CES571S - Spectre - Final Report

467261 - Yifu Wang

2018 - 12 - 19

1 Introduction

Spectre is a vulnerability that affects modern microprocessors that perform branch prediction. On most processors, the speculative execution resulting from a branch misprediction may leave observable side effects that may reveal private data to attackers. For example, if the pattern of memory accesses performed by such speculative execution depends on private data, the resulting state of the data cache constitutes a side channel through which an attacker may be able to extract information about the private data using a timing attack.

Spectre attack can be conduct on browser victims. But this require attacker to send an attacker controled code to victims and execute it localy. Furthermore this attack can only retrieve the data from the process which runing the attacker controled code due to the seperation policy defined by mordern browser.

Howerver a derivation of spectre attack, netspectre, a generic remote Spectre variant 1 attack, could allow attackers to read arbitrary memory from the systems available on the network containing the required Spectre gadgets—a code that performs operations like reading through an array in a loop with bounds check on each iteration.

The original goal of this project is to setup a spectre attack cross processes, which now I knew impractical. So instead I implement a PoC of netspectre attack.

The project repository is held by this link.

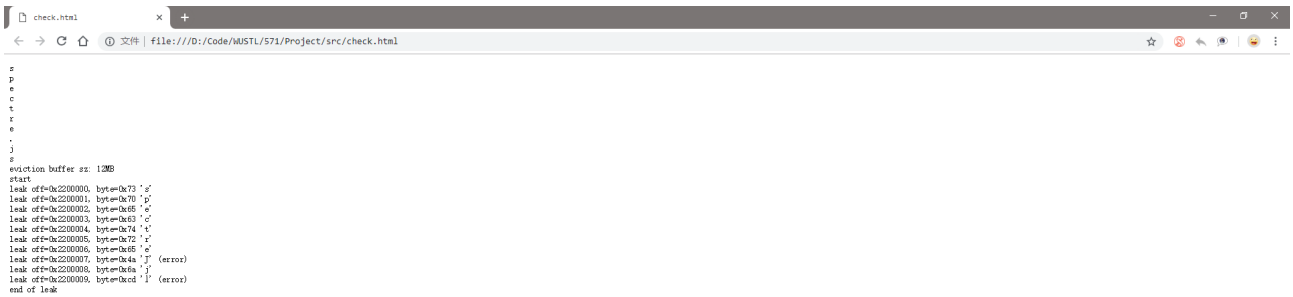
2 Implement local attack

The local attack is fairly easy to implement. By using C/C++, you can directly execute assmembly language in your program, which allow to clear cache and measure system ticks manually, both of which are very importent for a successful spectre attack.

```
PS D:\Program Files> .\spectre.exe
Putting 'Final Project Spectre: CSE 571S' in memory, address 0000000000405000
Reading 31 bytes:
00000000000000FC0 Success: 0x46='F' score=2
00000000000000FC1 Success: 0x69='i' score=2
00000000000000FC2 Success: 0x6E='n' score=2
00000000000000FC3 Success: 0x61='a' score=2
00000000000000FC4 Success: 0x6C='l' score=2
00000000000000FC5 Success: 0x20=' ' score=2
00000000000000FC6 Success: 0x50='P' score=2
00000000000000FC7 Success: 0x72='r' score=2
00000000000000FC8 Success: 0x6F='o' score=2
00000000000000FC9 Success: 0x6A='j' score=2
00000000000000FCA Success: 0x65='e' score=2
00000000000000FCB Success: 0x63='c' score=2
00000000000000FCC Success: 0x74='t' score=2
00000000000000FCD Success: 0x20=' ' score=2
00000000000000FCE Success: 0x53='S' score=2
00000000000000FCF Success: 0x70='p' score=2
00000000000000FD0 Success: 0x65='e' score=2
00000000000000FD1 Success: 0x63='c' score=2
00000000000000FD2 Success: 0x74='t' score=2
00000000000000FD3 Success: 0x72='r' score=2
00000000000000FD4 Success: 0x65='e' score=2
00000000000000FD5 Success: 0x3A=':' score=2
00000000000000FD6 Success: 0x20=' ' score=2
00000000000000FD7 Success: 0x43='C' score=2
00000000000000FD8 Success: 0x53='S' score=2
00000000000000FD9 Success: 0x45='E' score=2
00000000000000FDA Success: 0x20=' ' score=2
00000000000000FDB Success: 0x35='5' score=2
00000000000000FDC Success: 0x37='7' score=2
00000000000000FDD Success: 0x31='1' score=2
00000000000000FDE Success: 0x53='S' score=2
PS D:\Program Files>
```

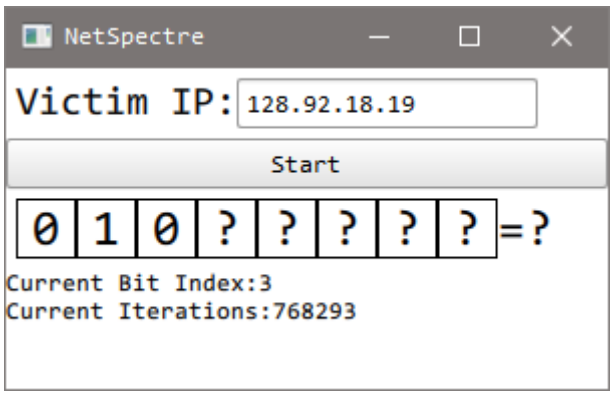
3 Implement browser attack

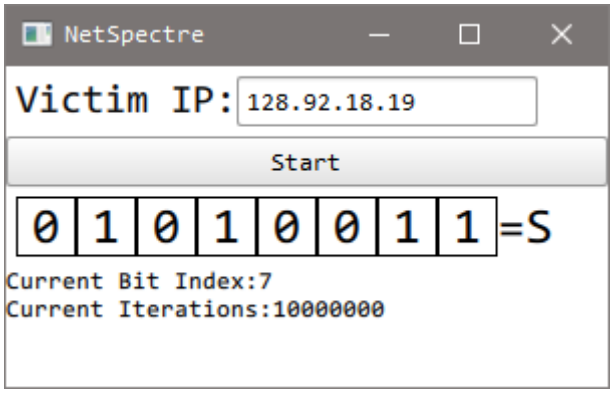
Since JavaScript couldn't manually control the caches. We have to evict cache before every iteration using some unrelavent junk data block. And since chrome has already patched this vulnerability by disabled SharedArray default. You need to enable it from setting and then you maight want to check if your browser is vulnerabel to spectre by this site.



4 Implement netspectre attack

Basicly to complete a netspectre attack you need so call 'gadget' be pre-insert into user's machine. Which will listen to attacker's remote packet to mistraining the predictor and send back the bits value. Meanwhile the attacker side will measure the response timing and repeat this process as many as possible until the result is satisfide.





5 Conclution

At present, spectre or netspectre is not a very serious problem that will endanger most common user. Since both of them need to pre-send some attacker code into user's machine. And even though netspectre attack claims to be able to retreive any data from a machine, but it's extremely slow. It's only reveal 15 bits per hour, namely 1 GB per 60822 years.

6 Reference

[1]:Spectre
[2]:NetSpectre