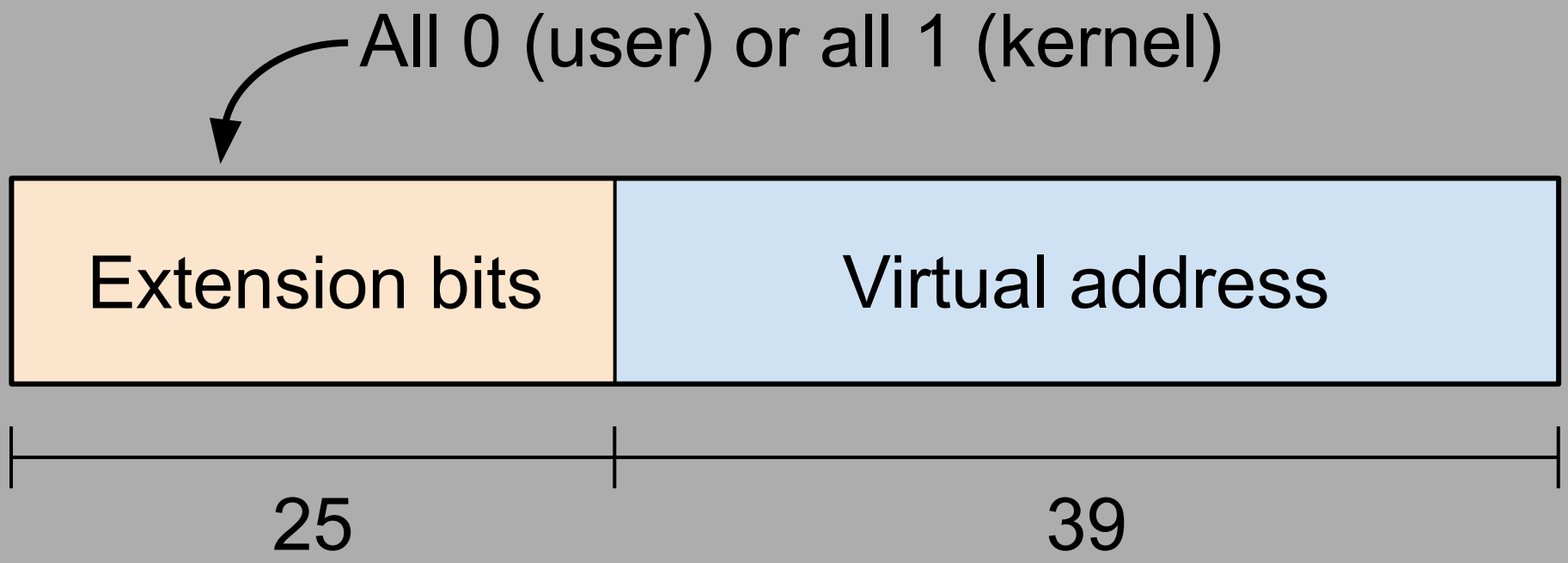
recovering kernel code execution

bazad

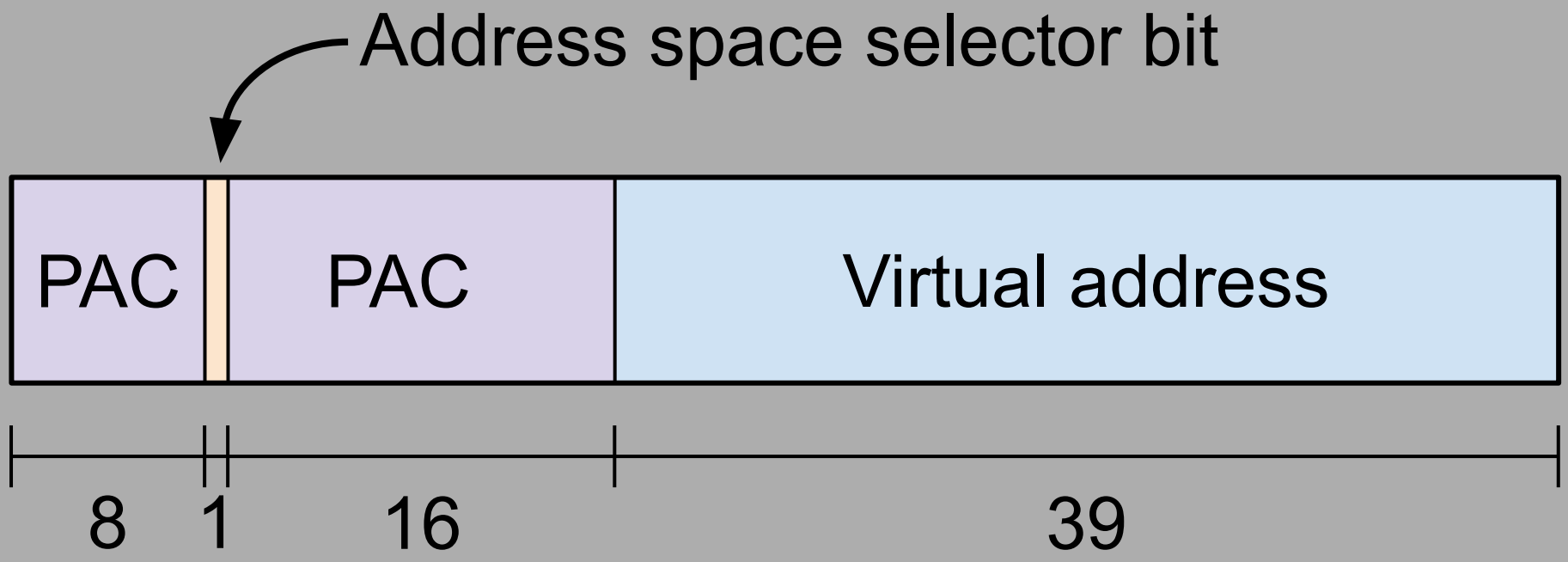
0x41con 2019

PAC on the A12

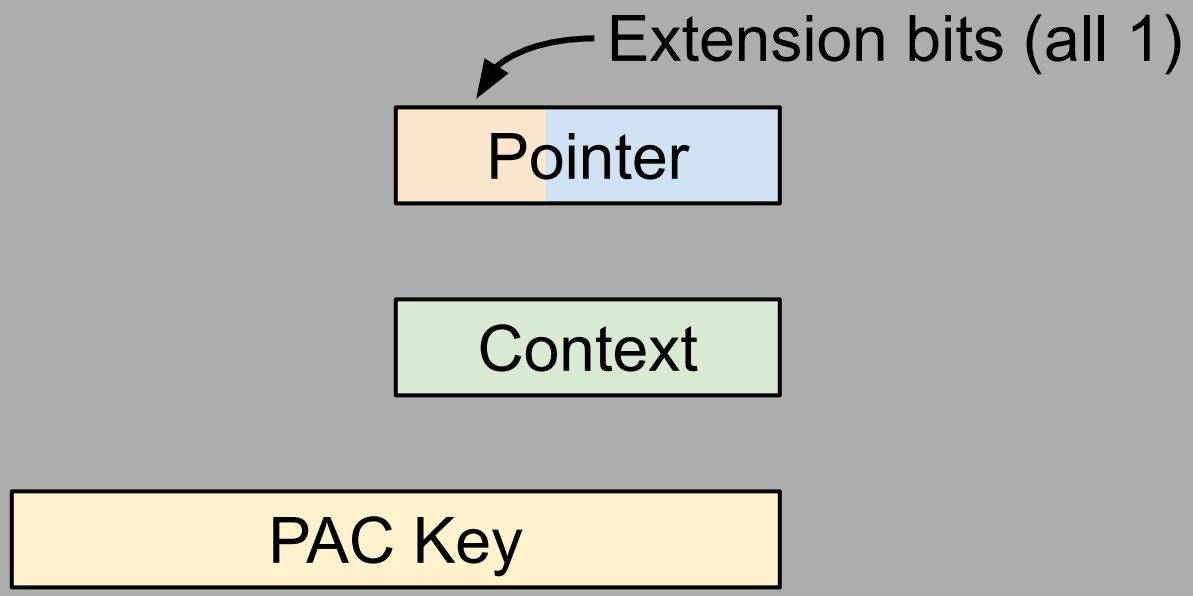
Pointer layout



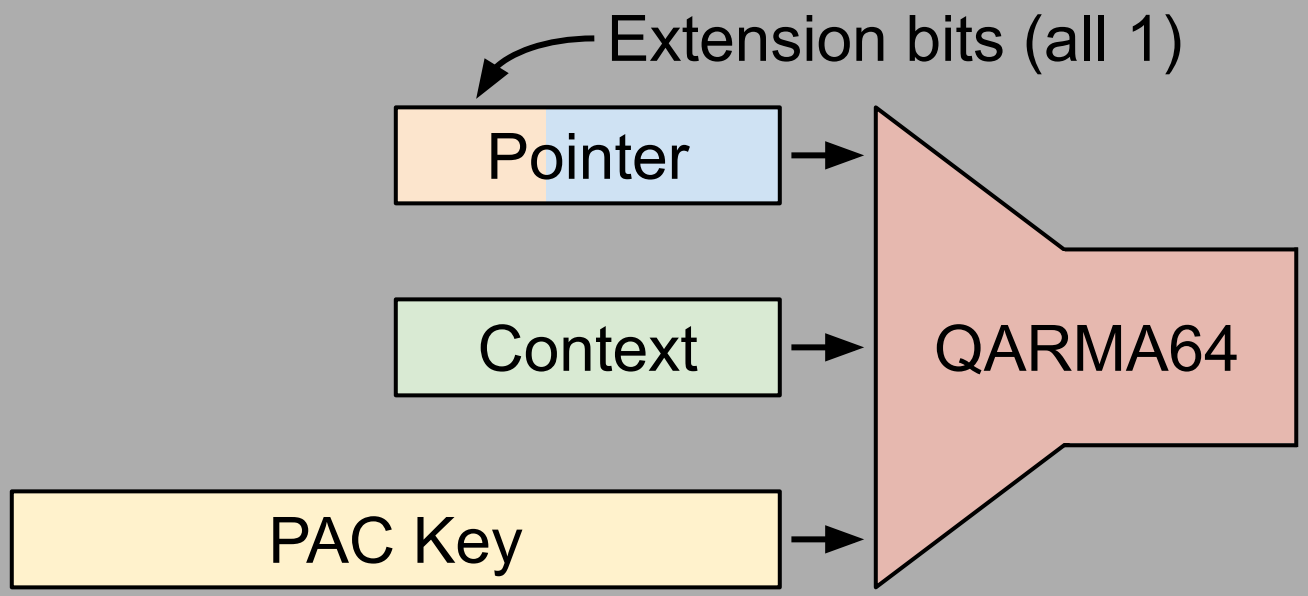
Pointer layout



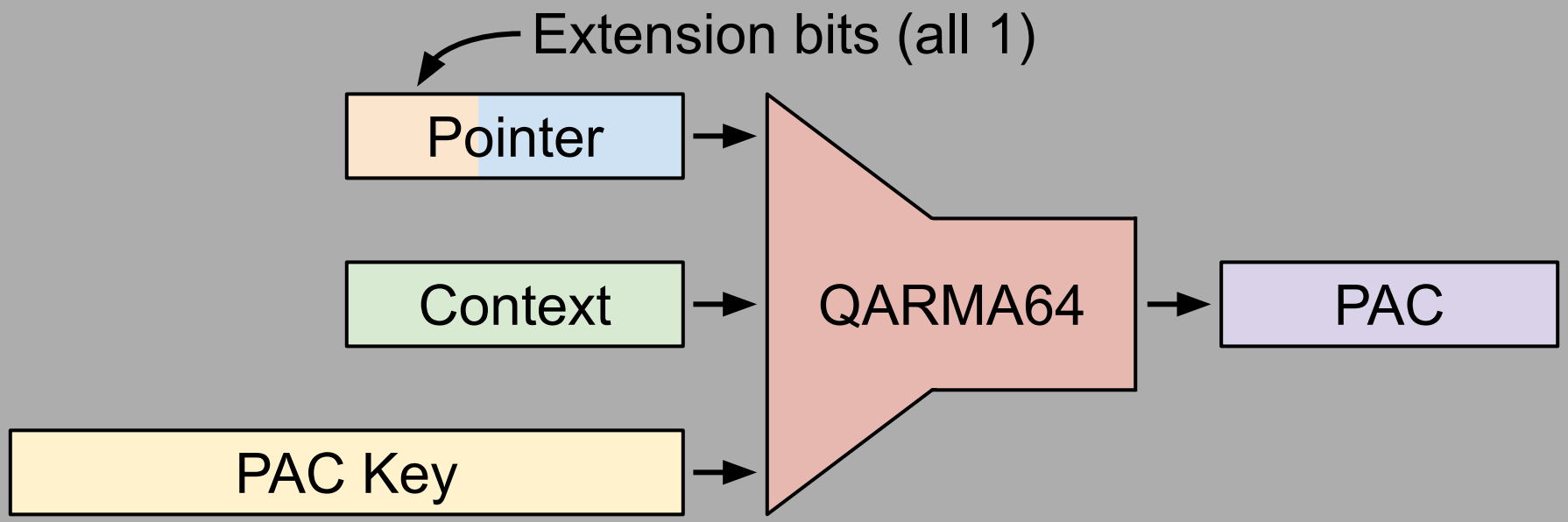
Pointer Authentication Codes



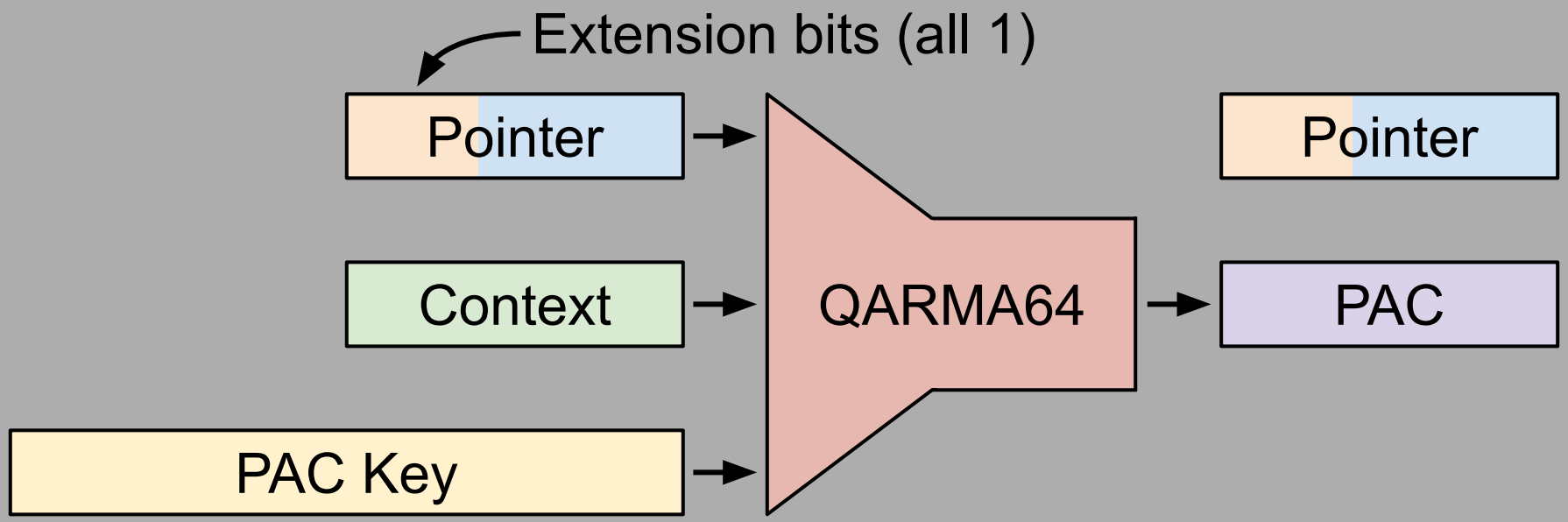
Pointer Authentication Codes



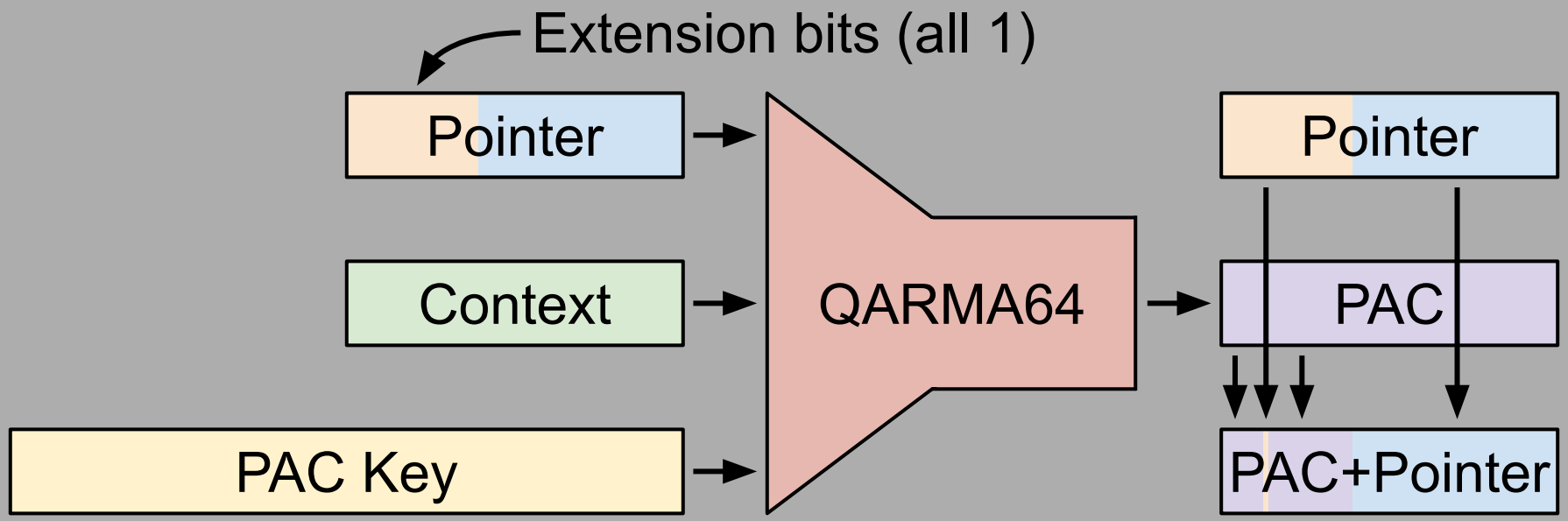
Pointer Authentication Codes



Pointer Authentication Codes



Pointer Authentication Codes



Apple implementation has further hardening

Bypass 3

All code pointers
in writable memory
must be protected

```
LEXT(_bcopyin)
    ARM64_STACK_PROLOG
    PUSH_FRAME
    SET_RECOVERY_HANDLER x10, x11, x3, copyio_error
...
    sub         x2, x2, #16
1:
    /* 16 bytes at a time */
    ldp         x3, x4, [x0], #16
    stp         x3, x4, [x1], #16
    subs        x2, x2, #16
    b.ge        1b
...
    CLEAR_RECOVERY_HANDLER x10, x11
    mov         x0, #0
    POP_FRAME
    ARM64_STACK_EPILOG
```

```
LEXT(_bcopyin)
    ARM64_STACK_PROLOG
    PUSH_FRAME
    SET_RECOVERY_HANDLER x10, x11, x3, copyio_error
...
    sub         x2, x2, #16
1:
    /* 16 bytes at a time */
    ldp         x3, x4, [x0], #16
    stp         x3, x4, [x1], #16
    subs        x2, x2, #16
    b.ge        1b
...
    CLEAR_RECOVERY_HANDLER x10, x11
    mov         x0, #0
    POP_FRAME
    ARM64_STACK_EPILOG
```

Code to execute if a fault occurs

```
.macro SET_RECOVERY_HANDLER
    mrs     $0, TPIDR_EL1           // Load thread pointer
    ldr     $1, [$0, TH_RECOVER]    // Save previous recovery handler
    adrp    $2, $3@page             // Load the recovery handler address
    add     $2, $2, $3@pageoff
    str     $2, [$0, TH_RECOVER]    // Set new recovery handler
.endmacro
```

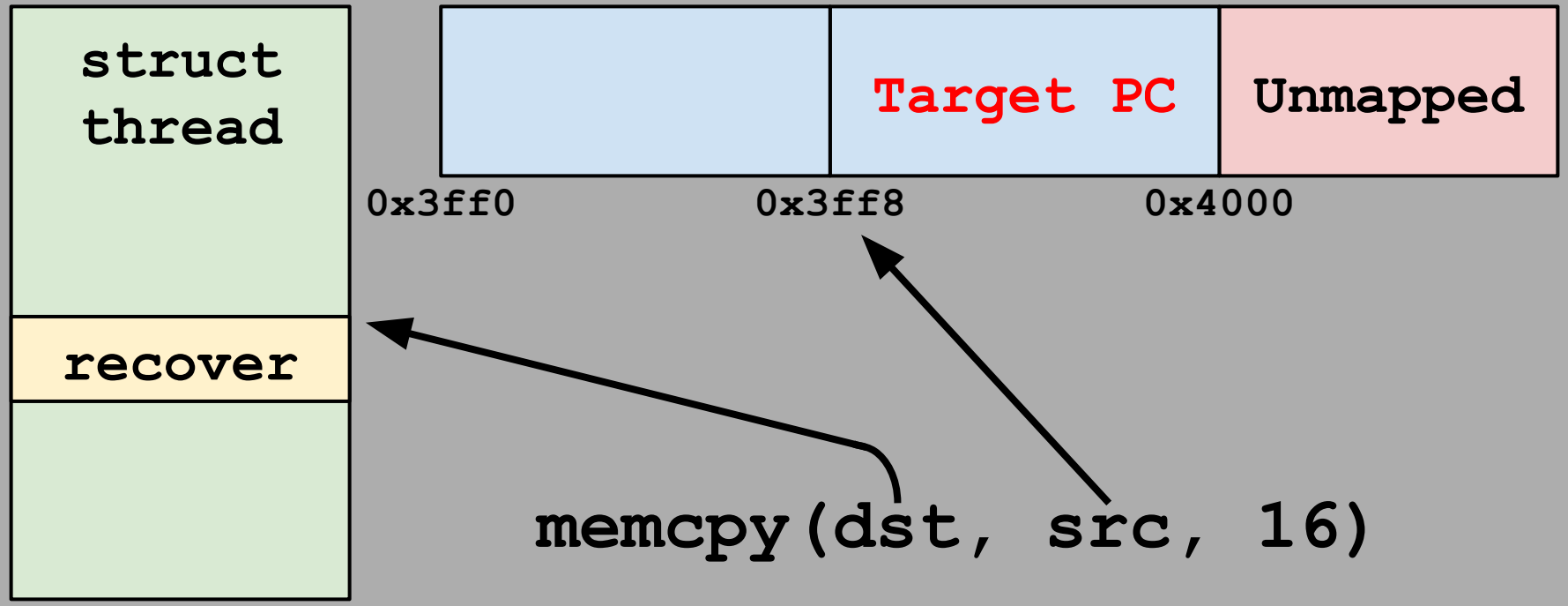
```
.macro SET_RECOVERY_HANDLER
    mrs     $0, TPIDR_EL1           // Load thread pointer
    ldr     $1, [$0, TH_RECOVER]    // Save previous recovery handler
    adrp    $2, $3@page             // Load the recovery handler address
    add     $2, $2, $3@pageoff
    str     $2, [$0, TH_RECOVER]    // Set new recovery handler
.endmacro
```

Raw function pointer
is stored in
thread struct

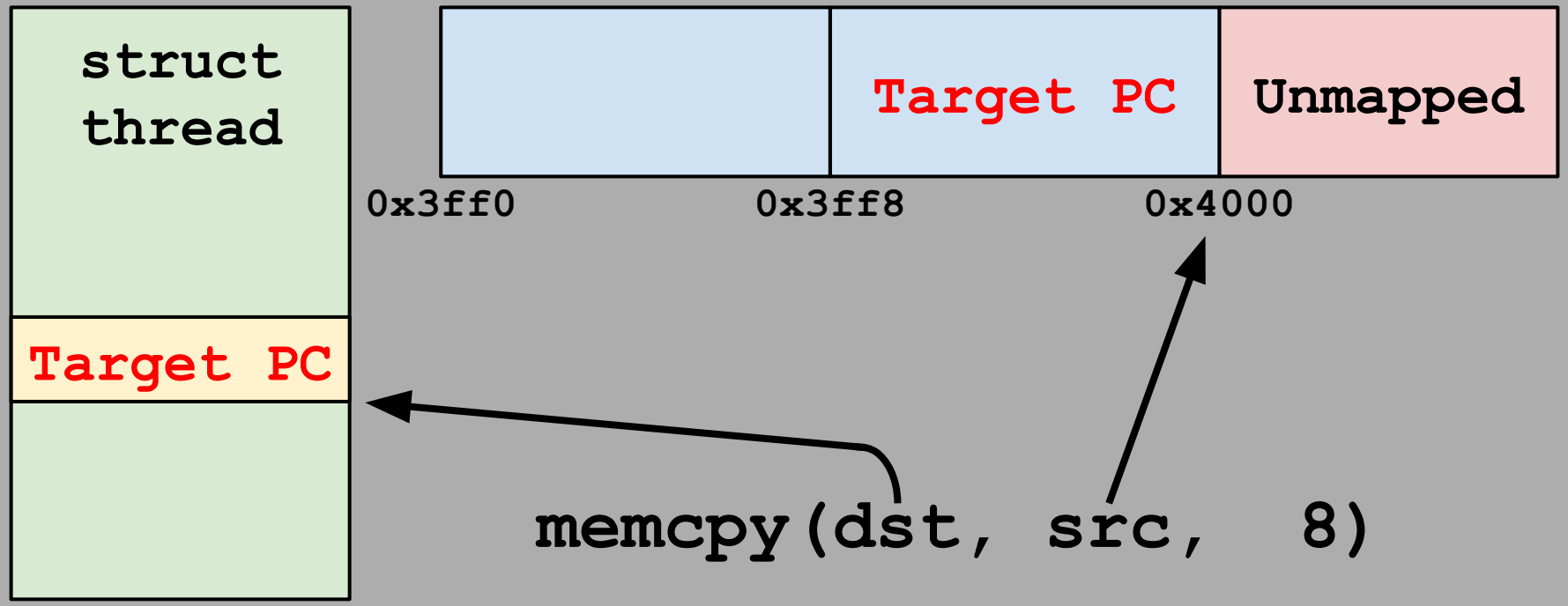
```
.macro SET_RECOVERY_HANDLER
    mrs     $0, TPIDR_EL1           // Load thread pointer
    ldr     $1, [$0, TH_RECOVER]    // Save previous recovery handler
    adrp    $2, $3@page             // Load the recovery handler address
    add     $2, $2, $3@pageoff
    str     $2, [$0, TH_RECOVER]    // Set new recovery handler
.endmacro
```

No PAC protection!

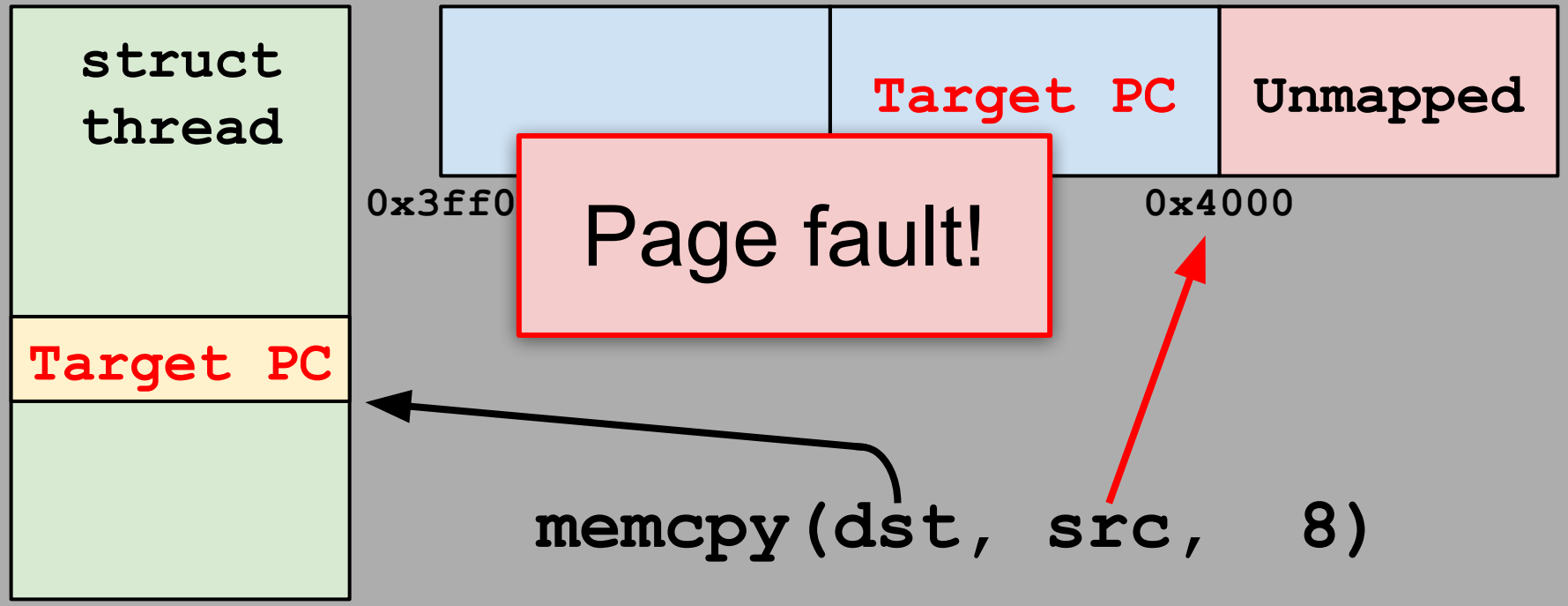
Bypass 3


struct
thread
recover
Target PC
Unmapped
0x3ff0
0x3ff8
0x4000
memcpy(dst, src, 16)

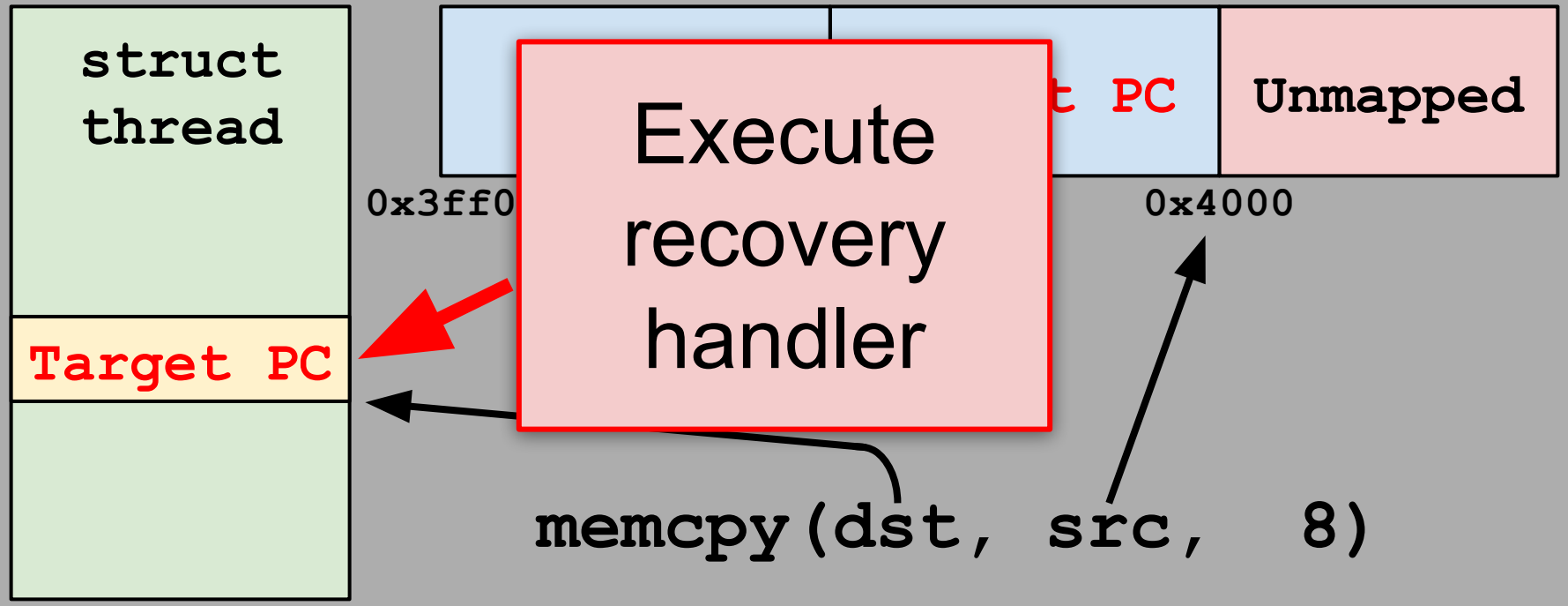
Bypass 3


struct thread
Target PC
Target PC
Unmapped
0x3ff0
0x3ff8
0x4000
memcpy(dst, src, 8)

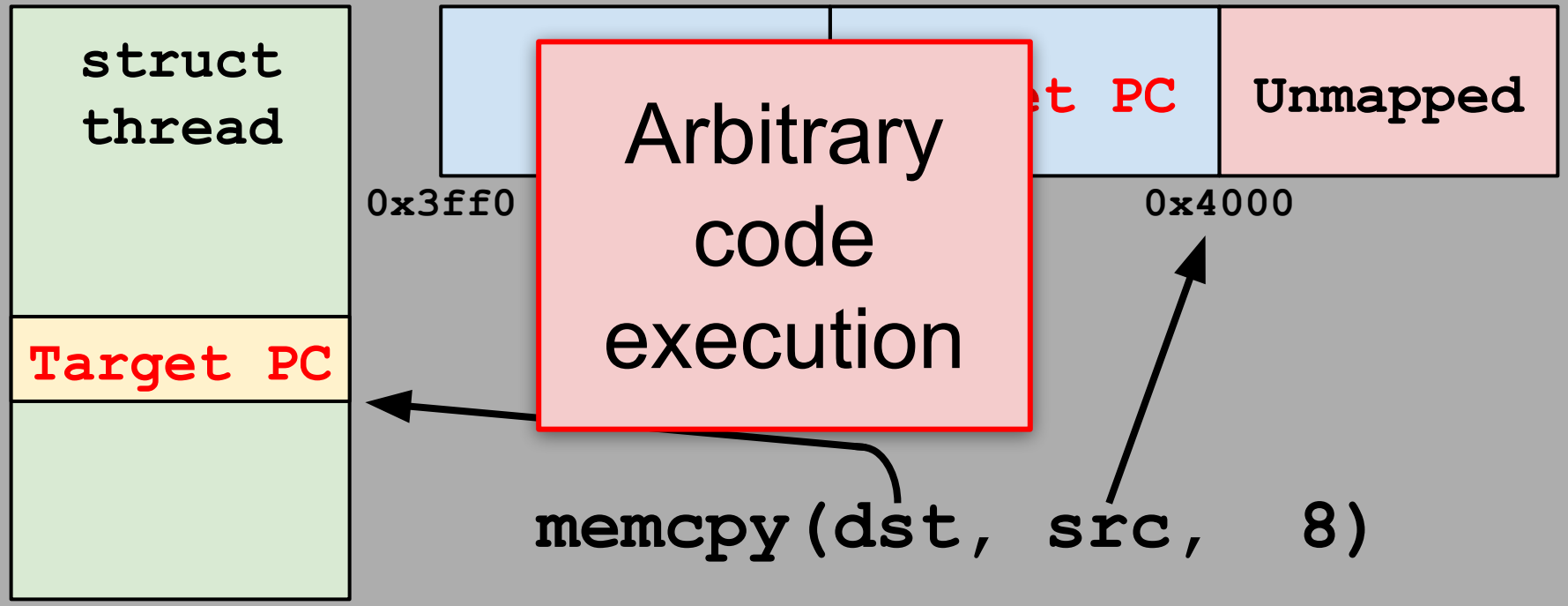
Bypass 3


struct thread
Target PC
Target PC
Unmapped
0x3ff0
0x4000
Page fault!
memcpy(dst, src, 8)

Bypass 3


struct thread
Target PC
t PC
Unmapped
0x3ff0
0x4000
Execute recovery handler
memcpy(dst, src, 8)

Bypass 3


struct thread
Target PC
0x3ff0
t PC
Unmapped
0x4000
Arbitrary code execution
memcpy(dst, src, 8)

DEMO

Home Moments Search Twitter Have an account? Log in

Tarjei Mandt
@kernelpool

Follow

iOS 12.3 beta 3 broke IDA graph view of switch statements for arm64e. Bonus points to people who can figure out the reason for the compiler change.

4:43 PM - 23 Apr 2019

2 Retweets **10** Likes

2 10

1,980 382 16.8K 904

Follow

Tarjei Mandt
@kernelpool

Senior Security Researcher

Sydney, Australia

mista.nu/research

Joined August 2009

Tweets Tweets & replies Media

Tarjei Mandt @kernelpool · Apr 23
If you know the answer and enjoy challenging state-of-the-art security mitigations, you should probably come work for us :)

1 1 3

Tarjei Mandt @kernelpool · Apr 23

New to Twitter?

Sign up now to get your own personalized timeline!

Sign up