CHERITech22
6th of September 2022

Data Compartmentalization using CHERI Hybrid

Andrei Lascu
CapableVMs
King’s College London

Motivation

```
void* new_ptr = malloc(24);
…
free(new_ptr);
void* other_ptr = malloc(36);
```

Motivation

- Easier to reason about capabilities
- Delegate capability-aware changes to a wrapper

Overview

Program Counter Capability (PCC) register

```
b c0
blr c0
```

Default Data Capability (DDC) register

```
ldr x0
str x0
```

Overview

Program Counter Capability (PCC) register

Default Data Capability (DDC) register

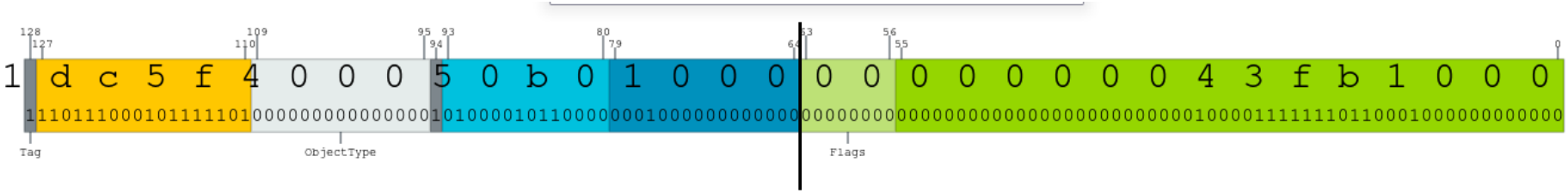
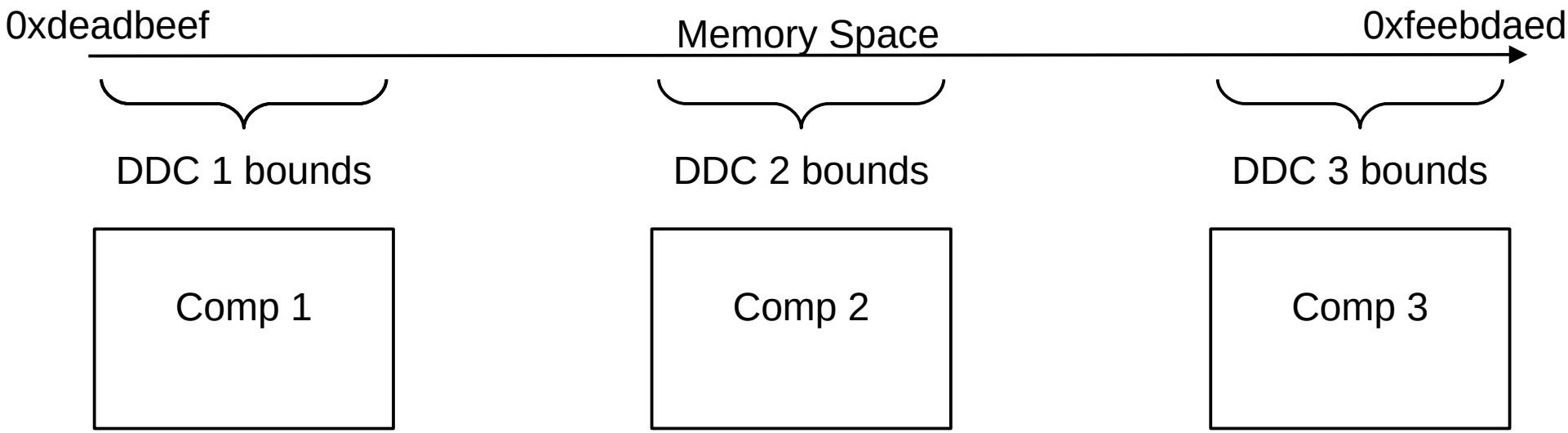
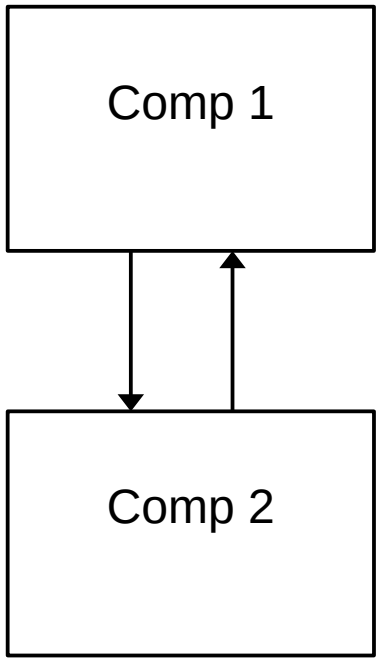


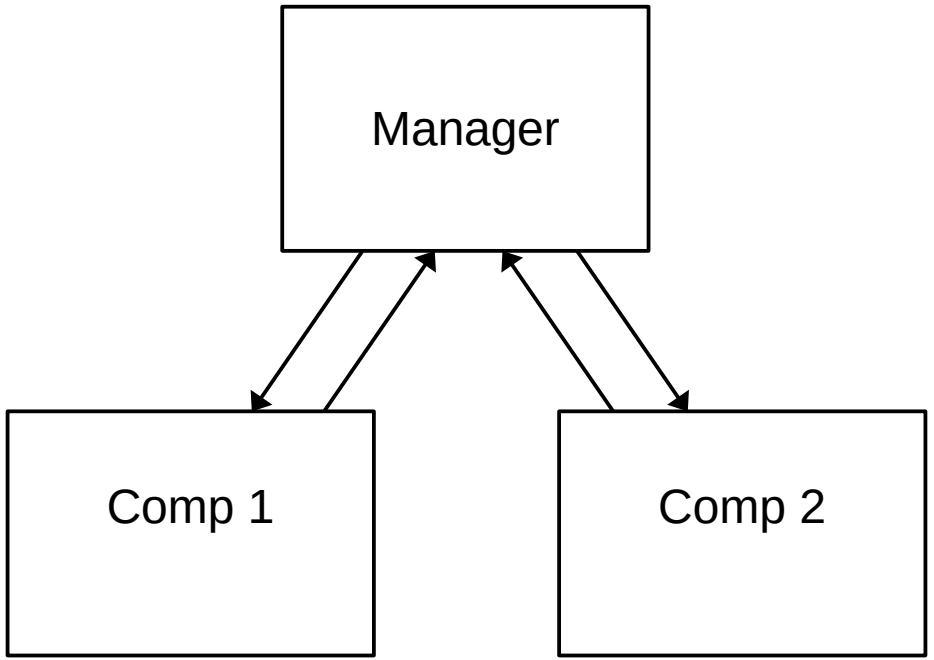
Image from:
https://www.morello-project.org/capinfo

Overview – DDC compartmentalization



Overview – DDC compartmentalization





Overview – transition

```
ldr c0, #DDC_ADDRESS
msr DDC, c0
```

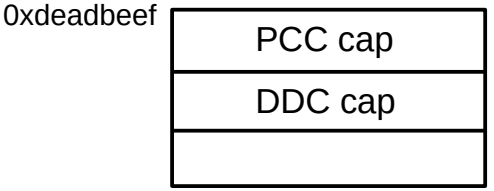
Overview – transition

```
ldr c0, #DDC_ADDRESS
ldr c1, #SEAL_CAP
seal c0, c0, c1
str c0, #DDC_ADDRESS
```

```
  ldr c0, #DDC_ADDRESS
{ ldr c1, #SEAL_CAP
{ unseal c0, c0, c1
  msr DDC, c0
```

Overview – transition

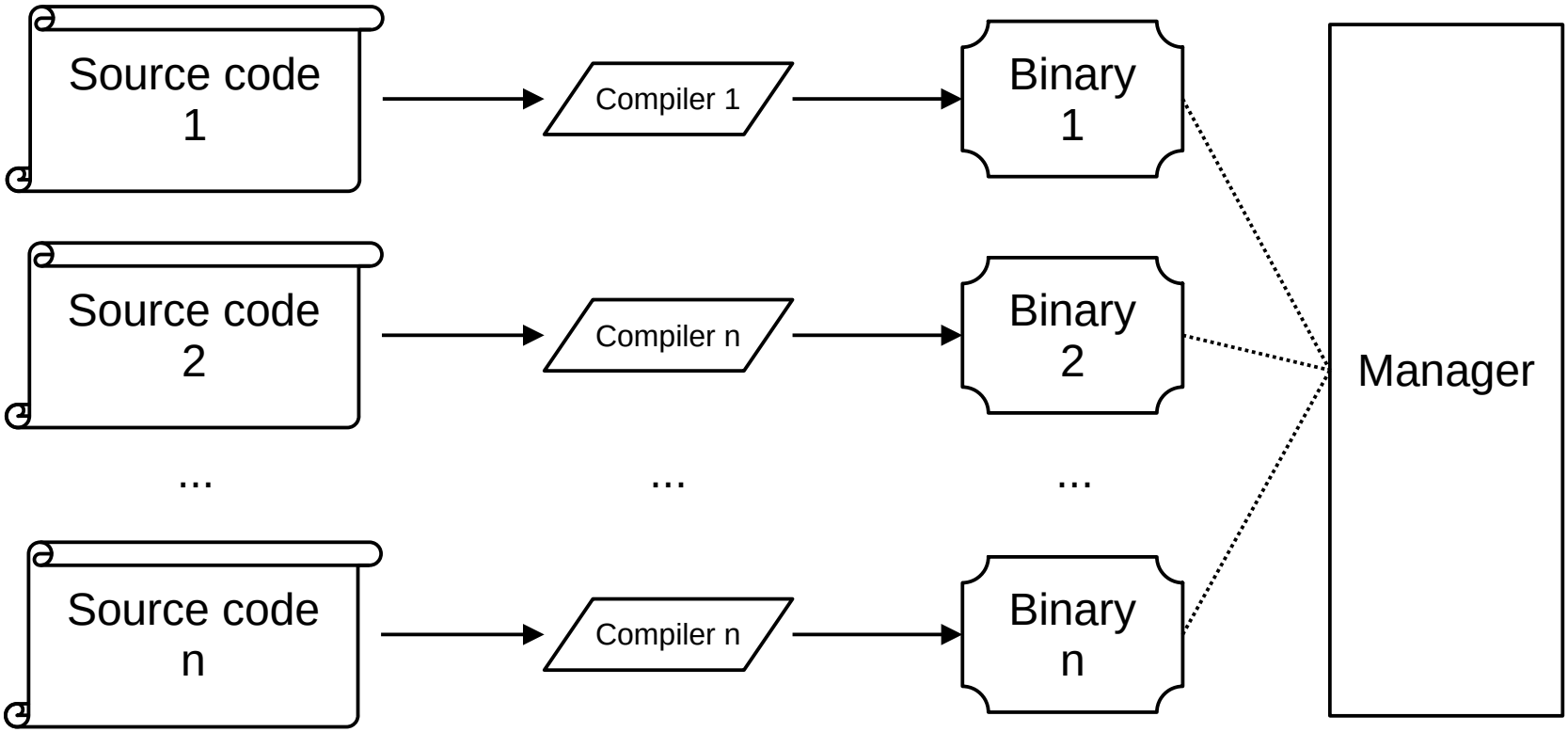
```
ldpb(l)r Ct, [<Cn|CSP>]
```




```
mov x0, #0xdeadbeef
cvtd c0, x0
seal c0, 0b10
```

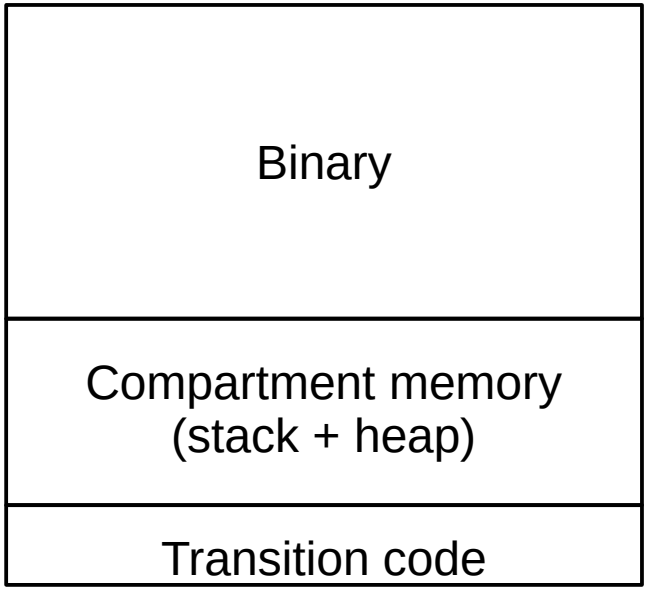
seal for use with `lpb`

Overview – binary compartments



Implementation

Compartment



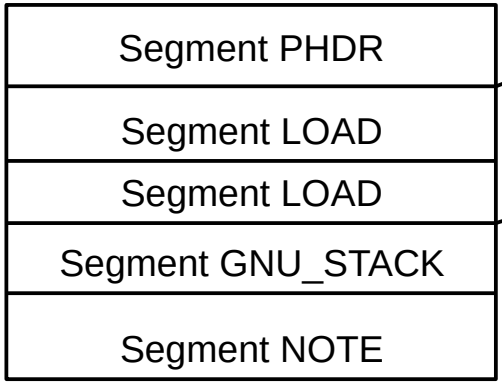
Manager

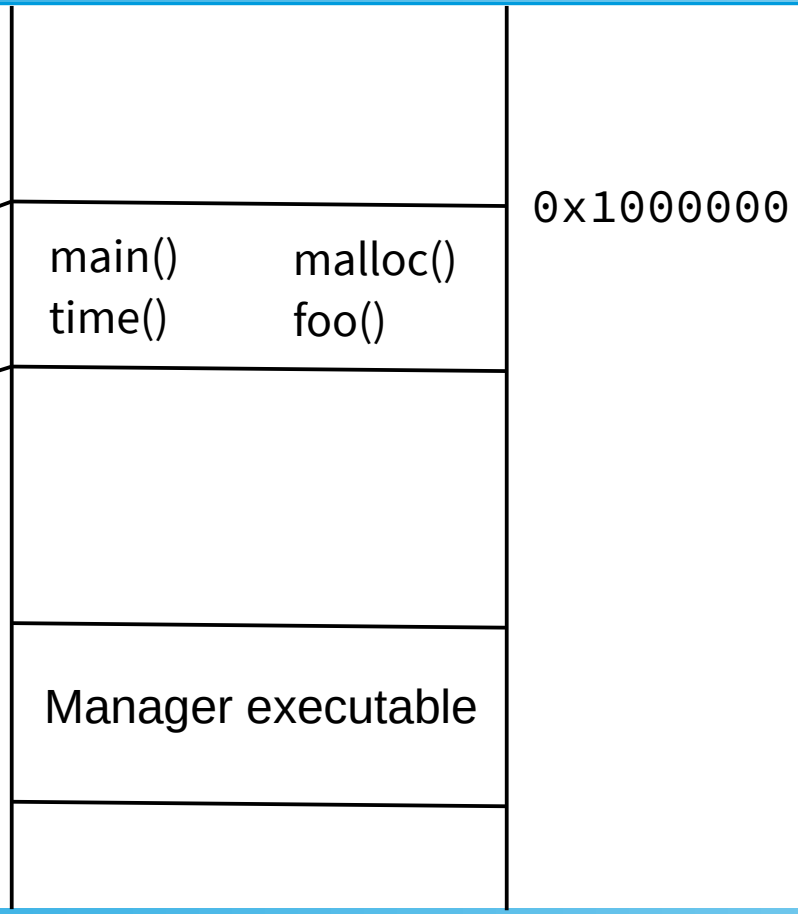
```
add_comp()
delete_comp()
exec_comp(func)
```

```
my_malloc(size_t)
my_realloc(void*, size_t)
my_fprintf(FILE*, const char*, ...)
```

Implementation - add_comp()

Statically compiled ELF binary
(`--image-base=0x1000000`)





Implementation – function interception

- external calls
- functions with syscalls
- any other functions (memory allocation)

Implementation – function interception

```
00000000010d4e60 <realloc>:
 10d4e60: ff c3 03 d1   sub     sp, sp, #240          // =240
 10d4e64: fd 7b 09 a9   stp     x29, x30, [sp, #144]
 10d4e68: fb 53 00 f9   str     x27, [sp, #160]
 10d4e6c: fa 67 0b a9   stp     x26, x25, [sp, #176]
```

```
compartment_transition_out:
   stp c29, clr, [sp, #-32]!
   ldpblr c29, [c11]
   ldp c29, clr, [sp], #32
   ret
```

```
// Get manager target address
movz x10, #target_addr:lo16
movk x10, #target_addr:hi16
// Get `ldpblr` capability addr
adrp x11, #OFFSET
ldr c11, [x11, #OFFSET]
b compartment_transition_out
```

Immediate work

- Passing arguments
- Inter-compartment communication (via manager)
- Design exposing compartment entry points in the compartment itself

Future Challenges

- Shared library calls
- Audit transition security
- Control forwarded function calls permissions (use existing capability permissions, something like OpenBSD’s `pledge`)

Closing Remarks

- Hybrid CHERI – easier to reason about security
- Proof of concept – lua VM data compartmentalization
- More performant data sharing

https://github.com/capablevms/CHERI-ELF-comp