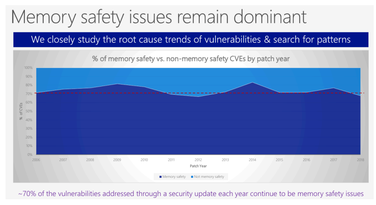
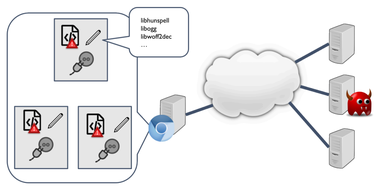
Capstone: An Architecture Design for Expressive Security

Jason Zhijingcheng Yu†, Prateek Saxena
School of Computing, National University of Singapore
† final-year PhD student open for hiring

Motivation: Patchwork of Security Extensions

Security Challenges





	Product Name	Vendor Name	Product Type	Number of Vulnerabilities
1	Debian Linux	Debian	OS	8161
2	Android	Google	OS	5136
3	Fedora	Fedoraproject	OS	4497
4	Ubuntu Linux	Canonical	OS	3961
5	Linux Kernel	Linux	OS	3186
6	Mac Os X	Apple	OS	3103
7	Windows 10	Microsoft	OS	3054
8	Windows Server 2016	Microsoft	OS	3042
9	Iphone Os	Apple	OS	2941
10	Chrome	Google	Application	2677

Memory Safety | Fine-grained Isolation | Confidential Computing

Patchwork of Security Extensions

Spatial Memory Safety	[Intel MPK, x86/64 DEP/NX] [Intel MPX, RISC-V/ARM CHERI]
Temporal Memory Safety	[ARM MTE]
Concurrent Thread Safety	[Intel TSX] [ARM TME]
Intra-process Sandboxing	[Intel SGX] [Intel MPK]
Process Sandboxing	[x86/64 Privilege Rings]
Virtualization	[AMD SEV] [Intel VT-x] [Intel TDX] [ARM CCA]
Red-Green Secure Worlds	[ARM TZ] [Intel TXT]
Nested / App Virtualization	[Intel VT-x] [Intel SGX]

Problem: Compose Security Extensions?

Example: SGX

+ Exception handling (Cui et al., 2021)

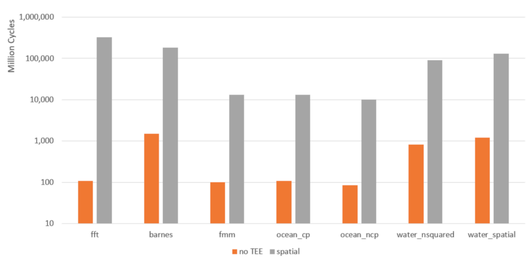
SmashEx

Arbitrary code execution

Affecting 9 SGX runtimes

CVE-2021-0186, CVE-2021-33767

+ Memory sharing (Yu et al., 2022)



2–3 orders of magnitude overhead

Goal

Can one design a unified foundation for multiple security goals? (Yu et al., 2023)

Minimal set of properties

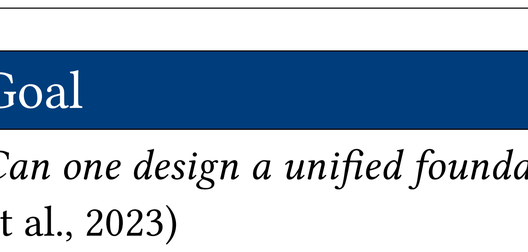
P1: **Exclusive Access**
P2: **Revocable Delegation**
P3: **Extensible Hierarchy**
P4: **Secure Domain Switching**

Capstone
Capstone (USENIX '23)

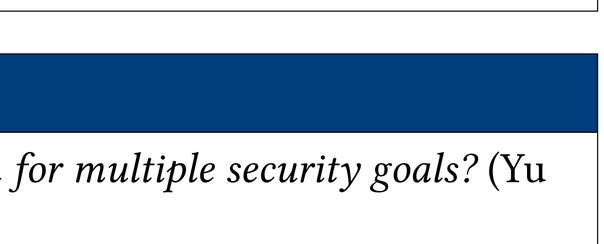
- Spatial Memory Safety
- Temporal Memory Safety
- Concurrent Thread Safety
- Intra-process Sandboxing
- Process Sandboxing
- Virtualization
- Red-Green Secure Worlds
- Nested / App Virtualization

Desired Properties

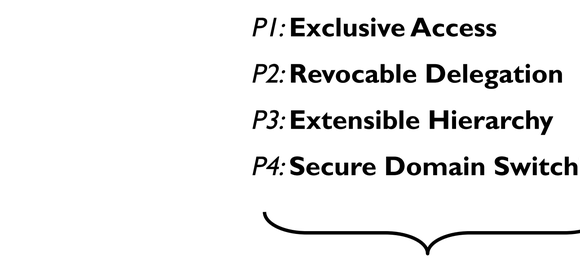
P1: Exclusive Access



P2: Revocable Delegation



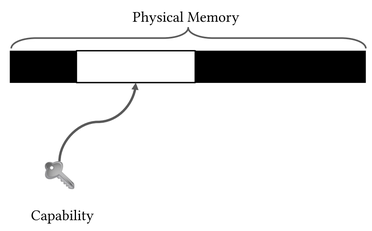
P3: Extensible Hierarchy



P4: Secure Domain Switching

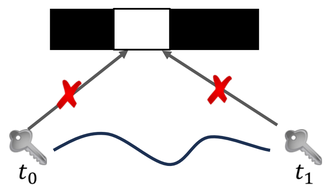
Base Design: Capability-based Security

Capability



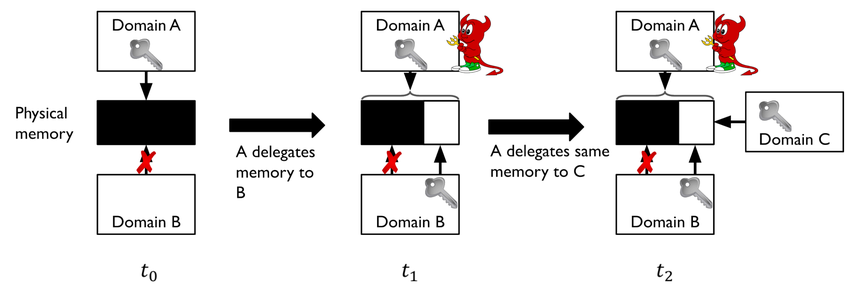
Unforgeability

Monotonicity



Base Capability-based Model Is Insufficient

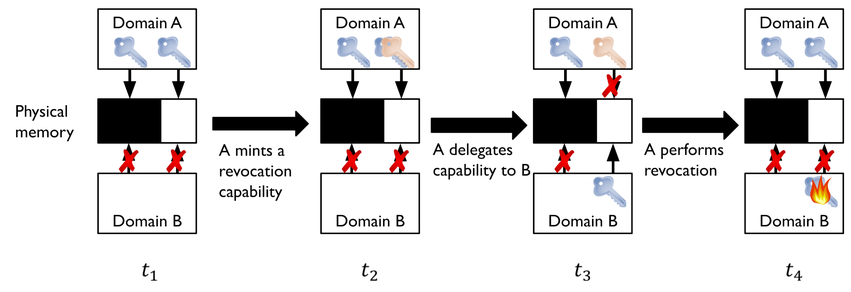
Example: P1 (Exclusive Access) cannot be achieved



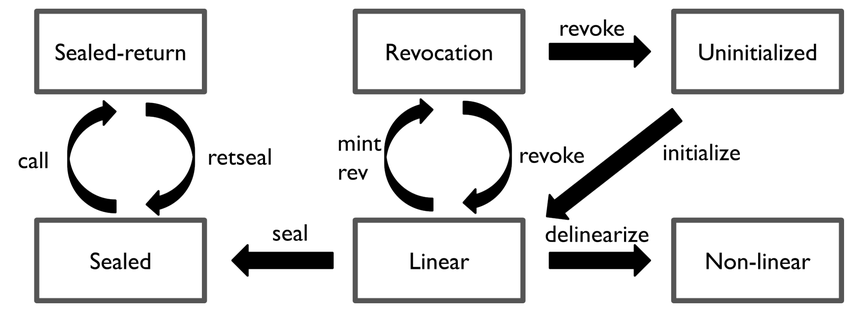
Capability-based Model in Capstone

- **Linear capability:** Non-duplicable
- **Revocation capability:** A capability "snapshot", usable only for revocation

Example:

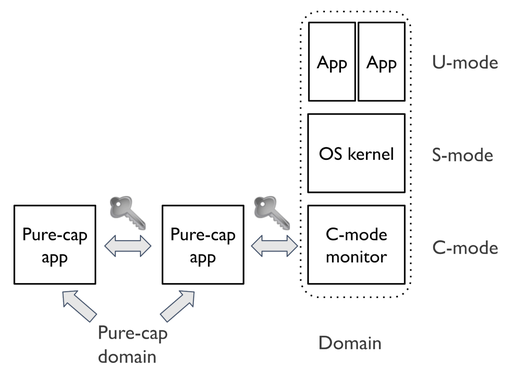


Other capability types:

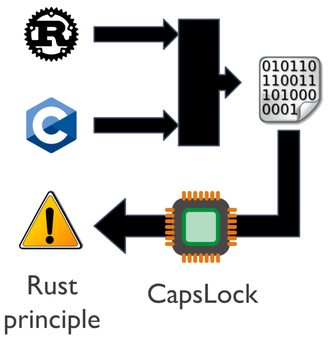


Example Use Cases

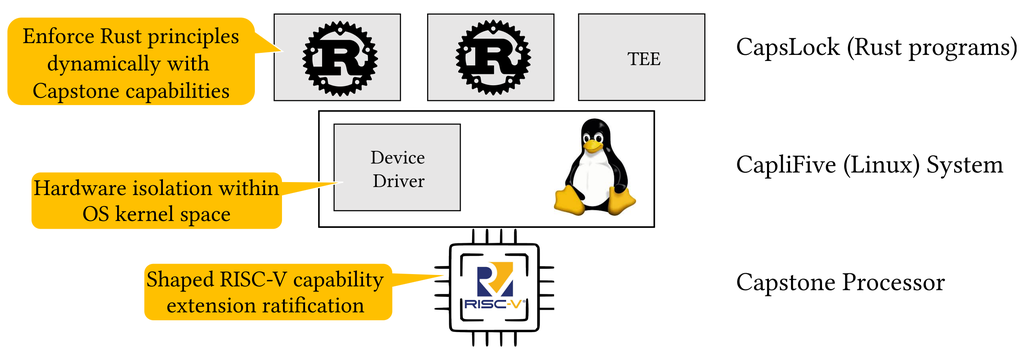
Nestable Two-way Isolation



Safety of Mixed Rust Code



Capstone Stack



NUS National University of Singapore

For more information, scan the QR codes →

Capstone | Documentation | Author