Muen

An x86/64 Separation Kernel for High Assurance

Reto Buerki
Adrian-Ken Rueegsegger

Institute for Networked Solutions
21.09.2017

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

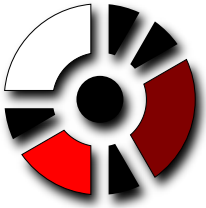
Overview

- Introduction
- Security of Complex Software
- Low Complexity Kernel
- SPARK for Operating Systems
- Conclusion

Who are we?

- MSc in Engineering from HSR
- 9+ years in IT security industry
- Researchers @ INS
- Focus on
  - Development of trustworthy systems
  - Platform security
  - Component-based systems

What are we currently doing?



Muen Separation Kernel

`https://muen.sk`

Examples of Software Failures

- RCE: Symantec Anti-Virus Engine (kernel) (CVE-2016-2208)
- RCE: Cisco ASA (CVE-2016-1287)
- RCE: glibc (CVE-2015-7547)
- Authentication Bypass: Cisco (CVE-2016-1329)
- VM escape: MS Hyper-V (CVE-2016-0088)
- Information disclosure: Heartbleed (CVE-2014-0160)

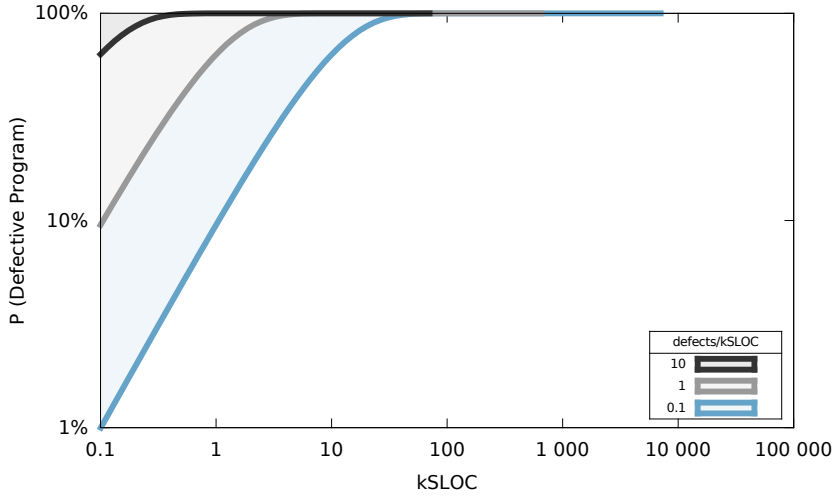
Getting Software right is hard

- Software is inherently complex
  - Accidental difficulty
  - Essential difficulty
- Real world problems are hard to solve
- There is no silver bullet [3]

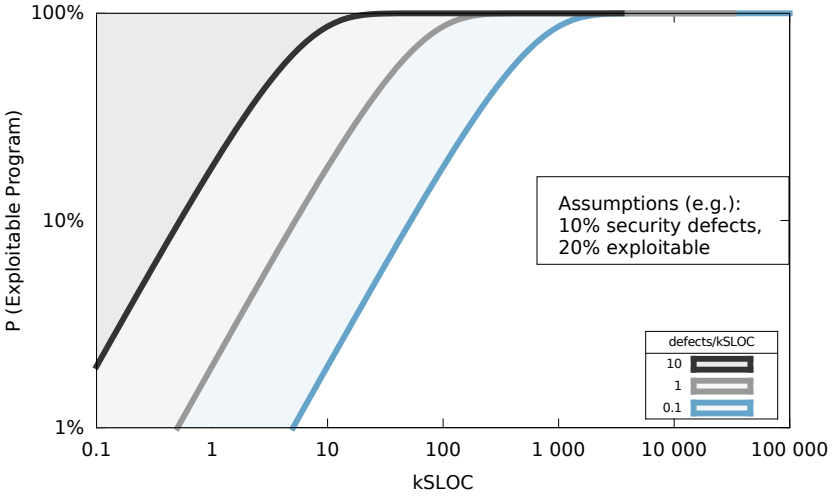
Security of Complex Software

$$P(Program\_Correct) = P(Line\_Correct)^{SLOC} \; [2]$$

Security of Complex Software



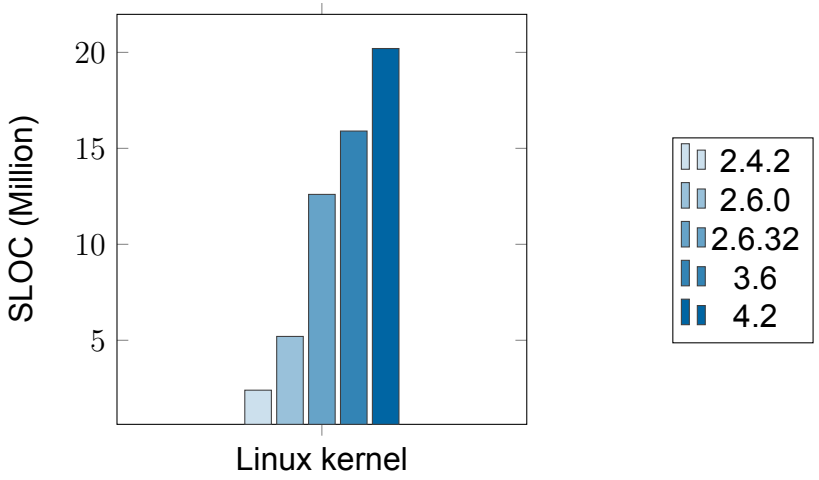
Security of Complex Software



Secure Software

- Tiny size
- *Very* low defect rate
- Low security defect ratio

SLOC

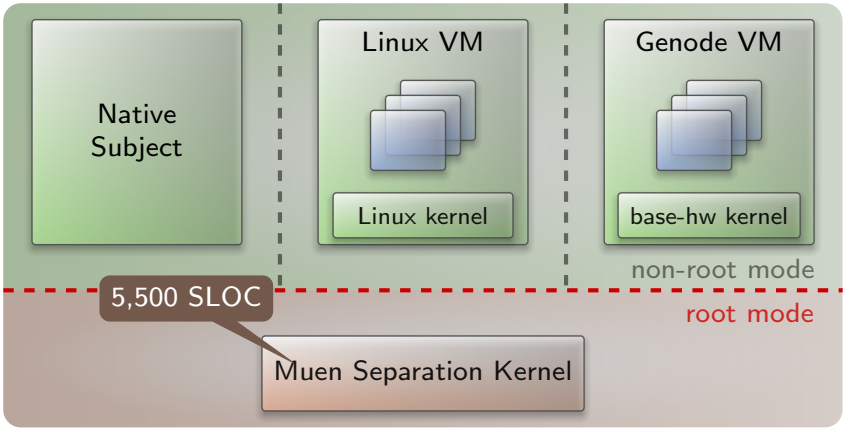


https://en.wikipedia.org/wiki/Source_lines_of_code

Separation Kernel I

- Concept introduced by John Rushby (1981) [6]
- Kernel must guarantee component separation
- Static partitioning and isolation of resources
- Static configuration during integration
- Only includes necessary features
- Well suited for formal verification

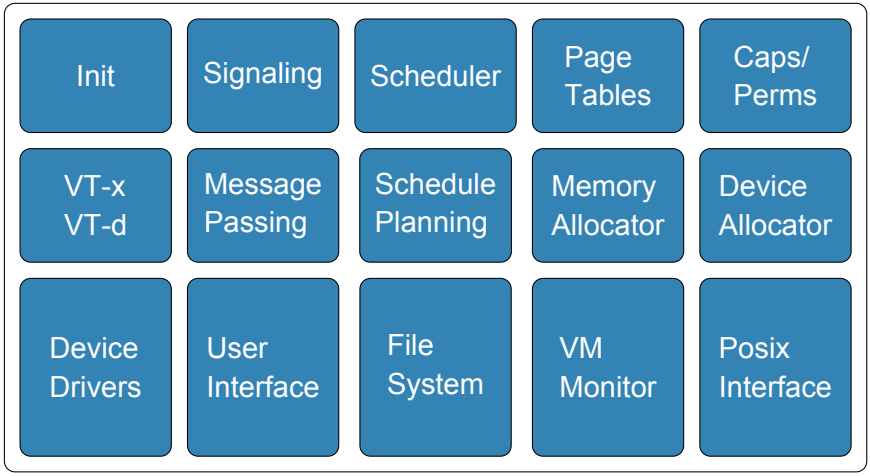
Separation Kernel II



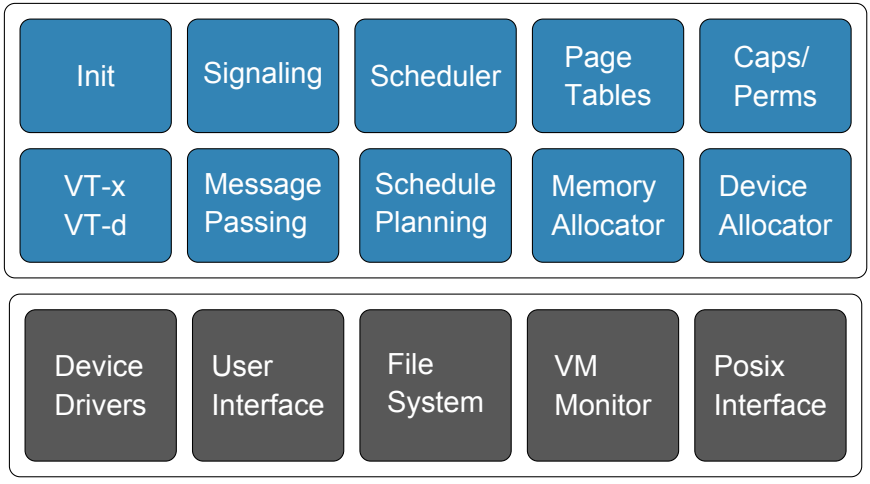
Low Complexity

- In trusted code, every line counts and should be avoided
- Shift complexity out of trusted components
- Design system in order to minimize overall TCB
- KISS
  - Greatly improves readability of code
  - Facilitates effective code review by third party

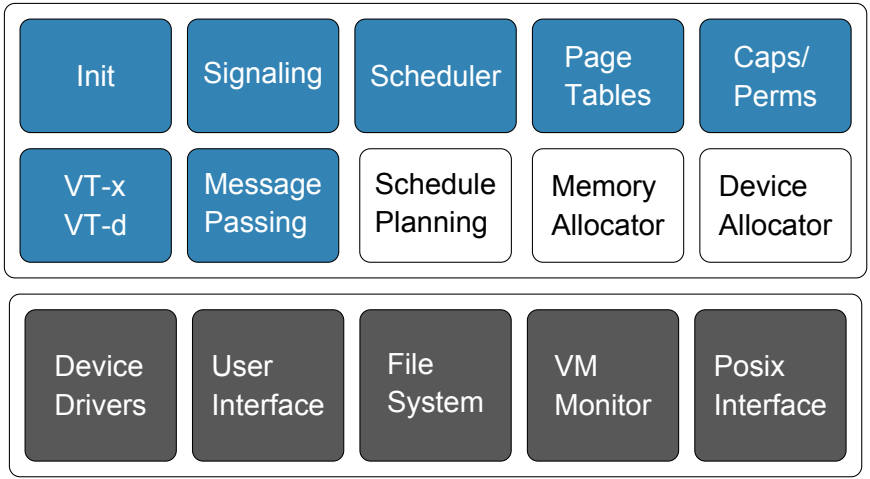
Low Kernel Complexity



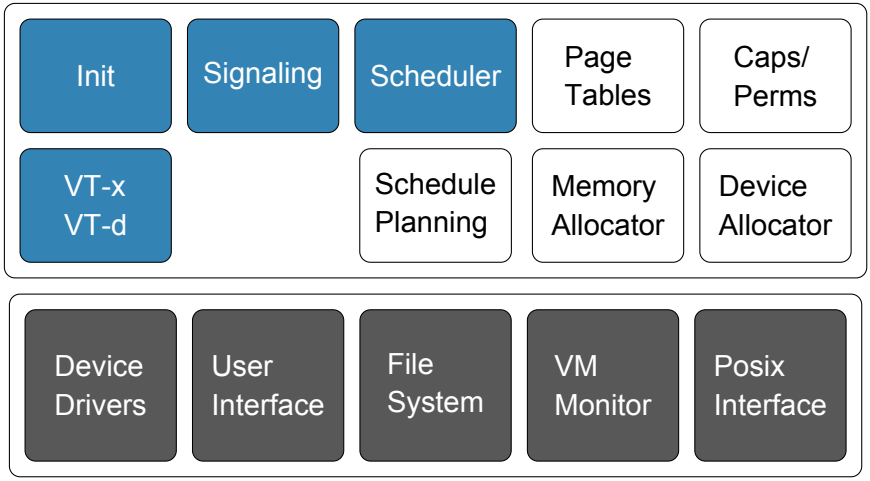
Low Kernel Complexity



Static Resource Allocation



Static Resource Allocation



SPARK 2014 for Operating Systems

- No pointers
- No dynamic memory allocation
- No concurrency

SPARK 2014 for Operating Systems

- No pointers
- No dynamic memory allocation
- No concurrency

- Fixed structures
- Static resource allocation
- One kernel instance / CPU
  Abort on host interrupts

SPARK 2014 for Operating Systems

- No pointers
- No dynamic memory allocation
- No concurrency

- Fixed structures
- Static resource allocation
- One kernel instance / CPU
  Abort on host interrupts

$\rightarrow$ Greatly simplified verification

Lean verification

- Proof annotations are part of the language
- Implicit generation of VCs for integrity preservation (Absence of runtime errors)
- All ARTE VCs proven automatically
- Integration of theorem provers possible *when needed*
- Speed allows proofs to be part of build process

Demo

This presentation is given on a system running on Muen

Summary

- Secure software is limited in complexity
- Separation of untrusted components essential
- Muen provides a solid foundation for high assurance systems
- Muen is the base of complex high security solutions in development
- SPARK 2014 enables lean verification
- Formal verification can be done under commercial constraints

Discussion

Get Muen at `https://muen.sk/`

Getting started with SPARK 2014
`http://university.adacore.com/courses/spark-2014/`

References I

AdaCore and Altran UK Ltd.
SPARK 2014 User's Guide: SPARK Tutorial.
http://docs.adacore.com/spark2014-docs/html/ug/tutorial.html.

Alex Beal.
The Probability of a Correct Program, December 2013.
http://www.usrsb.in/probability-of-a-correct-program.html.

Frederick P. Brooks, Jr.
No Silver Bullet Essence and Accidents of Software Engineering.
Computer, 20(4):10–19, April 1987.

References II

- Hanno Böck, Aaron Zauner, Sean Devlin, Juraj Somorovsky, and Philipp Jovanovic.
  Nonce-disrespecting adversaries: Practical forgery attacks on gcm in tls.
  Cryptology ePrint Archive, Report 2016/475, 2016.
  http://eprint.iacr.org/.

- Reto Buerki and Adrian-Ken Rueegsegger.
  Muen - An x86/64 Separation Kernel for High Assurance.
  Master's thesis, HSR University of Applied Sciences Rapperswil, August 2013.
  https://muen.codelabs.ch/muen-report.pdf.

References III

J. M. Rushby.
Design and Verification of Secure Systems.
SIGOPS Oper. Syst. Rev., 15(5):12–21, December 1981.