EMBOSOM

Emigrating Embedded Software Security into Modern Emerging Hardware Paradigms

Spandan Das
Universität Bremen

RUHR UNIVERSITÄT BOCHUM RUB

Universität Bremen

Motivation

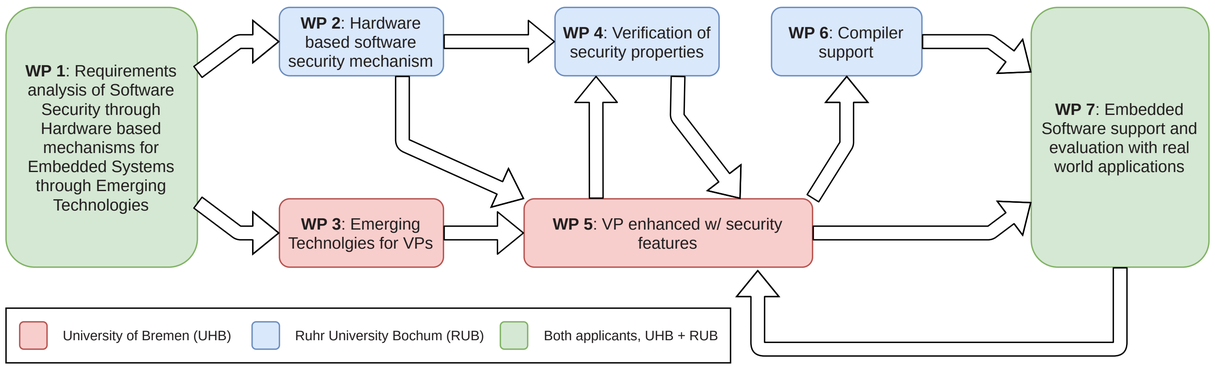
- Various protections are required to protect software against exploitation
- Prior work targets conventional systems and software-only solutions
- Embedded domain (e.g., IoT, Edge) becomes more and more important

$\Longrightarrow$ **Challenge:** Design solutions for constrained embedded devices

Goals

- Design hardware-based security mechanisms for embedded devices
- Utilize emerging technologies
- Accelerate development via virtual prototyping
- Verify and validate security mechanisms

Structure of Work Programme

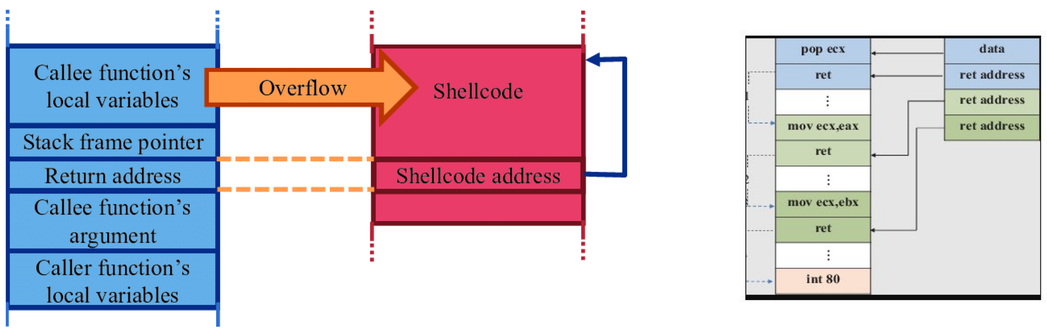


Work packages and interdependence of planned collaboration

Hardware-based Pointer Protection for RISC-V

Motivation

- Memory unsafe languages can be used to bypass software security
- Adversaries can hijack pointers and overwrite mission-critical data
- Attacks like buffer overflow and return-oriented programming can lead to memory vulnerabilities

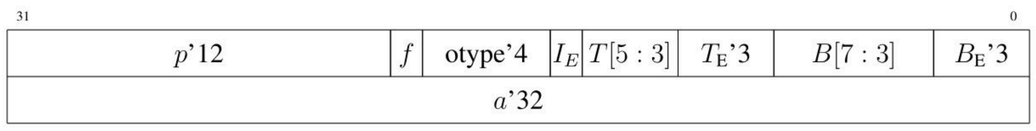


a b

Example of memory attacks
a. Buffer overflow b. Return-oriented programming

Possible Solution

- Employ hardware to validate pointer access $\rightarrow$ CHERI
- CHERI differentiates between pointer and data using tags
- CHERI employs both bounds checks as well as hardware permissions in enhanced pointers (known as capabilities)
- CHERI offers protection against common memory vulnerabilities, like buffer overflow, use-after-free and return/jump-oriented programming attacks
- CHERI machines can run legacy codes too (backward compatibility)

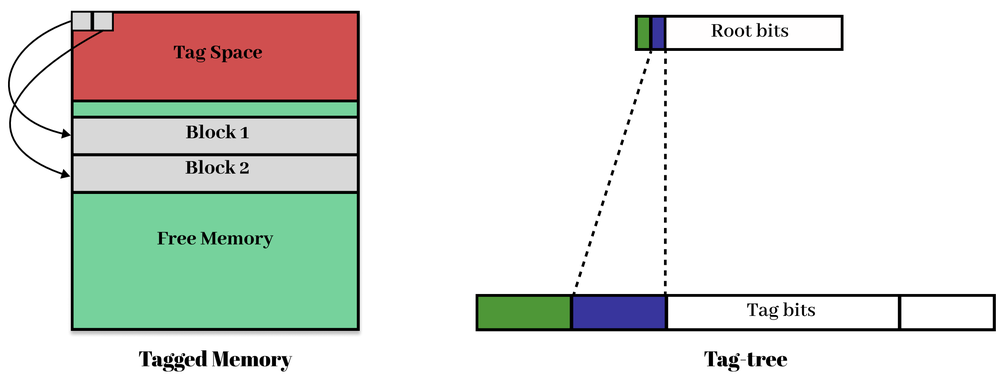


f: flag p: permissions otype: object type a: pointer address

Latest compressed capability format of CHERI

Implementing Tagged Memory in the RISC-V VP

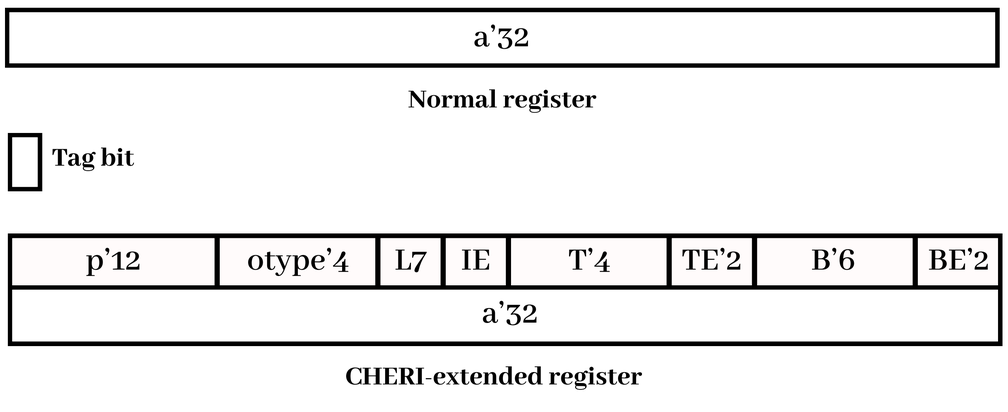
- The memory is partitioned into a data space and a tag space
- The tag space is not available for data storage
- Each capability-sized chunk of data space is mapped to a single bit in the tag space, which decides whether the chunk holds capability or data
- Tag space is organized in a tree-like data structure for easier access of tags



Tagged Memory Tag-tree

Implementing CHERI-CPU in the RISC-V VP

- Registers are extended to hold full capabilities (tag is stored separately)
- Special Capability Registers added (e.g. DDC) or extended from existing CSRs (e.g. mtvec $\rightarrow$ MTCC)
- Capability structure changed: unnecessary architecture bit used for better bounds precision



Extending the Instruction Set

The RISC-V instruction set is extended with:

- Capability-inspection instructions
- Capability-modification instructions
- Capability-arithmetic instructions
- Capability-comparison instructions
- Capability-based control flow instructions
- Capability-based load/store instructions (for both data and capability)

Future Plans and Publications

Future Plans

- Adding compressed CHERI instructions
- Verification of security notions like non-interference and monotonicity in the CHERI VP
- Designing a faster tag space using in-memory computing

Publications

- Das, S., Lüth, C. and Drechsler, R., 2025. Designing Memory Protection for a RISC- V Nano-VP. In 3rd Workshop on Nano Security: From Nano-Electronics to Secure Systems (NanoSec'25)

DFG Deutsche Forschungsgemeinschaft

Contact:
Christoph Lüth[1], **Tim Güneysu**[2], **Pascal Sasdrich**[2], **Rolf Drechsler**[1]
[1] Universität Bremen, [2] Ruhr-Universität Bochum
{clueth,drechsler}@uni-bremen.de
{tim.guneysu,pascal.sasdrich}@rub.de