Perspectives on security kernels for IoT

IoT expansion and associated threats

Safer through isolation – getting inspired from security kernels for high-critical systems

Strong isolation guarantees for constrained objects: the hopes/the challenges

Nicolas Dejon, *Orange Labs, Université de Lille*

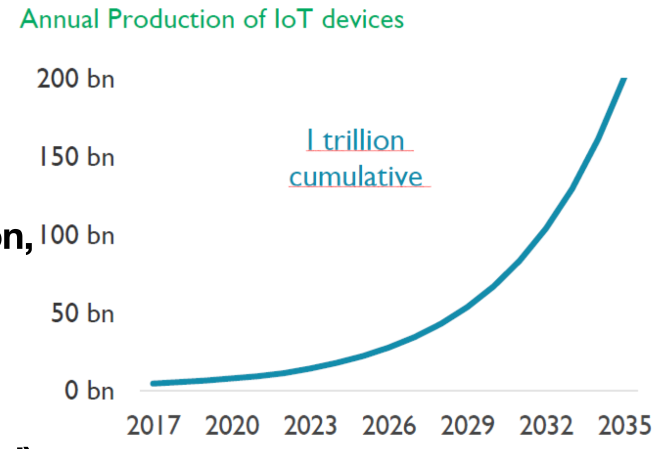
Chrystel Gaber, *Orange Labs*

Gilles Grimaud, *Université de Lille*

16/12/2020

RESSI (Rendez-Vous de la Recherche et de l'Enseignement de la Sécurité des Systèmes d'Information) 2020

IoT expansion and associated threats



Source: SoftBank and ARM estimates

IoT everywhere

- industry, building, energy, agriculture, healthcare, transportation, retail, household appliances…

Broad attack surface

- COTS, SOUP, third-party librairies, networking stack (connected)

Cyberattacks

- Mirai, Stuxnet, hacked Jeep Cherokee, hacked coffee machine
- 300 billion-2 trillion $ losses per year worldwide -> real impacts unknown
- Consequences on safety ? Trust ?

Target is the weakest link

- Heterogeneous IoT devices + limited resources
  - Focus here on constrained objects

Source: hacked coffee machine, Martin Hron

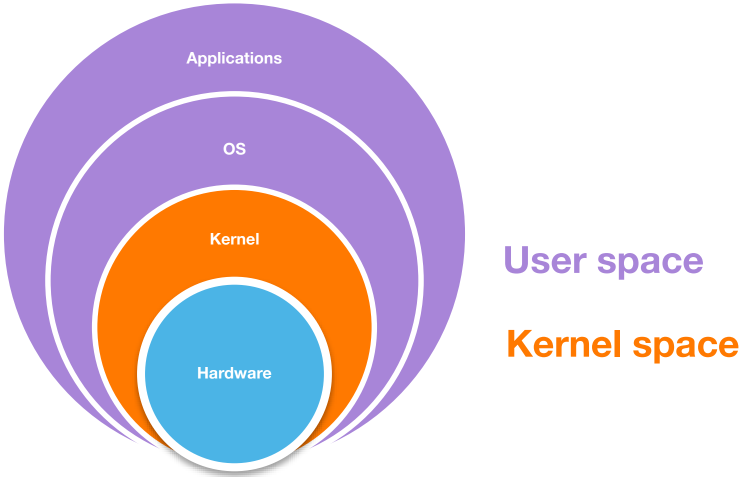
Safer through isolation – getting inspired from security kernels for high-critical systems

Isolation is a mean to security (confidentiality + integrity)

- isolation of critical components
- isolation of flawed/malicious components

- **protect against the memory vulnerabilities class: illegal memory accesses, memory corruption, privilege escalation**

- **security kernel = innermost layer of a system responsible for its security (access to resources) which is correct and isolated**



Safer through isolation – getting inspired from security kernels for high-critical systems

Solutions exist to ensure strict spatial memory isolation

- **FreeRTOS-MPU, EwoK, TockOS, ChoupiOS, RIOT (MPU), Zephyr (MPU), TF-M…**

but are given personnal trust: knowledge of source code or documentation, knowledge of leveraged protection mechanisms, code testing

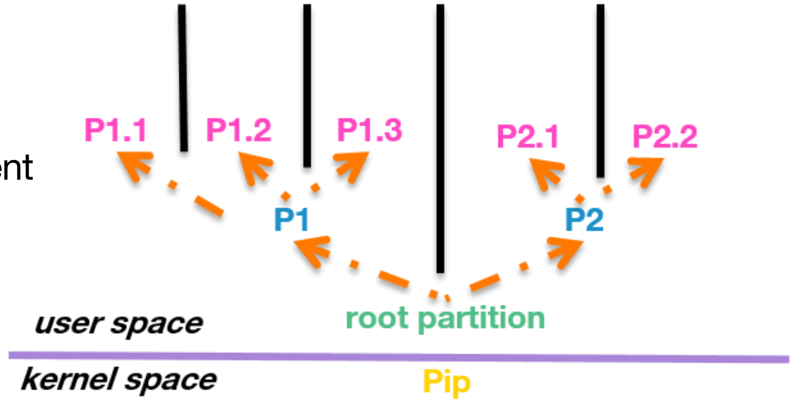
-> Use of formal methods to achieve generalized (mathematical) trust

- the small, simple code becomes a strength

Formally proven kernels

- **seL4, Pip**
  - but need a Memory Management Unit (MMU) not present in constrained devices

-> adapt Pip for constrained devices



Strong isolation guarantees for constrained objects: the hopes/the challenges

From MMU (Memory Management Unit) to MPU (Memory Protection Unit)

- **use of the common memory protection feature**
- **adaptation to the HW constraints: limited MPU regions, restricted memory, less privilege levels**
- **Pip's flexibility**
  - how to deal with the hardware ?
- **adapt the formal proofs (e.g. manual proofs, invariants heavily MMU dependent)**
  - keep the proof efforts low -> what degree of reuse ?
- **ease of adoption, broad use cases**
  - what consequences/modifications on the existing use cases ?
  - what performances to expect ? No tradeoffs for security
- **reach the lowest possible assumptions**
  - common assumptions with formally proven kernels: the boostrapping routine, the hardware platform and the MPU, the source code to machine code tools, the software loader, the theorem prover
  - link with processor specification ?

References

- Radanliev (P.), De Roure (D.), Cannady (S.), Montalvo (R.), Nicolescu (R.) et Huth (M.). – Economic impact of IoT cyber risk - analysing past and present to predict the future developments in IoT risk analysis and IoT cyber insurance. Living in the Internet of Things: Cybersecurity of the IoT - 2018, pp. 3 (9 pp.).
- Ryad Benadjila, Arnauld Michelizza, Mathieu Renard, Philippe Thierry, and Philippe Trebuchet. Wookey: Designing a trusted and efficient USB device. *ACM International Conference Proceeding Series*, pages 673–686, 2019.
- Guillaume Bouffard and Léo Gaspard. Hardening a Java Card Virtual Machine Implementation with the MPU. 2018.
- Emmanuel Baccelli, Oliver Hahm, W Matthias, and Thomas C Schmidt. RIOT OS : Towards an OS for the Internet of Things. pages 2453–2454, 2013.
- Gerwin Klein, Kevin Elphinstone, Gernot Heiser, June Andronick, David Cock, Philip Derrin, Dhammika Elkaduwe, Kai Engelhardt, Rafal Kolanski, Michael Norrish, Thomas Sewell, Harvey Tuch, and Simon Winwood. SeL4: Formal verification of an OS kernel. *SOSP'09 – Proceedings of the 22nd ACM SIGOPS Symposium on Operating Systems Principles*, pages 207–220, 2009.
- Narjes Jomaa, David Nowak, and Paolo Torrini. Formal Development of the Pip Protokernel. 2018.
- FreeRTOS . Website of: Freertos-mpu (freertos). https://www.freertos.org/, FreeRTOS-MPU-memory-protection-unit.htmll, 2020. [Online; accessed November 20, 2020].
- Linaro. Website of: Trustedfirmware-m. https://www.trustedfirmware.org/projects/tf-m/, 2020. [Online; accessed November 20, 2020].
- Tock development team . Website of: Tock. https://www.tockos.org/, 2020. [Online; accessed November 20, 2020].
- Zephyr Project . Website of: The zephyr project. https://www.zephyrproject.org/, 2020. [Online; accessed November 20, 2020].
- Martin Hron (Avast Threat Labs). Website of: https://decoded.avast.io/martinhron/the-fresh-smell-of-ransomed-coffee/, 2020. [Online; accessed November 20, 2020].

Thank you for listening
Merci de votre écoute

Contact details

- **Nicolas Dejon**
  - nicolas.dejon@univ-lille.fr
  - nicolas.dejon@orange.com
- **Chrystel Gaber, PhD**
  - chrystel.gaber@orange.com
- **Prof. Gilles Grimaud**
  - gilles.grimaud@univ-lille.fr