spectre-v5笔记

- [RISC-V CPU侧信道攻击原理与实践（8）---Spectre-V5 - 知乎 (zhihu.com)](#)
- 攻击者和受害者都向一个寄存器写入一个值，并调用函数 in_place。该函数从寄存器中检索值，并将其作为输入传递给 usleep 函数。在调用 in_place 时，攻击者和受害者的返回地址都会被压入返回堆栈缓冲区（RSB）。当受害者先醒来时，它返回并从 RSB 中弹出最后一个地址，这导致它错误地猜测返回到攻击者的返回地址，从而泄露了秘密数据。
- 内联函数是一种特殊类型的函数，它的代码在编译时直接插入调用它的地方，而不是通过常规的函数调用机制。所以in_place函数声明为noinline
- main-gadget.c
- 连续两个pop使函数返回两个调用层次（回到main函数），然后将返回地址flush掉，引发推测执行，预测的是gadget函数的下一条指令temp &= array[*secret * 512];

```
void gadget()
{
  // return to main
  asm volatile(
      "pop %rdi\n"//从栈中弹出一个值并将其存储到寄存器 rdi 中
      "pop %rbp\n"//将栈顶的值弹出到基指针寄存器 rbp 中。rbp 通常用于保存当前栈帧的基址
      "nop\n"
      //使用 clflush 指令清除栈顶地址（即当前栈指针 rsp 指向的地址）中的数据。
      //这是为了确保这个地址的数据被从缓存中删除，后续对这个地址的访问将会是从主内存中读取，而不是从缓存中读取，从而造成更明显的时间差异。
      "clflush (%rsp)\n"
      "retq\n");//用于返回，实际上是将控制权返回到调用这个 gadget 的地方
}
```