CHERITech'26
CONFERENCE

CHERI

12–13 Nov 2026
Munich, DE

Intra-allocation bounds in CheriBSD

Next steps in spatial memory safety

Alfredo Mazzinghi
Senior Research Engineer
University of Cambridge

Located at semi | SEMICON EUROPA

Agenda

- The goal of privilege minimization
- Bounds precision
- State of the art: compiler-assisted sub-object bounds
- Extending to general intra-allocation bounds
- Patterns in intra-allocation bounds
- Conclusions & Next steps

Privilege Minimization

- Core principle of CHERI hardware/software design
  - Minimize access to virtual address space (Spatial safety)
  - Minimize access across time (Temporal safety)
  - Isolation among software components (Compartmentalization)
- Monotonic delegation of capabilities across firmware, kernel and user programs
- Enforced by hardware, organized by software

Spatial Safety in CheriBSD

- Fork of FreeBSD with mature CHERI support
  - CHERI spatial safety in kernel and user space
- What does this mean?
  - Bounds on global and implicit language pointers
  - Bounds on stack allocations
  - Memory mapping interfaces return bounded capabilities
  - Allocators must also produce bounded capabilities
- SW determines when/how to narrow bounds
  - Extend the API contract of allocator interfaces

Bounds Precision

- Consequence of capability compression encoding
  - "Large" capability lengths require some alignment of the base
  - Bounds can be narrowed as a best-effort operation, rounding base and length to the closest representable boundary
  - Allows trade-off between software disruption and security properties.
- In general, not a problem
  - Alignment constraints managed by system allocators
  - Compiler handles stack and data layout accordingly
  - Precise bounds are part of the kernel allocators API contract

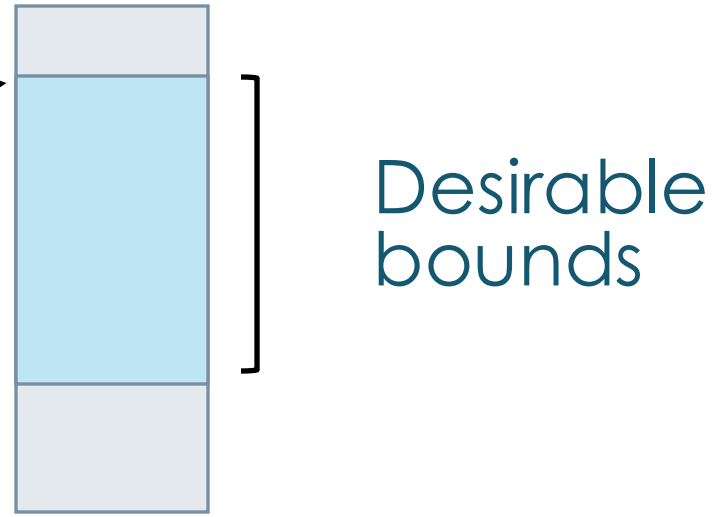
Sub-object bounds in CheriBSD

- Bounds are enforced at allocation granularity
  - But

```
struct foo {
    int header;
    char bar[128];
    void *ptr;
};

// ...

do_bar(f->bar);
```



Desirable bounds

Sub-object bounds in CheriBSD

- Compiler automatically narrows bounds
  - Triggered by address-of operator or array-decay
- Good
  - Reduces access privilege to a region within an allocation
  - Automated, just a compile-time knob
- Current limitations
  - Currently best-effort in terms of bounds precision
  - Some software compatibility friction

Sub-object bounds in CheriBSD

- Sub-objects are not guaranteed to be representable
  - Measured the extent of the problem in the CheriBSD kernel
  - < 0.3% of the data structures exhibits imprecision
  - ~4500 structures with VLA

Target	# of imprecise members	# of imprecise array members	# of imprecise members with reachable pointers
RV64	224	143	84
Morello	78	57	28

- Prototype kernel with precise sub-object bounds
  - Demonstrates feasibility of fail-close semantics

Intra-allocation Bounds

- Generalization of sub-object bounds
  - Apply whenever a region of memory is subdivided in a set of logical partitions, possibly holding different data types.
  - Compiler-assisted bounds do not cover all cases
- Three broad categories in the CheriBSD kernel
  - Aggregate type sub-objects
  - Sub-allocation
  - Domain-specific memory partitioning

Sub-allocation Pattern

- A single allocation is split up to satisfy multiple, related, allocations.
  - Ex. malloc(sizeof(header) + cnt * sizeof(entry))
  - Variable Length Arrays (VLA) are a special case of this pattern
- Notable cases in CheriBSD
  - Allocation of struct uio for scatter-gather I/O, such as readv()
  - ELF Auxargs region layout
  - Kernel stack layout with embedded struct pcb at the bottom

Domain-specific Patterns

- Covers cases that require additional domain knowledge to effectively narrow bounds.
  - Sub-allocation like patterns, but
  - The layout and placement of the sub-regions is constrained and can not be modified.
- Notable cases in CheriBSD
  - Unmapped mbuf pages accessed via the direct map
  - Memory-mapped device regions, layout dictated by device spec

Conclusions

- CHERI spatial safety is very effective
  - We can go further
- Effective deployment of precise sub-object bounds is possible
- Understanding intra-allocation bounds principles is important to minimize access rights
  - Not as automated as sub-object bounds
  - CHERI software development practices must take this into account

CHERITech'26 CONFERENCE

12–13 Nov 2026
Munich, DE

Located at semi | SEMICON EUROPA

CHERI

THANK YOU

Contact

alfredo.mazzinghi@cl.cam.ac.uk
contact@cheri-alliance.org

Web

www.cheri-alliance.org

CHERI™ and the CHERI logo are protected trademarks. Use of these marks is subject to the CHERI Alliance Trademark Policy – https://cheri-alliance.org/legal-and-policies/trademark/

This work © 2026 by CHERI Alliance is licensed under CC BY-SA 4.0 (Creative Commons Attribution-ShareAlike 4.0 International) – https://creativecommons.org/licenses/by-sa/4.0/