REPRODUCING SPECTRE ATTACK WITH GEM5

PIERRE AYOUB - EURECOM

PIERRE.AYOUB@EURECOM.FR

CLÉMENTINE MAURICE - UNIV LILLE, CNRS, INRIA

CLEMENTINE.MAURICE@INRIA.FR

ICHSA'21

01 JUNE, 2021

PROCESSOR'S MICROARCHITECTURE IS A BLACK BOX

Consequence: Micro-architectural security is **hard**

PROCESSOR'S MICROARCHITECTURE IS A BLACK BOX

Consequence: Micro-architectural security is **hard**

Transient Instruction

PROCESSOR'S MICROARCHITECTURE IS A BLACK BOX

Consequence: Micro-architectural security is **hard**

Transient Instruction

⇒ Affects the processor **micro-architectural state** – leaving its architectural state as prior the execution

PROCESSOR'S MICROARCHITECTURE IS A BLACK BOX

Consequence: Micro-architectural security is **hard**

Transient Instruction

⇒ Affects the processor **micro-architectural state** – leaving its architectural state as prior the execution

Spectre

PROCESSOR'S MICROARCHITECTURE IS A BLACK BOX

Consequence: Micro-architectural security is **hard**

Transient Instruction

⇒ Affects the processor **micro-architectural state** – leaving its architectural state as prior the execution

Spectre

⇒ Execute malicious transient instructions exploiting the **branch predictor**

SIMULATION CAN BREAK THIS BLACK BOX

How: Allowing the user to view the micro-architecture's behavior

GEM5

GEM5

Cycle-accurate simulator

GEM5

Cycle-accurate simulator

⇒ Simulate very precisely **hardware** entirely **in software**

GEM5

Cycle-accurate simulator

⇒ Simulate very precisely **hardware** entirely **in software**

Build a system

GEM5

Cycle-accurate simulator

⇒ Simulate very precisely **hardware** entirely **in software**

Build a system

⇒ Instantiate and parameterize **Python objects**

GEM5

Cycle-accurate simulator

⇒ Simulate very precisely **hardware** entirely **in software**

Build a system

⇒ Instantiate and parameterize **Python objects**

Run a system

GEM5

Cycle-accurate simulator

⇒ Simulate very precisely **hardware** entirely **in software**

Build a system

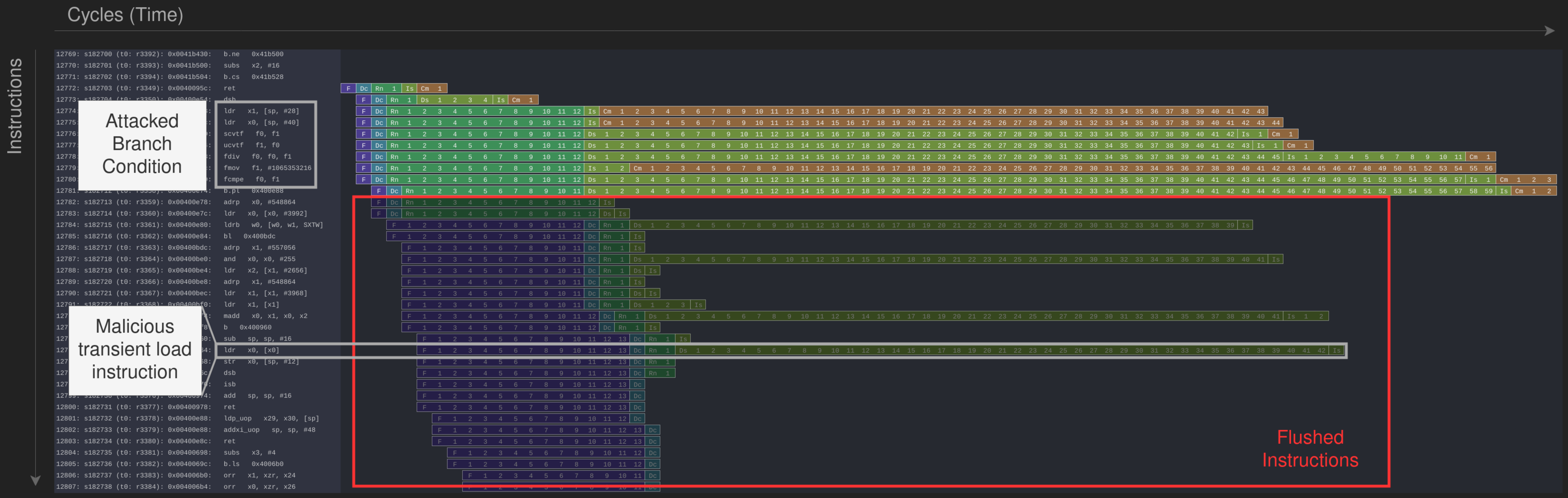
⇒ Instantiate and parameterize **Python objects**

Run a system

⇒ Launch the Python script, then **view and inspect** the running system!

OBSERVING SPECTRE WITH KONATA

SUCCESS SCENARIO



```
12769: s182700 (t0: r3392): 0x0041b430:   b.ne   0x41b500
12770: s182701 (t0: r3393): 0x0041b500:   subs   x2, #16
12771: s182702 (t0: r3394): 0x0041b504:   b.cs   0x41b528
12772: s182703 (t0: r3349): 0x0040095c:   ret
12773: s182704 (t0: r3350): 0x00400e54:   dsb
12774                                     ldr   x1, [sp, #28]
12775                                     ldr   x0, [sp, #40]
12776                                     scvtf   f0, f1
12777                                     ucvtf   f1, f0
12778                                     fdiv   f0, f0, f1
12779                                     fmov   f1, #1065353216
12780                                     fcmpe   f0, f1
12781: s182712 (t0: r3358): 0x00400e74:   b.pl   0x400e88
12782: s182713 (t0: r3359): 0x00400e78:   adrp   x0, #548864
12783: s182714 (t0: r3360): 0x00400e7c:   ldr   x0, [x0, #3992]
12784: s182715 (t0: r3361): 0x00400e80:   ldrb   w0, [w0, w1, SXTW]
12785: s182716 (t0: r3362): 0x00400e84:   bl   0x400bdc
12786: s182717 (t0: r3363): 0x00400bdc:   adrp   x1, #557056
12787: s182718 (t0: r3364): 0x00400be0:   and   x0, x0, #255
12788: s182719 (t0: r3365): 0x00400be4:   ldr   x2, [x1, #2656]
12789: s182720 (t0: r3366): 0x00400be8:   adrp   x1, #548864
12790: s182721 (t0: r3367): 0x00400bec:   ldr   x1, [x1, #3968]
12791: s182722 (t0: r3368): 0x00400bf0:   ldr   x1, [x1]
127                                    :   madd   x0, x1, x0, x2
127                                    8   b   0x400960
127                                    60:   sub   sp, sp, #16
127                                    64:   ldr   x0, [x0]
127                                    68:   str   x0, [sp, #12]
127                                    6c:   dsb
127                                    70:   isb
12799: s182730 (t0: r3376): 0x00400974:   add   sp, sp, #16
12800: s182731 (t0: r3377): 0x00400978:   ret
12801: s182732 (t0: r3378): 0x00400e88:   ldp_uop   x29, x30, [sp]
12802: s182733 (t0: r3379): 0x00400e88:   addxi_uop   sp, sp, #48
12803: s182734 (t0: r3380): 0x00400e8c:   ret
12804: s182735 (t0: r3381): 0x00400698:   subs   x3, #4
12805: s182736 (t0: r3382): 0x0040069c:   b.ls   0x4006b0
12806: s182737 (t0: r3383): 0x004006b0:   orr   x1, xzr, x24
12807: s182738 (t0: r3384): 0x004006b4:   orr   x0, xzr, x26
```

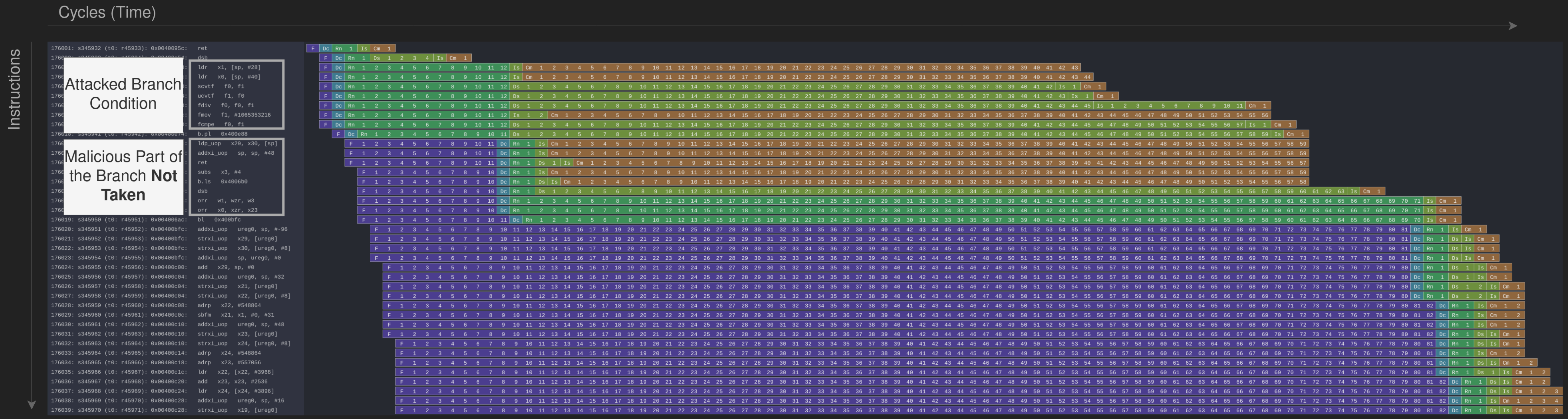
Attacked Branch Condition

Malicious transient load instruction

Flushed Instructions

Allows to understand how the attack works

DEFEATED BY THE BRANCH PREDICTOR

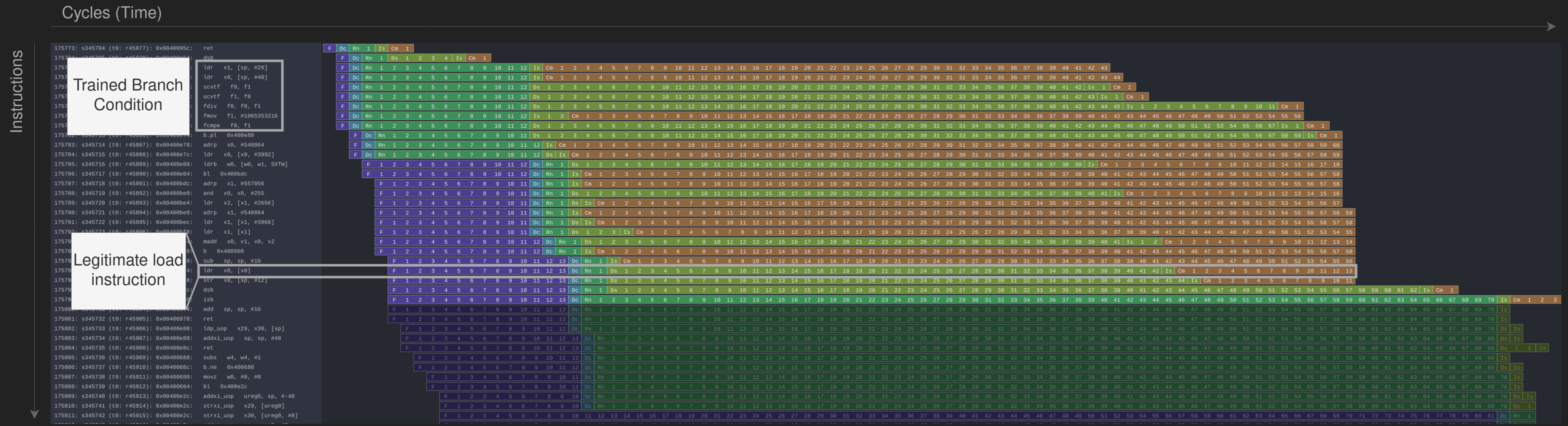


Cycles (Time)
Instructions
Attacked Branch Condition
Malicious Part of the Branch **Not Taken**

```
176001: s345932 (t0: r45933): 0x0040095c:   ret
1760                                      :   dsb
1760                                      :   ldr   x1, [sp, #28]
1760                                      :   ldr   x0, [sp, #40]
1760                                      :   scvtf   f0, f1
1760                                      :   ucvtf   f1, f0
1760                                      :   fdiv   f0, f0, f1
1760                                      :   fmov   f1, #1065353216
1760                                      :   fcmpe   f0, f1
176010: s345941 (t0: r45942): 0x00400e74:   b.pl   0x400e88
1760                                      :   ldp_uop   x29, x30, [sp]
1760                                      :   addxi_uop   sp, sp, #48
1760                                      :   ret
1760                                      :   subs   x3, #4
1760                                      :   b.ls   0x4006b0
1760                                      :   dsb
1760                                      :   orr   w1, wzr, w3
1760                                      :   orr   x0, xzr, x23
176019: s345950 (t0: r45951): 0x004006ac:   bl   0x400bfc
176020: s345951 (t0: r45952): 0x00400bfc:   addxi_uop   ureg0, sp, #-96
176021: s345952 (t0: r45953): 0x00400bfc:   strxi_uop   x29, [ureg0]
176022: s345953 (t0: r45954): 0x00400bfc:   strxi_uop   x30, [ureg0, #8]
176023: s345954 (t0: r45955): 0x00400bfc:   addxi_uop   sp, ureg0, #0
176024: s345955 (t0: r45956): 0x00400c00:   add   x29, sp, #0
176025: s345956 (t0: r45957): 0x00400c04:   addxi_uop   ureg0, sp, #32
176026: s345957 (t0: r45958): 0x00400c04:   strxi_uop   x21, [ureg0]
176027: s345958 (t0: r45959): 0x00400c04:   strxi_uop   x22, [ureg0, #8]
176028: s345959 (t0: r45960): 0x00400c08:   adrp   x22, #548864
176029: s345960 (t0: r45961): 0x00400c0c:   sbfm   x21, x1, #0, #31
176030: s345961 (t0: r45962): 0x00400c10:   addxi_uop   ureg0, sp, #48
176031: s345962 (t0: r45963): 0x00400c10:   strxi_uop   x23, [ureg0]
176032: s345963 (t0: r45964): 0x00400c10:   strxi_uop   x24, [ureg0, #8]
176033: s345964 (t0: r45965): 0x00400c14:   adrp   x24, #548864
176034: s345965 (t0: r45966): 0x00400c18:   adrp   x23, #557056
176035: s345966 (t0: r45967): 0x00400c1c:   ldr   x22, [x22, #3968]
176036: s345967 (t0: r45968): 0x00400c20:   add   x23, x23, #2536
176037: s345968 (t0: r45969): 0x00400c24:   ldr   x24, [x24, #3896]
176038: s345969 (t0: r45970): 0x00400c28:   addxi_uop   ureg0, sp, #16
176039: s345970 (t0: r45971): 0x00400c28:   strxi_uop   x19, [ureg0]
```

Allows to visualize the root cause of a failed attack!

TRAINING SCENARIO



```
175773: s345704 (t0: r45877): 0x0040095c:   ret
175774: s345705 (t0: r45878): 0x00400e54:   dsb
1757                                    :   ldr   x1, [sp, #28]
1757                                    :   ldr   x0, [sp, #40]
1757                                    :   scvtf   f0, f1
1757                                    :   ucvtf   f1, f0
1757                                    :   fdiv   f0, f0, f1
1757                                    :   fmov   f1, #1065353216
1757                                    :   fcmpe   f0, f1
175782: s345713 (t0: r45886): 0x00400e74:   b.pl   0x400e88
175783: s345714 (t0: r45887): 0x00400e78:   adrp   x0, #548864
175784: s345715 (t0: r45888): 0x00400e7c:   ldr   x0, [x0, #3992]
175785: s345716 (t0: r45889): 0x00400e80:   ldrb   w0, [w0, w1, SXTW]
175786: s345717 (t0: r45890): 0x00400e84:   bl   0x400bdc
175787: s345718 (t0: r45891): 0x00400bdc:   adrp   x1, #557056
175788: s345719 (t0: r45892): 0x00400be0:   and   x0, x0, #255
175789: s345720 (t0: r45893): 0x00400be4:   ldr   x2, [x1, #2656]
175790: s345721 (t0: r45894): 0x00400be8:   adrp   x1, #548864
175791: s345722 (t0: r45895): 0x00400bec:   ldr   x1, [x1, #3968]
175792: s345723 (t0: r45896): 0x00400bf0:   ldr   x1, [x1]
17579                                  4:   madd   x0, x1, x0, x2
17579                                  8:   b   0x400960
17579                                  0:   sub   sp, sp, #16
17579                                  4:   ldr   x0, [x0]
17579                                  8:   str   x0, [sp, #12]
17579                                  c:   dsb
17579                                  0:   isb
17580                                  4:   add   sp, sp, #16
175801: s345732 (t0: r45905): 0x00400978:   ret
175802: s345733 (t0: r45906): 0x00400e88:   ldp_uop   x29, x30, [sp]
175803: s345734 (t0: r45907): 0x00400e88:   addxi_uop   sp, sp, #48
175804: s345735 (t0: r45908): 0x00400e8c:   ret
175805: s345736 (t0: r45909): 0x00400688:   subs   w4, w4, #1
175806: s345737 (t0: r45910): 0x0040068c:   b.ne   0x400680
175807: s345738 (t0: r45911): 0x00400680:   movz   w0, #0, #0
175808: s345739 (t0: r45912): 0x00400684:   bl   0x400e2c
175809: s345740 (t0: r45913): 0x00400e2c:   addxi_uop   ureg0, sp, #-48
175810: s345741 (t0: r45914): 0x00400e2c:   strxi_uop   x29, [ureg0]
175811: s345742 (t0: r45915): 0x00400e2c:   strxi_uop   x30, [ureg0, #8]
```

We can even identify more scenarios…

FAITHFULNESS OF THE SIMULATION

FAITHFULNESS OF THE SIMULATION

Comparing a real system and a simulation

FAITHFULNESS OF THE SIMULATION

Comparing a real system and a simulation

⇒ Very **similar result** of number of **mispredicted branches**

⇒ **Helped** to **implement** the attack on a **real system**

FAITHFULNESS OF THE SIMULATION

Comparing a real system and a simulation

⇒ Very **similar result** of number of **mispredicted branches**

⇒ **Helped** to **implement** the attack on a **real system**

Limitations

FAITHFULNESS OF THE SIMULATION

Comparing a real system and a simulation

⇒ Very **similar result** of number of **mispredicted branches**

⇒ **Helped** to **implement** the attack on a **real system**

Limitations

⇒ **Slow** simulation, possibly **inaccurate** models

CONCLUSION

PAPER

Reproducing Spectre Attack with gem5

How To Do It Right?

Pierre Ayoub
pierre.ayoub@eurecom.fr
EURECOM
Sophia Antipolis, France

Clémentine Maurice
clementine.maurice@inria.fr
Univ Lille, CNRS, Inria
Lille, France

Pierre Ayoub and Clémentine Maurice. *Reproducing Spectre Attack with gem5: How To Do It Right?* (**EuroSec ’21**), April 26, 2021

- **GitHub**: https://pierreay.github.io/reproduce-spectre-gem5/
- **Contact**: pierre.ayoub@eurecom.fr