Poster: μRAI: Securing Embedded Systems with Return Address Integrity

Naif Saleh Almakhdhub
Purdue University and
King Saud University

Abraham A. Clements
Sandia National Laboratories

Saurabh Bagchi
Purdue University

Mathias Payer
EPFL

Abstract

Embedded systems are deployed in security critical environments and have become a prominent target for remote attacks. Microcontroller-based systems (MCUS) are particularly vulnerable due to a combination of limited resources and low level programming which leads to bugs. Since MCUS are often a part of larger systems, vulnerabilities may jeopardize not just the security of the device itself but that of other systems as well. For example, exploiting a WiFi System on Chip (SoC) allows an attacker to hijack the smart phone's application processor.

Control-flow hijacking targeting the backward edge (e.g., Return-Oriented Programming–ROP) remains a threat for MCUS. Current defenses are either susceptible to ROP-style attacks or require special hardware such as a Trusted Execution Environment (TEE) that is not commonly available on MCUS.

We present μRAI [1], a compiler-based mitigation to *prevent* control-flow hijacking attacks targeting backward edges by enforcing the *Return Address Integrity (RAI)* property on MCUS. μRAI does not require any additional hardware such as TEE, making it applicable to the wide majority of MCUS. To achieve this, μRAI introduces a technique that moves return addresses from writable memory, to readable and executable memory. It re-purposes a single general purpose register that is never spilled, and uses it to resolve the correct return location. We evaluate against the different control-flow hijacking attacks scenarios targeting return addresses (e.g., arbitrary write), and demonstrate how μRAI prevents them all. Moreover, our evaluation shows that μRAI enforces its protection with negligible overhead.

1 Reference

This work will appear at NDSS 2020:

Naif Saleh Almakhdhub, Abraham A Clements, Saurabh Bagchi, and Mathias Payer. "μRAI: Securing Embedded Systems with Return Address Integrity". In Proceedings of the Network and Distributed System Security Symposium (NDSS), 2020.

2 DOI

Network and Distributed Systems Security (NDSS) Symposium 2020
23-26 February 2020, San Diego, CA, USA
ISBN 1-891562-61-4
https://dx.doi.org/10.14722/ndss.2020.24016
www.ndss-symposium.org

[1] https://github.com/embedded-sec/uRAI

µRAI : Securing Embedded Systems with Return Address Integrity[1]

Naif Saleh Almakhdhub — *Purdue and King Saud University*
Abraham A. Clements — *Sandia National Laboratories*
Saurabh Bagchi — *Purdue University*
Mathias Payer — *EPFL*

Problem

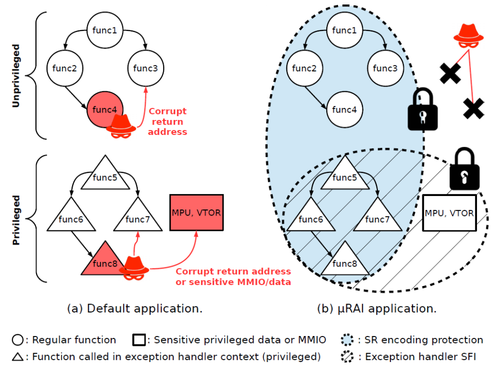
- **Embedded systems** and **IoT** are run on Microcontroller systems (**MCUS**)
- **MCUS** lack basic mitigations and are **prone to control-flow hijacking attacks** such as Return Oriented Programming (**ROP**)
- Proposed defenses have limited security guarantees, high runtime overhead, or require special hardware features

Objectives

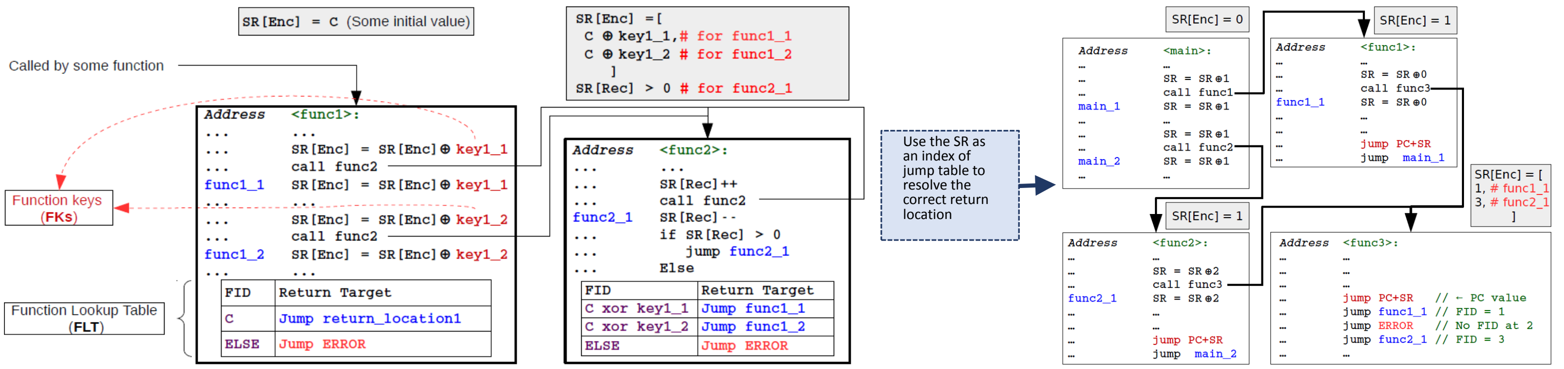
- Return Address Integrity (**RAI**) prevents ROP attacks on MCUS
- RAI results in **low runtime overhead**
- RAI **does not require** special hardware

µRAI

- Identifies the possible return targets of each function in the call graph
- Transforms the set of return targets to a jump table in R+X memory
- Introduces a State Register (**SR**), which is **never spilled** and is exclusively used by µRAI
- Uses the SR at run time to resolve the correct return location from the jump table
- **Enforces the RAI property** since the SR and jump table are **inaccessible** to an adversary
- Protects sensitive Memory Mapped IO (MMIO) by enforcing Software-based Fault Isolation (SFI) on functions callable within an exception handler context to protect sensitive such as the MPU
- Partitions the SR into segments to curb path explosion
- Applies a type-based CFI for forward edges

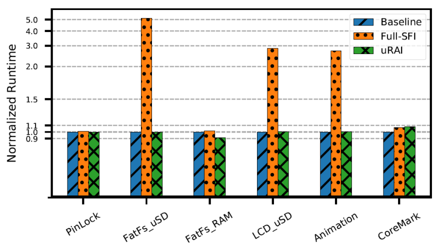


Compiler Transformation



Evaluation

Runtime



µRAI enforces the RAI property with low overhead in contrast to mechanisms requiring full-SFI

Comparison to backward edge Type-based CFI

App	Type-based CFI Target Set Max.	Type-based CFI Target Set Ave.
PinLock	8	3
FatFs_uSD	94	21
FatFs_RAM	94	27
LCD_uSD	49	11
Animation	49	11
CoreMark	52	12

µRAI eliminates the remaining attack surface for control-flow bending attacks

Security

Attack	Prevented
Buffer overflow	✓
Arbitrary write	✓
Stack pivot	✓

µRAI prevents all control-flow hijacking attack scenarios targeting return addresses

References

[1] Naif Saleh Almakhdhub, Abraham A Clements, Saurabh Bagchi, and Mathias Payer. In *The Annual Network and Distributed System Security Symposium* (**NDSS**), ***2020***

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525 SAND2020-0912 C.