Fixing* Hardware Trace Reconstruction Issues for Runtime Code?

Suchakra Sharma

15th Sept 2017
Tracing/BPF Micro Conference, Linux Plumbers, LA
Hardware Trace with Intel PT

1. CPU
2. Intel PT Hardware
3. Intel PT Software Decoder
4. Trace Packets
5. Reconstructed Execution Flow
6. Binary
... Intel Processor Trace data: size 8544 bytes

- 00000000: 02 82 02 82 02 82 02 82 02 82 02 82 02 82 02 82 PSB
- 00000010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 PAD
- 00000016: 19 ba 39 4d 7b 89 5e 04 TSC 0x45e897b4d39ba
- 0000001e: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 PAD
- 00000026: 02 73 57 64 00 1c 00 00 TMA CTC 0x6457 FC 0x1c
- 0000002e: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 PAD
- 00000030: 02 03 27 00 CBR 0x27
- 00000034: 02 23 PSBEND
- 00000036: 59 8b MTC 0x8b
- 00000038: 59 8c MTC 0x8c

- 00000304: f8 TNT TTTTNN (6)
- 00000305: 06 00 00 TNT T (1)
- 00000308: 4d e0 3c 6d 9c TIP 0x9c6d3ce0
- 0000030d: 1c 00 00 TNT TTN (3)
- 00000310: 2d f0 3c TIP 0x3cf0
- 00000313: 06 TNT T (1)
- 00000314: 59 2e MTC 0x2e
- 00000316: 94 TNT NNTNTN (6)
- 00000317: a8 TNT NTNTNN (6)
- 00000318: a6 TNT NTNNTT (6)
Current Limitations

Reconstruction requires file backed executable code

Runtime Compiled Code
- Needs compiler specific APIs to regularly copy code cache for later reconstruction
- Can be done with code instrumentation that allows dumping runtime code
- Failed reconstruction

Self-Modifying Code
- Lack of updated copy of code section
- Wrong reconstruction
With Runtime Compilation

CPU

Intel PT Hardware

Intel PT Software Decoder

Trace Packets
- TNT - T
- TNT - N

Static Code
- jnz
- add
- nop
- jz

Reconstructed Execution Flow

Binary
With Runtime Compilation

CPU

Intel PT Hardware

Intel PT Software Decoder

Trace Packets

TNT - T
TNT - N

Static Code

jnz
add
nop
jz

Runtime Generated Code

Reconstructed Execution Flow

Binary

2.b
With Runtime Compilation

CPU
Intel PT Hardware

Intel PT Software Decoder

Trace Packets
TNT - T
TNT - N

Static Code
jnz
add
nop
jz

Runtime Generated Code

Reconstructed Execution Flow

Binary
With Runtime Compilation

Intel PT Software Decoder

Trace Packets
- TNT - T
- TNT - N

Static Code
- jnz
- add
- nop
- jz

Runtime Generated Code

Binary

Reconstructed Execution Flow
With Self Modifying Code

CPU

Intel PT Hardware

Intel PT Software Decoder

Trace Packets

TNT - T
TNT - N

Static Code

jnz
add
jmp
jz

Binary

Reconstructed Execution Flow
With Self Modifying Code

CPU

Intel PT Hardware

Intel PT Software Decoder

Trace Packets

TNT - T
TNT - N

Static Code

jnz
add
nop
jz

Binary

Reconstructed Execution Flow

2.c

Suchakrapani Datt Sharma
Possible Solution - FlowJIT

Target Process
- Runtime Code

ioctl()

Page Access Control
- Tracked Pages
  - NX
  - NX

PF Handler
  - X
  - X

Trace Decoder
- ID
- Timestamp
- Instruction Pointer
- Runtime Code

Userspace
- Query

Kernel

FlowJIT Events
- 3.a

Suchakrapani Datt Sharma
uBPF Example

; PT Re-construction Output
.
..
4c 89 73 18          mov %r14, 0x18(%rbx)
48 83 c4 30          add $0x30, %rsp
5b                      pop %rbx
4c 89 e0              mov %r12, %rax
5d                      pop %rbp
41 5c                  pop %r12
41 5d                  pop %r13
41 5e                  pop %r14
  c3                   ret
  48 85 c0          ; return from ubpf_compile()
  74 55                  ; we found JIT fn
  48 8b 74 24 10             ; prepare arguments
  4c 89 ff         ; we call the JIT function
  ff d0                  (7f33654ce000 in this case)

..  
.
.
7f33654ce000: error no memory mapped at this address

This needs to be resolved
uBPF Example

; Raw PT Packets
.
.
tip 3: 400ff7 ; main+673 (instructions just preceding the call to JITed code)

pad

tnt8 N
pad
tip 3: 7f33654ce000 ; fn == NULL? No, so go ahead

pad

fup 3: 7f33654ce000 ; Generated because it is a compound PT packet as FilterEn was set

pad
.
.
tip.pgd 0: 0 ; Packet Generation Disabled
tip.pge 1: e000 ; Packet Generation Enabled for previous FUP (IP compressed here)
tnt8 NNNNNN ; These should be 100 as per the uBPF program loop
tnt8 NNNNNN ; We have the image of program in jit_data->buf
tnt8 NNNNNN

Hardware trace has this value

Dumped from kernel patch

Test Program Loop in Runtime JITed code
uBPF Example

; as i == 100, the loop exits

; some function

; printf result of the filter
Limitations & Discussions

Initial patch (v4.7) : https://github.com/tuxology/flowjit

- Needs to be in critical sections (PF handler etc.)
- No proper mechanism to dump code pages yet
  - Maybe use Perf aux-buffer
  - Only 1 page code dump
- Limited tests (uBPF & static-key instrumentation)
- Similar approach by mmiotrace
  - Wrap ioremap (mmio-mod.c)
  - Elegant registration, handlers (kmmio.c)
- Integrate with perf for better usability?
Fin

suchakra@shiftleft.io
suchakrapani.sharma@polymtl.ca
@tuxology