What the Stack? On Memory Exploitation and Protection in Resource Constrained Automotive Systems

Aljoscha Lautenbach Magnus Almgren Tomas Olovsson

Dept. of Computer Science and Engineering
Chalmers University of Technology
Gothenburg, Sweden

CRITIS 2017
The International Conference on Critical Information Infrastructures Security

Lucca, Italy

11 October, 2017

Why automotive security?

Connected vehicles

- Communication systems: Bluetooth, Wireless LAN, IEEE 802.11p, Radio Data System (RDS), telecommunication (3G/4G/soon 5G), ...
- Intelligent transport systems (ITS)
- Self-driving cars

Why automotive security?

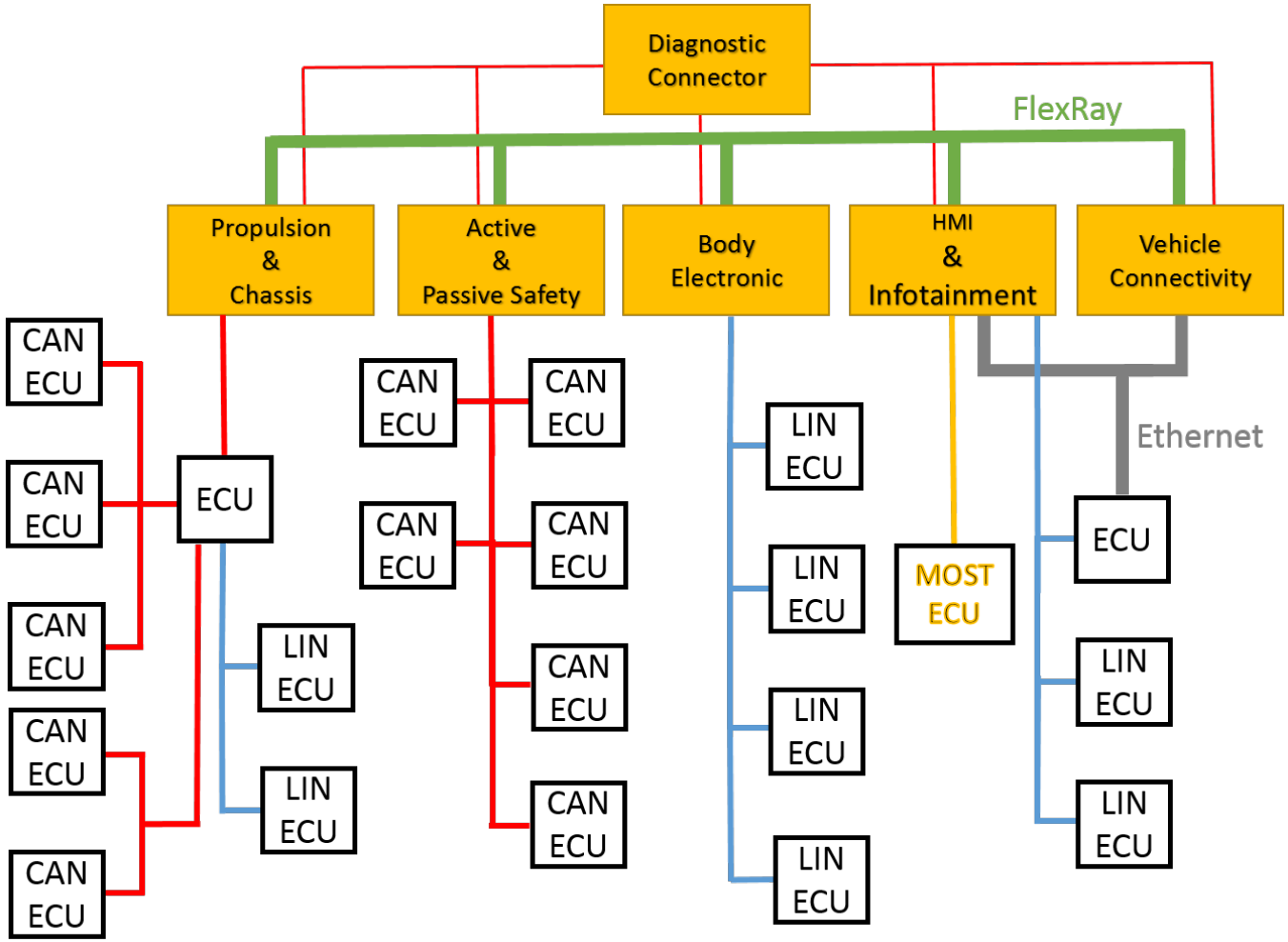
Connected vehicles

- Communication systems: Bluetooth, Wireless LAN, IEEE 802.11p, Radio Data System (RDS), telecommunication (3G/4G/soon 5G), ...
- Intelligent transport systems (ITS)
- Self-driving cars

Security perspective

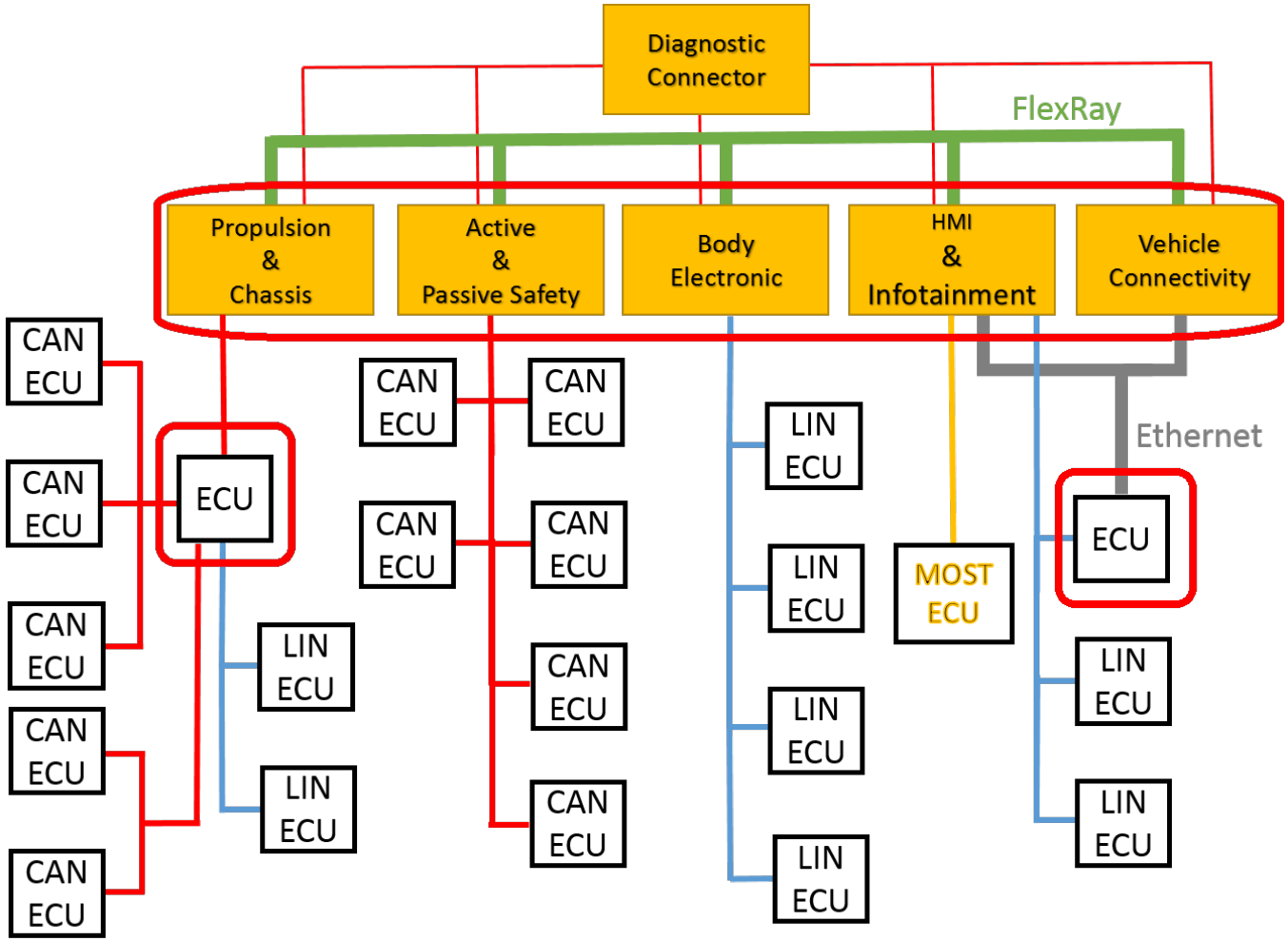
- Automotive security mostly focuses on communications

The in-vehicle network



Picture source: Nowdehi N., Lautenbach A., Olovsson T. "In-vehicle CAN message authentication: An evaluation based on industrial criteria". Vehicular Technology Conference Fall 2017.

The in-vehicle network



Picture source: Nowdehi N., Lautenbach A., Olovsson T. "In-vehicle CAN message authentication: An evaluation based on industrial criteria". Vehicular Technology Conference Fall 2017.

Electronic control unit (ECU) security

What is the problem?

Exploitation of memory corruption bugs!

Electronic control unit (ECU) security

What is the problem?

Exploitation of memory corruption bugs!

Checkoway et al. (2011)

"We also encourage the use of simple anti-exploitation mitigations such as stack cookies and ASLR that can be easily implemented even for simple processors [...]"

Typical ECU hardware

Typical ranges for resource constrained microcontrollers

Hardware	Specification	Most Common
RAM	4 KB - 500 KB	40 KB
Flash Memory	256 KB - 6 MB	1 MB
Processor Speed	16 - 150 MHz	80 MHz

ECU memory (1)

Types of memory

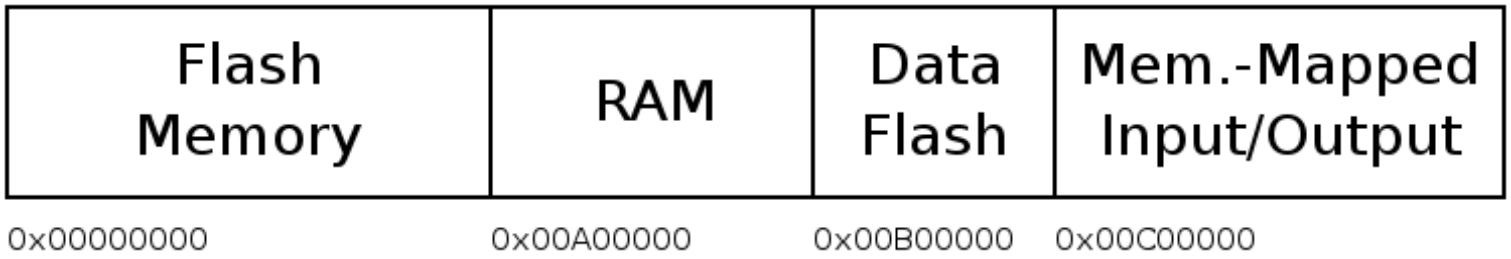
- Flash memory: boot loader, OS, tasks
- RAM
- Data flash
- Memory mapped Input/Output

ECU memory (2)

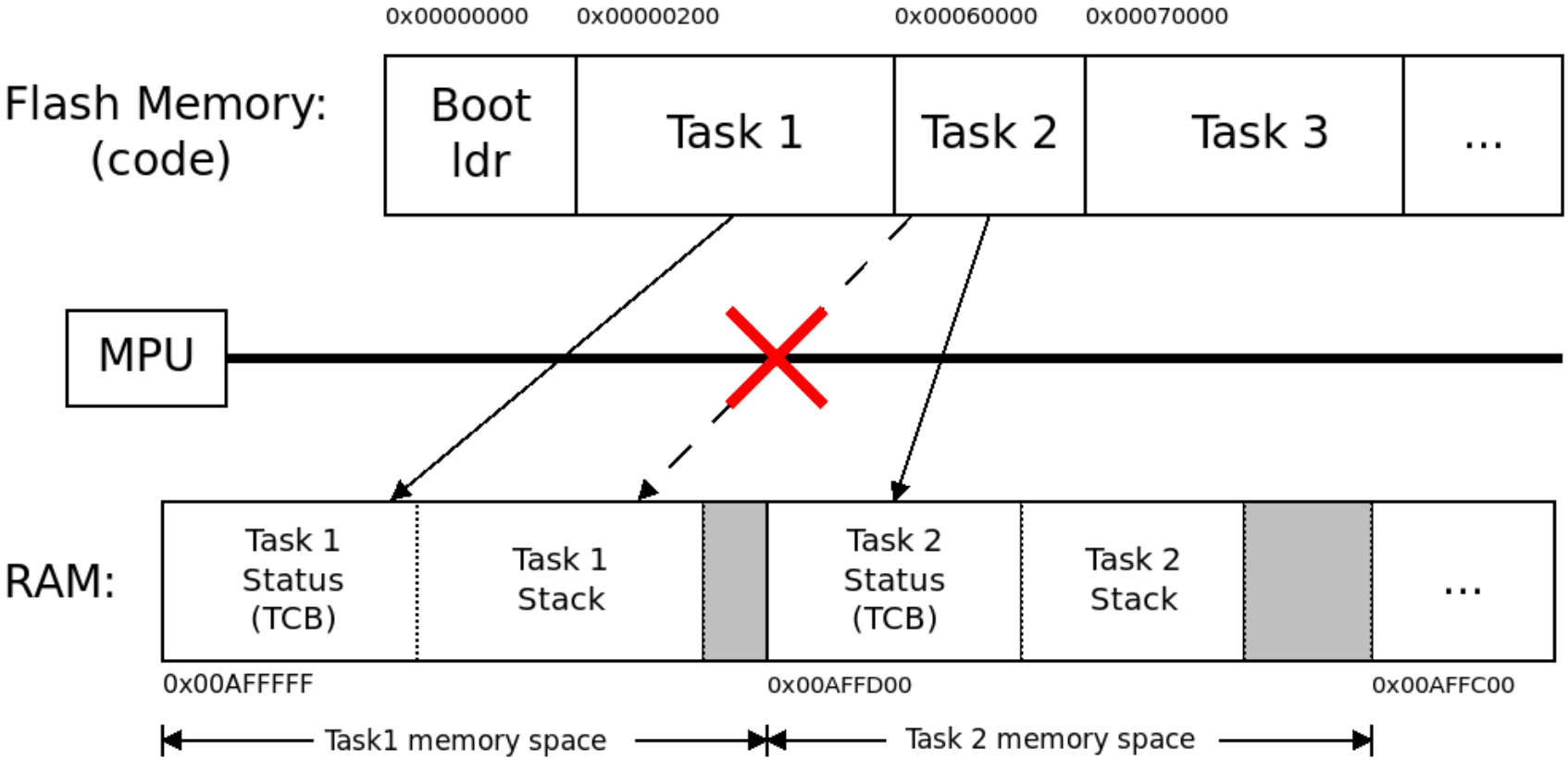
Memory properties

- No memory management unit (MMU), i.e., no virtual memory
- Statically assigned memory
- Memory protection unit (use optional)

Linear address space



Static task memory mapping into RAM

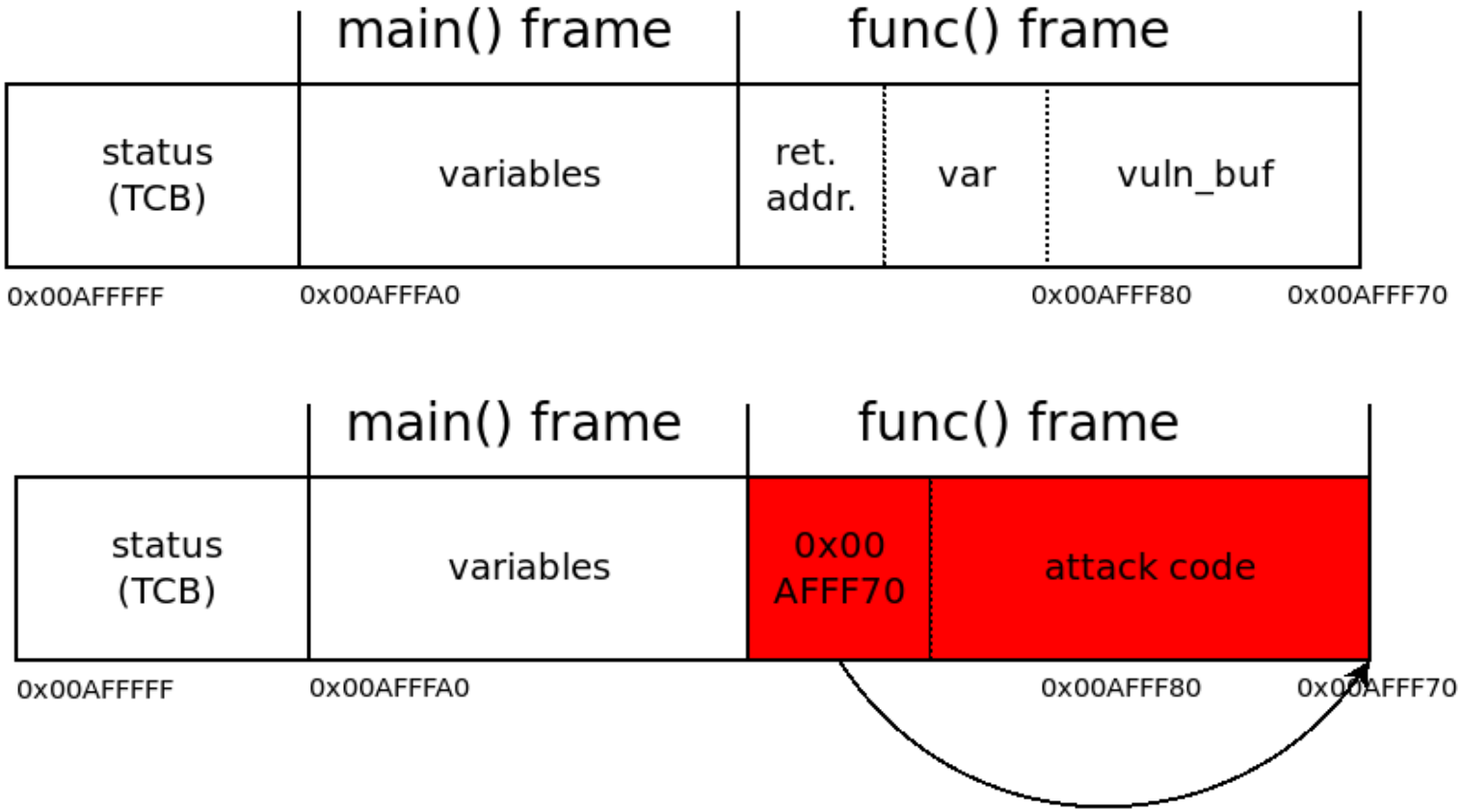


Memory exploitation and protection

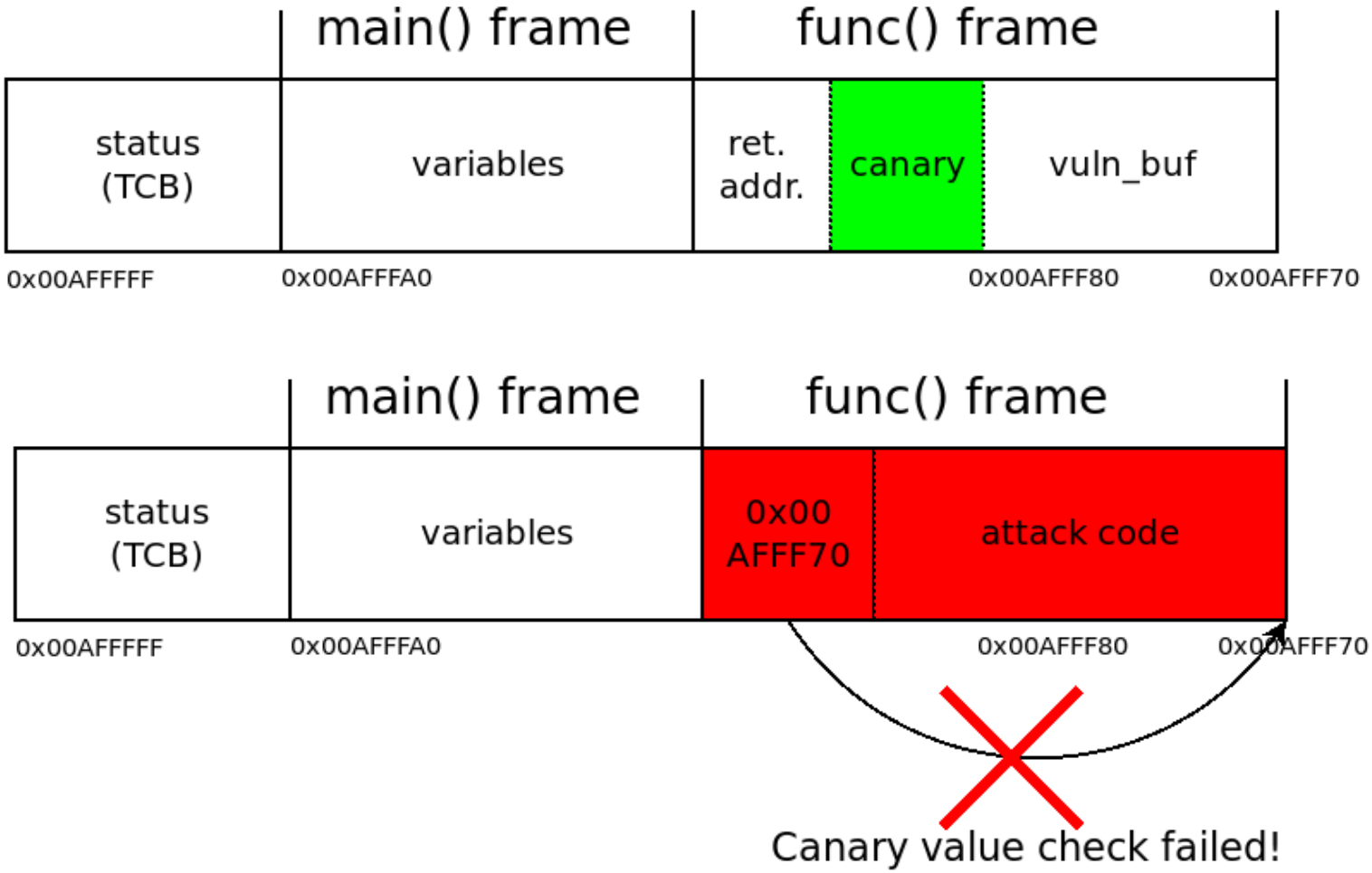
Attacks and defenses

1. Buffer overflows and stack canaries (aka stack cookies or stack-guards)
2. Buffer overflows and non-executable RAM
3. Return oriented programming (ROP) and compile-time randomization

Exploiting buffer overflows



Buffer overflows and stack canaries



Stack canaries - pros and cons

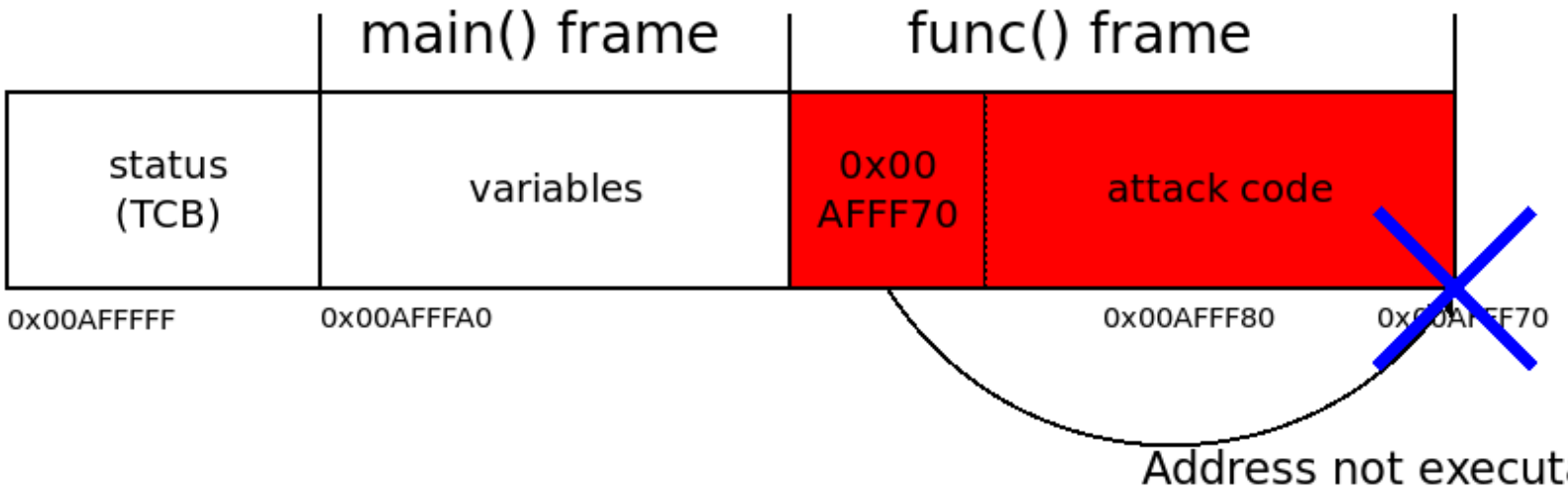
Pros

- Protects against simple buffer overflow attacks
- Simple to implement
- Proven technique (desktops and servers)

Cons

- Performance degradation

Buffer overflows and non-executable RAM



Non-executable RAM - pros and cons

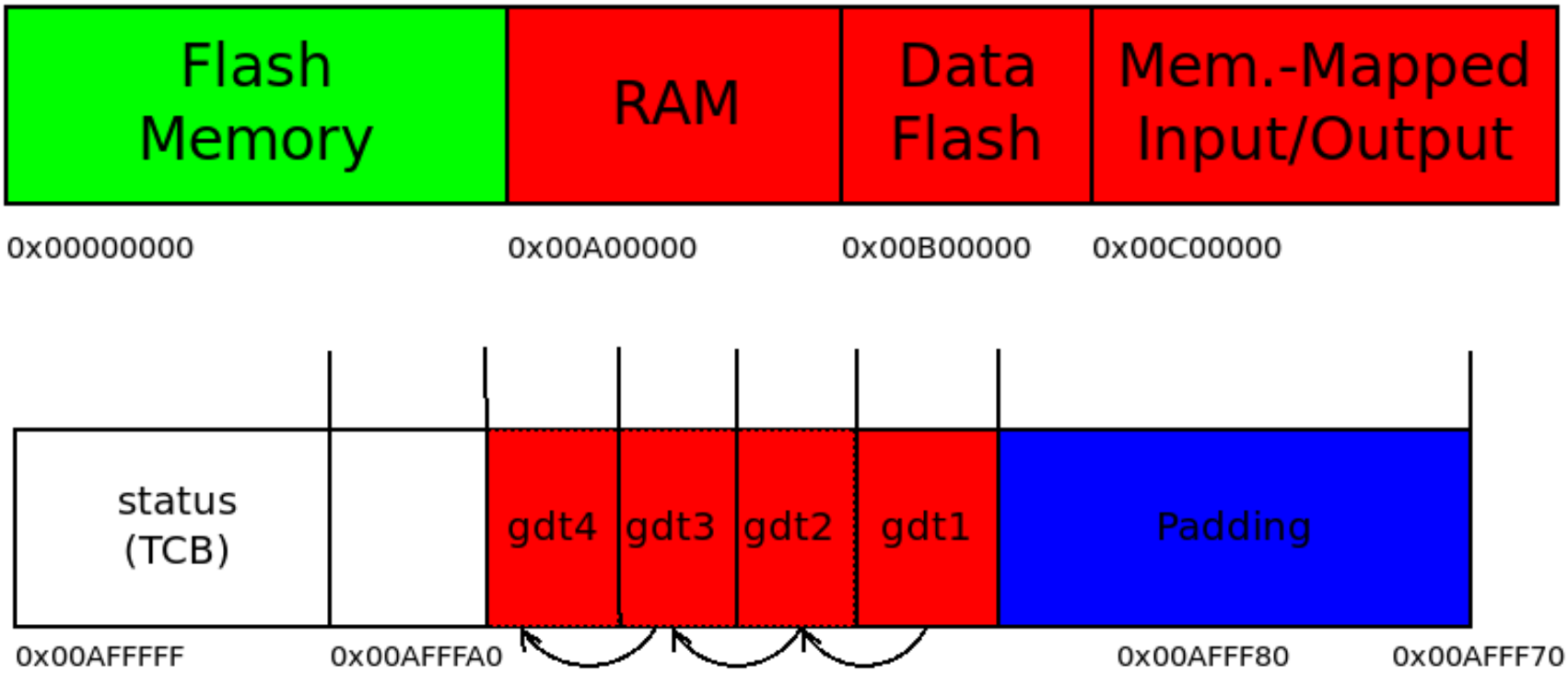
Pros

- Protects against all standard buffer overflow attacks
- Already supported by MPUs

Cons

- Sometimes RAM must be executable, e.g., during firmware upgrades.

Return oriented programming (ROP)

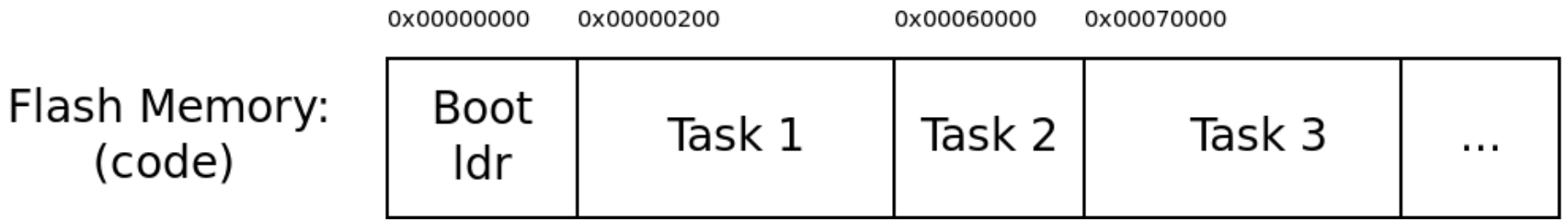


CRITIS 2017

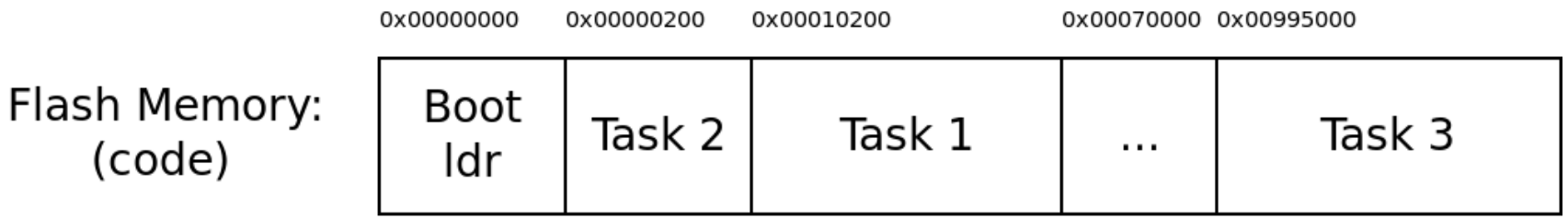
CHALMERS

Compile-time layout randomization

Standard layout



Randomized layout



CRITIS 2017

CHALMERS

Compile-time layout randomization - pros and cons

Pros

- Raises the bar for successful ROP attacks

Cons

- Low entropy
- Requires production changes (costly)
- Only of interest once non-executable RAM has been implemented

Conclusions

- Stack canaries should be used
- Non-executable RAM should be used
- Layout randomization is hard to implement in practice $\rightarrow$ further research is needed

Thank you!