Attack on ROB and Prefetcher

Meet Udeshi, Nirmal Boran
Prof. Virendra Singh

CADSL - IIT Bombay

May 29, 2019

Disabling Prefetcher to avoid Cache Pollution

- Cache side-channels work by extracting information about cache accesses of victim process
- Any other process or hardware block which accesses the cache will increase noise in the side-channel
- Fuchs et al. [1] have proposed a method to make prefetcher increase noise in cache and disrupt cache side-channel attacks
- We propose to disable the prefetcher by not allowing it to learn the stride patterns
- This can be done by running a third process in parallel which makes random loads at PC which alias with victim process' load PCs

[1] Disruptive prefetching: impact on side-channel attacks and cache designs - SYSTOR'15

Prefetcher attack implementation

- Place load instructions at multiple PC address to fill up the prefetcher table
- Randomize stride for every load address
- Restrict load address to single cache line by keeping stride less than 64 bytes
- Ensure cache miss every time by flushing before load using `clflush`
- Use `nop` instructions to align loads to desired PC address

- Drawback: targets whole prefetcher and not few entries of victim

Prefetcher attack implementation

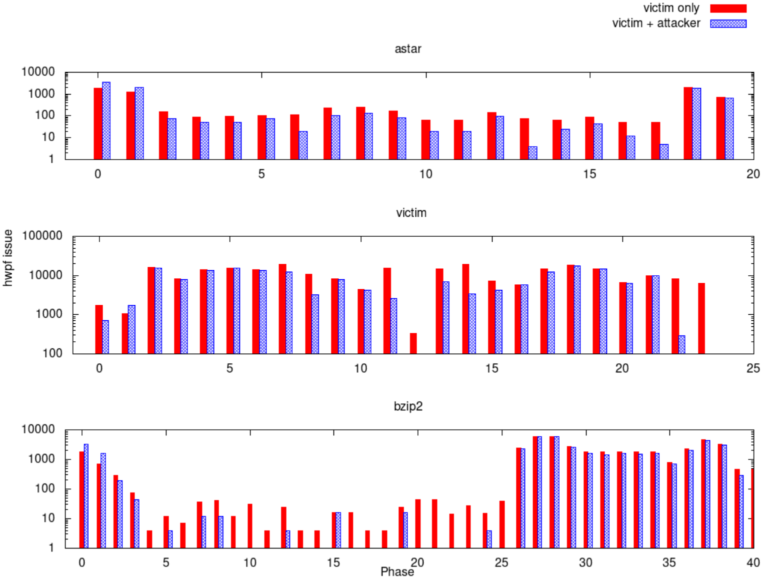
```
00000000000006ca <attack>:
 6ce:   8b 58 36          mov     0x36(%rax),%ebx
 6d1:   90                nop
 6d2:   0f ae 78 36       clflush 0x36(%rax)
 6d6:   8b 58 08          mov     0x8(%rax),%ebx
 6d9:   90                nop
 6da:   0f ae 78 08       clflush 0x8(%rax)
 6de:   8b 58 3f          mov     0x3f(%rax),%ebx
 6e1:   90                nop
 6e2:   0f ae 78 3f       clflush 0x3f(%rax)
 6e6:   8b 58 38          mov     0x38(%rax),%ebx
 6e9:   90                nop
 6ea:   0f ae 78 38       clflush 0x38(%rax)
 6ee:   8b 58 20          mov     0x20(%rax),%ebx
 6f1:   90                nop
```

Simulation setup

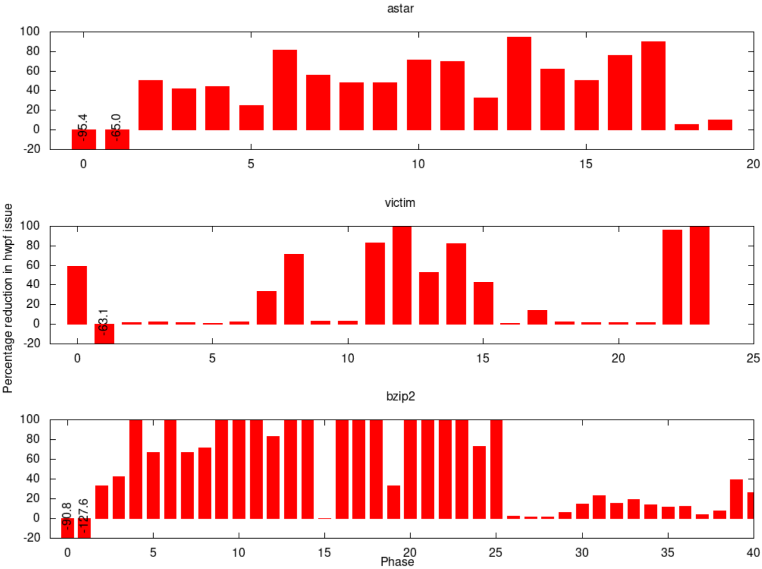
- Simulator: gem5 X86
- Cores: 2
- L1: 32K icache 32K dcache private to each core
- L2: 256K 16-way shared between cores
- L2prefetcher: Stride 64-entry 4-way, confidence threshold 4

- Victim process runs on core1 and attacker runs on core2
- Number of prefetches issued are measured for every 100,000 instructions of victim process

Result plots: Number of prefetches



Result plots: Percent reduction



Future work

- Analyse why prefetches are not reducing to 0
- Devise attack tailored to victim's loads
- Try with openssl AES code

The End