**CHERI Research**

* CHERI (Capability Hardware Enhanced RISC Instructions)
* Enhances processor architectures with fine-grained memory protection to mitigate security vulnerabilities

Capabilities

* In CHERI, memory pointers are replaced with ***capabilities***. A capability is a bounded, unforgeable reference to memory that includes metadata such as:
  + The base address of the memory region.
  + The length of the memory region.
  + Permissions (e.g., read, write, execute).
* This contrasts with traditional memory where pointers are just addresses, with no inherent information about the bounds or permissions of the memory they reference.

Spatial Memory Safety

* Capabilities enforce ***spatial memory safety*** by ensuring a program cannot access memory outside the bounds of the capability.
* In traditional memory, pointers can be used to access arbitrary memory locations, which can lead to buffer overflows and other vulnerabilities.

Temporal Memory Safety

* Capabilities can be *revoked* when memory is deallocated, reducing risks like use-after-free vulnerabilities.
* In traditional memory, there is no built-in mechanism to track or revoke access to freed memory.

Fine-Grained Access Control:

* Each capability has an associated set of permissions that define what operations are allowed (e.g., reading, writing, or executing memory).
* Traditional memory does not enforce such granular control; permissions are typically applied at the page level (e.g., using page tables).

Unforgeability**:**

* Capabilities are unforgeable, meaning they cannot be created or modified by software. They are tightly controlled by hardware mechanisms.
* In contrast, traditional pointers can be arbitrarily modified, leading to potential exploits like pointer corruption.

Hardware Support:

* CHERI relies on hardware extensions to implement capabilities, including tagged memory to associate metadata with memory locations.
* Normal memory lacks this hardware support, relying on software and the MMU (Memory Management Unit) for protection mechanisms like paging.

**Advantages of CHERI Memory:**

* **Increased security**: By enforcing strict bounds and permissions, CHERI mitigates many common vulnerabilities like buffer overflows, use-after-free, and code injection.
* **Compatibility**: CHERI can extend existing architectures (e.g., ARM, RISC-V) without requiring complete rewrites of existing software.

**Challenges:**

* **Performance Overheads**: Maintaining and checking capabilities can introduce performance penalties.
* **Complexity**: Adopting CHERI may require changes to software and programming models.