

Experiments on automation of formal verification of devices at the binary level

Thomas Lacroix

INSA Lyon

Soutenance de PFE (Option R&D)

19/06/2019



Section 1

Motivation

Table of Contents

- 1 Motivation
 - Security critical systems
 - Network Interface Controllers (NIC)
 - How to verify hardware devices?
- 2 Non proof-producing verification
 - Subsection 1
 - How trustful is it?
- 3 Proof-producing verification
 - Subsection 1
- 4 Conclusion

Security critical systems

Privacy

- Smartphones
- Smart TVs

Security

- Hospital equipment
- Traffic control systems
- Power plants

Security critical systems - vulnerable

<https://www.wired.com/2014/04/hospital-equipment-vulnerable/>
It's Insanely Easy to Hack
Hospital Equipment

<https://www.wired.com/2015/07/hackers-remotely-kill-jEEP-highway/>
Hackers Remotely Kill a Jeep on
the Highway—With Me in It

Security critical systems - vulnerable

Vulnerabilities come because of:

- Increased surface of attack (more and more features, codebases explode in size)
- Connected to networks → remote attacks

Secure operating systems



¹<https://sel4.systems/Info/FAQ/proof.pml>

Secure operating systems

Formal proof¹:

- The binary code correctly implements its **abstract specification**.
- The specification guarantees **integrity** and **confidentiality**.



¹<https://sel4.systems/Info/FAQ/proof.pml>

Secure operating systems



Formal proof¹:

- The binary code correctly implements its **abstract specification**.
- The specification guarantees **integrity** and **confidentiality**.
- **Integrity**: data cannot be *changed* without permission.
- **Confidentiality**: data cannot be *read* without permission.

¹<https://sel4.systems/Info/FAQ/proof.pml>

Secure operating systems

Proof assumptions:

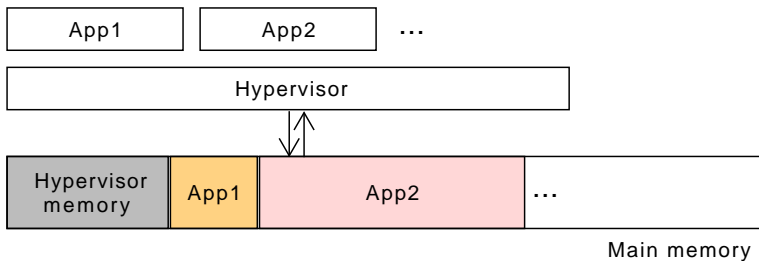


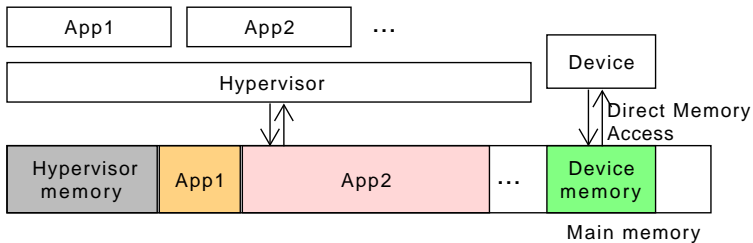
Secure operating systems



Proof assumptions:

- Use of Direct Memory Access (DMA) is excluded, or only allowed for **trusted drivers that have to be formally verified by the user.**





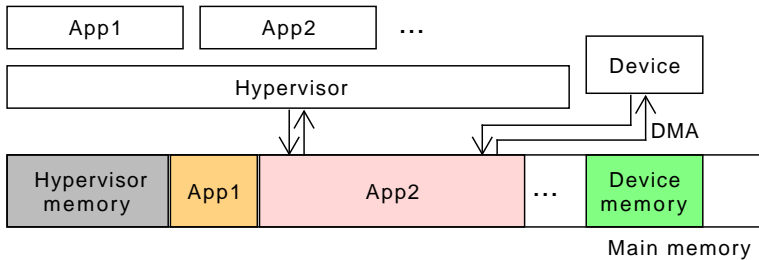


Table of Contents

- 1 Motivation
 - Security critical systems
 - Network Interface Controllers (NIC)
 - How to verify hardware devices?
- 2 Non proof-producing verification
 - Subsection 1
 - How trustful is it?
- 3 Proof-producing verification
 - Subsection 1
- 4 Conclusion

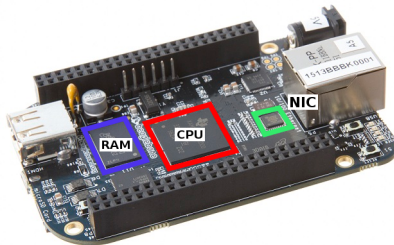
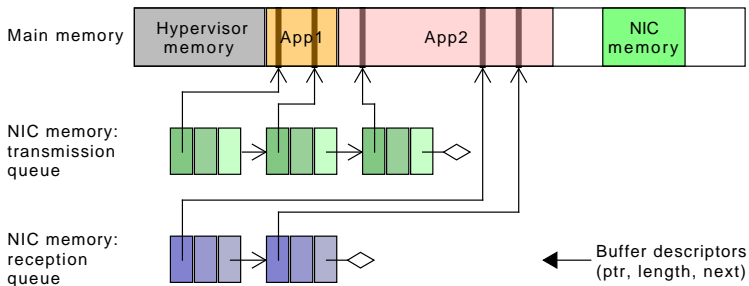


Figure: BeagleBone Black.



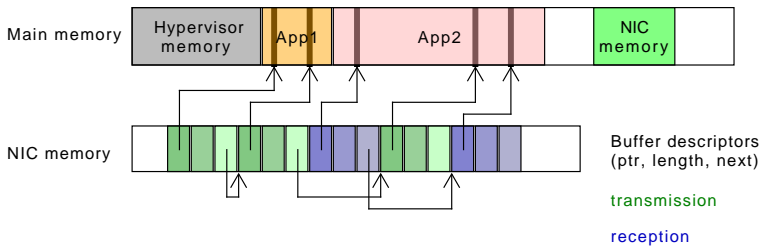




Table of Contents

- 1 Motivation
 - Security critical systems
 - Network Interface Controllers (NIC)
 - How to verify hardware devices?
- 2 Non proof-producing verification
 - Subsection 1
 - How trustful is it?
- 3 Proof-producing verification
 - Subsection 1
- 4 Conclusion

Section 2

Non proof-producing verification

Table of Contents

- 1 Motivation
 - Security critical systems
 - Network Interface Controllers (NIC)
 - How to verify hardware devices?
- 2 Non proof-producing verification
 - Subsection 1
 - How trustful is it?
- 3 Proof-producing verification
 - Subsection 1
- 4 Conclusion

Title

Table of Contents

- 1 Motivation
 - Security critical systems
 - Network Interface Controllers (NIC)
 - How to verify hardware devices?
- 2 Non proof-producing verification
 - Subsection 1
 - How trustful is it?
- 3 Proof-producing verification
 - Subsection 1
- 4 Conclusion

Title

Section 3

Proof-producing verification

Table of Contents

- 1 Motivation
 - Security critical systems
 - Network Interface Controllers (NIC)
 - How to verify hardware devices?
- 2 Non proof-producing verification
 - Subsection 1
 - How trustful is it?
- 3 Proof-producing verification
 - Subsection 1
- 4 Conclusion

Title

Section 4

Conclusion

Questions

References I