot Heiser | gernot.heiser@data61.csiro.au | @GernotHeiser
Trustworthy Systems | Data61

Linaro Connect SFO’17

https://sel4.systems
Car Hacking – What’s Behind?

Networking for:
• Entertainment
• Connected car
• Safety (tire pressure…)
• Maintenance (OTA upgrades)

No security whatsoever on CAN bus!
Challenge of Networking

Networking creates remote attack opportunities
• from passengers (wifi, Bluetooth)
• from nearby cars (wifi, Bluetooth) – drive-by shooting, spread of viruses
• from anywhere (cellular)

Attack vectors:
• Insecure protocols
• Reusing crypto keys

Software vulnerabilities
Software Vulnerabilities

Complexity Drivers
• Features/functionality
• Legacy reuse

Software-engineering rule of thumb:
• 1–5 bugs per 1,000 lines of quality code

Bluetooth protocol stack:
Multiple 100,000 lines

Linux kernel:
Tens of millions lines
Unsafe at any clock speed: Linux kernel security needs a rethink

Ars reports from the Linux Security Summit—and finds much work that needs to be done.

J.M. PORUP (UK) - The enemy will be on the platform!

The Linux kernel today faces an unprecedented safety crisis. Much like when...
OK, So Let’s Patch Regularly

Patch-and-Pray: A losing proposition
So, Let’s Use Firewalls!

- Imposes overhead (SWaP) or
- Runs on vulnerable OS ⇒ worthless if OS compromised
- Even more code – may *increase* attack surface
- No help for valid messages that trigger bugs in software

Firewalls treat symptoms, not causes of problems!
Let’s Use AI to Detect Compromise!

- Runs on vulnerable OS ⇒ worthless if OS compromised
- Even more code – may *increase* attack surface
- Can only detect that system is *already compromised*

Intrusion detection: admission of defeat
Fundamental Security Requirement: Isolation

- Uncritical/untrusted
- Sensitive/critical/trusted

Enforced by trustworthy separation kernel

Strong Isolation

Processor

Communication subject to global security policy
A system is **trustworthy** if and only if:
- it behaves **exactly** as it is specified,
- in a **timely** manner,
- while ensuring **secure** execution

**Claim:**
A system must be considered **untrustworthy** unless **proved** otherwise!

**Corollary [with apologies to Dijkstra]:**
Testing, code inspection, etc. can only show **lack of trustworthiness**!
Provably Secure Operating System

- World's fastest (5–10X faster) operating system designed for security and safety
- Suitable for real-world deployment
- Code that runs in privileged mode of the hardware
  - Most critical part
- Non-kernel code can only access resources and communication channels if explicitly authorised with a per-object access token (capability)
  - Confined damage
  - Least privilege
- ~10,000 lines of C and ASM code
  - Small attack surface, Amenable to full verification
- Small, fast, capability-based, operating system kernel

Unprivileged mode

Privileged mode
sel4: The latest (and most advanced) member of the L4 microkernel family

API Inheritance

Code Inheritance

iOS security processor

sel4

OKL4-µKernel

OKL4-Microvisor

Codezero

Qualcomm modem chips
**Proving Trustworthiness of seL4**

- **Confidentiality**
  - Proof
  - Exclusions (at present):
    - Initialisation
    - Low-level MMU model
    - Caches
    - Multicore
    - Covert *timing* channels

- **Integrity**
  - Proof
  - Abstract Model
  - Functional correctness [SOSP’09]
  - Isolation properties [ITP’11, S&P’13]

- **Availability**
  - Proof
  - Translation correctness [PLDI’13]
  - C Implementation
  - Worst-case execution time [RTSS’11, RTAS’16]
  - Binary code
  - Proving Trustworthiness of seL4

- **Provably impossible:**
  - Buffer overflow
  - Null-pointer dereference
  - Code injection
  - Memory leaks
  - Kernel crash
  - Undefined behaviour
  - Privilege escalation
# How Does seL4 Compare?

<table>
<thead>
<tr>
<th>Feature</th>
<th>seL4</th>
<th>Other hypervisors, RTOSes, separation kernels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>Fastest</td>
<td>2–10 × slower</td>
</tr>
<tr>
<td>Functional correctness</td>
<td>Proved</td>
<td>No Guarantee</td>
</tr>
<tr>
<td>Isolation</td>
<td>Proved</td>
<td>No Guarantee</td>
</tr>
<tr>
<td>Worst-case latency bounds</td>
<td>Sound &amp; complete</td>
<td>Estimates only</td>
</tr>
<tr>
<td>Storage channel freedom</td>
<td>Proved</td>
<td>No Guarantee</td>
</tr>
<tr>
<td>Timing channel prevention</td>
<td>Low overhead</td>
<td>None or High Overhead</td>
</tr>
<tr>
<td>Mixed-criticality support</td>
<td>Fully supported, high utilisation</td>
<td>Limited, resource-wastive</td>
</tr>
</tbody>
</table>
Virtualisation

- User
- Kernel
- Hyp

VM

Guest apps
VMM

guest OS

Syscall

Hypercall

Exception IPC
Uncritical/untrusted:

- Apps
- Linux

Critical control: Extract critical bits, run native

Device driver

NW stack

Incremental process: migrate in pieces

Cyber-retrofit!

Virtual machine for legacy

Uncri@cal/untrusted

Security by Architecture

Linaro Connect SFO'17
Real-World Example: DARPA HACMS

- Retrofit existing system!
- Boeing Unmanned Little Bird
- US Army Autonomous Trucks
- Develop technology
- SMACCMcopter Research Vehicle
- TARDEC GVR-Bot
Example: Communicating Processes

Communication endpoint (port)
Higher-level abstractions of low-level seL4 constructs

Component Middleware: CAmkES

Component

connector

interface

CompA:A

CompB:B

CompC:C

SharedData

AsynchEvent

RPC
Example: Simplified HACMS UAV

Diagram:

- Radio Driver
- Data Link
- Crypto
- CAN Driver
- Uncritical/untrusted, contained
  - Wifi
  - Camera
  - Linux
Enforcing the Architecture

Low-level access rights

Architecture specification language

Compiler/Linker

binary
Open-Source Architecture Analysis

Eclipse-based IDE ➞ AADL ➞ CAmkes ➞ .h, .c ➞ Compile ➞ Binary

Analysis Tools

Safety ✔

Architecture Analysis & Description Language

Generate

Component Description

Eclipse-based IDE

AADL

CAmkes

.h, .c

Glue Code

Design

Generate

Component Description
Military-Grade Security

Cross-Domain Desktop Compositor

Multi-level secure terminal
- Successful defence trial in AU
- Evaluated in US, UK, CA
- Formal security evaluation soon

Pen10.com.au crypto communication device undergoing formal security evaluation in UK
Contributions

I want to contribute to a sel4 project!

- Sign Contributor License Agreement
- Email signed CLA to licensing@sel4.systems

Pull request or patch on mailing list

Review
Thank you

Linaro

arm

Robin Randhawa

Please check out https://sel4.systems
Military-Grade Security for You!

Security is no excuse for poor performance!

Gernot Heiser | gernot.heiser@data61.csiro.au | @GernotHeiser
Linaro Connect SFO’17

http://sel4.systems