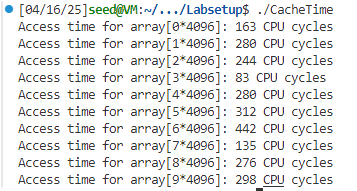
|  |
| --- |
| 哈尔滨工业大学(深圳) |
| **《网络与系统安全》 实验报告** |
|  |
| 实验一  Spectre Attack 实验  学 院: 计算机科学与技术学院   |  |  | | --- | --- | | 姓 名: | 刘睿 | | 学 号: | 220110720 | | 专 业: | 计算机科学与技术 | | 日 期: | 2025年3月 | |

# 一、实验过程

任务1：从缓存读取数据与从内存读取数据的比较

第一次运行结果



经过多次运行后统计

缓存命中项（array[3] 和 array[7]）的平均访问时间约为 308.7 cycles

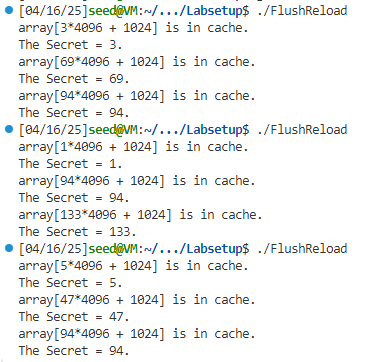
非命中项的平均访问时间约为 398.3 cycles

由此设定一个阈值350cycles来区分两种类型的内存访问

任务2：使用缓存作为侧信道

设定 #define CACHE\_HIT\_THRESHOLD (200)

前三次运行结果

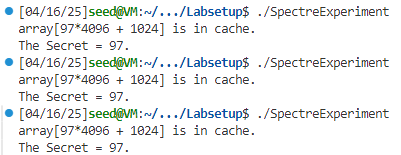


对程序进行了 20 次独立运行，观察 array[94\*4096 + 1024] 是否被判定为“在缓存中”。其中在 18 次运行中成功检测到 Secret = 94

少数情况下存在其他缓存命中（例如 array[69], array[133]），属于系统噪声影响

任务3：乱序执行与分支预测

未经过修改后多次运行结果得到



每次都能成功检测到 array[97] 被缓存，说明 victim 函数中 本不应被执行的访问指令 被 CPU 推测性执行

注释掉\_\_mm\_clflush(&size)后，运行结果如下



当注释掉 \_mm\_clflush(&size) 后，CPU 能快速获取 size 的值，预测窗口被消除，推测执行不会发生，array[97] 不再被缓存

将第④行替换为victim(i+20)，运行结果如下



当将训练数据改为 victim(i+20)（i+20 > size），CPU 分支预测器不会预测 if 条件为真，推测执行也不会发生

任务4：Spectre攻击

前三次运行结果如下



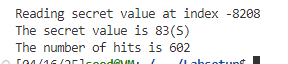
程序成功检测到缓存中命中的位置为 array[83\*4096 + 1024]，说明 restrictedAccess() 函数中的非法访问在 CPU 推测执行中被提前执行

缓存的命中位置恰好对应 secret 的第一个字符 'S'（ASCII 83），说明攻击成功

任务5：提高攻击准确性

编译并运行SpectreAttackImproved.c程序，运行结果





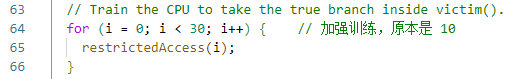
1. 为什么最高分可能是scores[0]，如何修复

scores[0] 出现最高分的情况通常意味着：

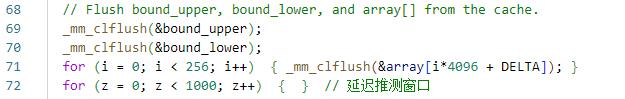
CPU 未预测执行非法访问，restrictedAccess(index\_beyond) 返回了 0，不是 secret。也就是 推测执行失败了，真正的 secret 没有被访问到，从而缓存命中的是 array[0\*4096 + DELTA]。最终 array[0] 的分数暴涨，导致你读出了 The secret value is 0()，而不是 'S'

修复方案

加强训练



延迟推测窗口

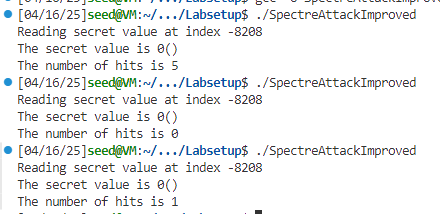


强制非法访问并泄露 secret



1. printf(“\*”)看似无用，但对攻击有影响，可能的原因是？

注释掉后的运行结果



可能的原因是·

printf("\*\*\*\*\*\n"); 会扰动分支预测器的状态或刷新指令缓冲区，从而间接提高 Spectre 攻击的成功率

1. usleep(10)的作用与观测

我将休眠时间依次增大到100、1000、2000，个人的体感是，随着休眠时间的增长，攻击命中率有提升。这可能是因为更长的休眠时间有助于 CPU 缓存状态稳定、缓存探测测时更精准，并降低系统噪声对测量精度的影响

任务6：窃取整个秘密字符串

# 二、对本次实验的问题或建议