CS3235-19

Side-channel attack on kernel space ASLR

Eldon Ng Kai Foo, Li Yunfan, Low Tian Wei, Yeo Zhuan Yu

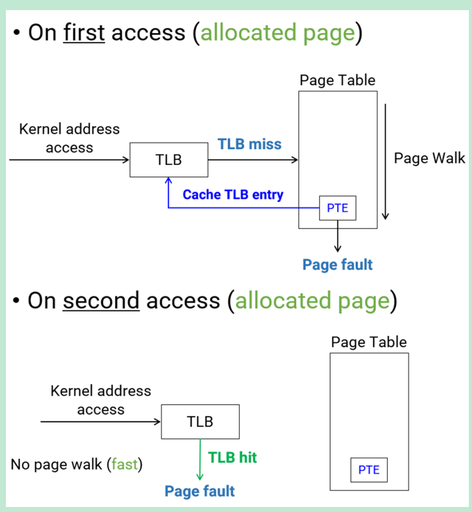
INTRODUCTION

- Most modern computers have implemented Address Space Layout Randomization (ASLR).
- This prevent attackers from knowing the specific address of a target data at runtime.
- However, this method of obscuring the address space may be susceptible to side-channel attacks (E.g. Meltdown & Spectre).
- We explore two types of side-channel timing attack:
  1. Using page faults[1]
  2. Using Intel TSX[2]

METHODS

Using page faults[1]

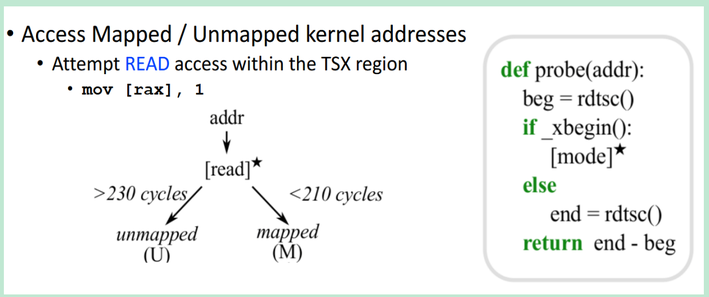
- Double page faults allow us to check whether a kernel address is allocated.



- Normally, accessing a kernel address from user space would result in a page fault.
- Accessing an allocated kernel address would cause the page table entry to be cached in the TLB first before generating a page fault due to wrong privilege level.
- The result is that a TLB hit would take lesser time compared to a TLB miss (unallocated kernel address).
- Recording the difference in timing allows us to map out the kernel space.

Using Intel TSX[2]

- Use TSX as an exception handler instead of the default way by the OS.
- Record the timestamp before accessing the kernel memory.
- When the abort occurs, measure the difference in timing at the exception handler.



Source: https://www.blackhat.com/docs/us-16/materials/us-16-Jang-Breaking-Kernel-Address-Space-Layout-Randomization-KASLR-With-Intel-TSX.pdf

RESULTS

Using page faults[1]

- We repeat the attack multiple times and average it out to reduce any possible inaccuracies from noise or outliers.

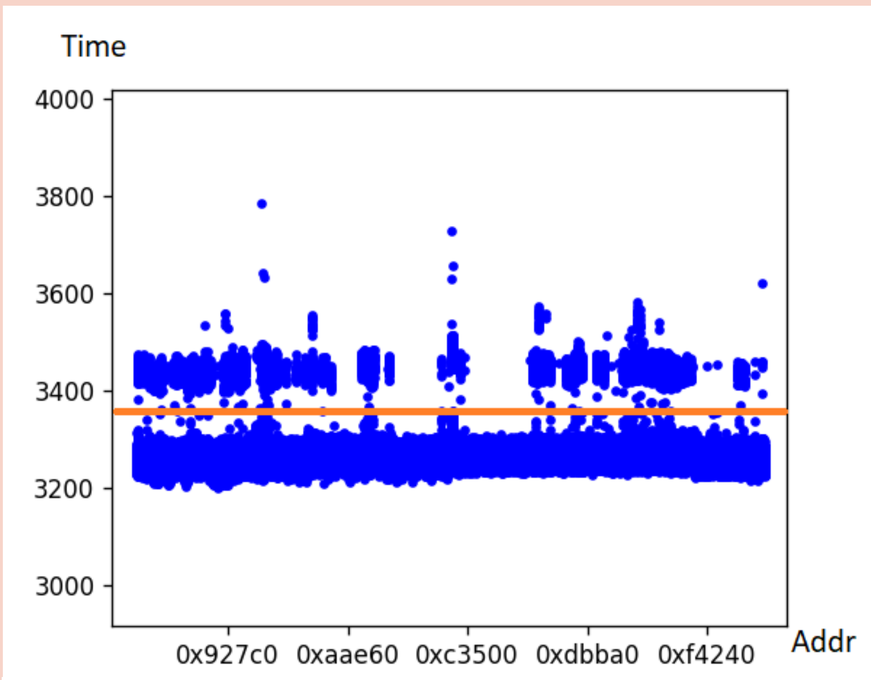


Figure 1: Graph using double page faults

- Figure 1 shows that even though we took noise and outliers into consideration, the differences in timing is not obvious enough.
- This makes it challenging for us to be able to deduce the correct kernel address accurately.

Using Intel TSX[2]

- Like the double page fault attack, we repeat multiple times to reduce inaccuracies.

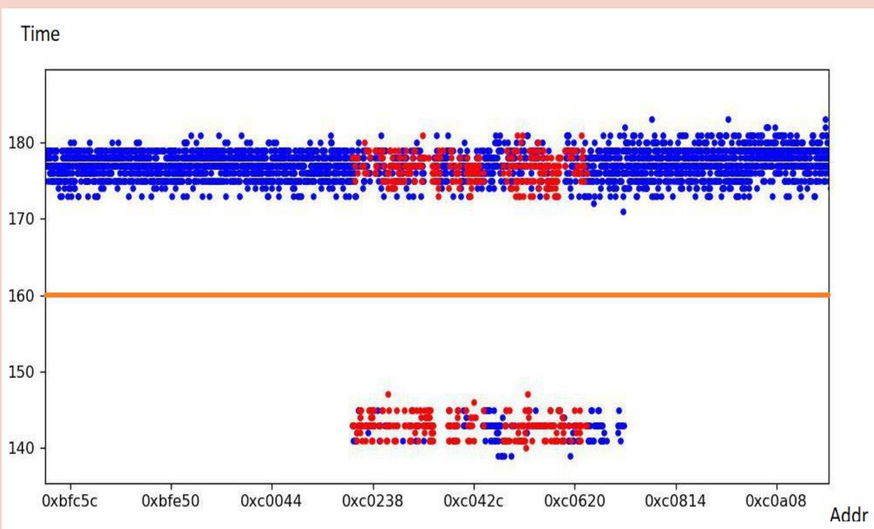


Figure 2: Graph using TSX

- As seen in Figure 2, if we take the minimum values (red dots) for kernel addresses, there is a distinct difference. We can clear out the noise and non-kernel addresses to map out the kernel space.

MITIGATION

Here are three ways to prevent the aforementioned side-channel timing attacks:

1. Perform access permission check first before the MMU creates a page table entry, thereby preventing a TLB entry from being created in this way.
2. Modify the execution time of the OS execution handler so that the timing between an allocated and unallocated kernel space will be the same.
3. Modern computers have Kernel page-table isolation (KPTI), which isolates kernel and user address space to prevent users from accessing kernel memory[3].

CONCLUSION

- We provided a proof-of-concept to show how easy it is to derandomize Linux kernel ASLR.
- An attacker could inject a malicious program to your computer and extract out sensitive information (E.g. passwords, emails, or personal photos).
- Side-channel attacks should not be taken lightly, since there are still computers susceptible to such exploits.

FURTHER INFORMATION

For more information, please visit our GitHub repository:

https://github.com/thefoggycity/cs3235-aslr

REFERENCES

- Hund, R., et al., 2013. Practical Timing Side Channel Attacks Against Kernel Space ASLR. IEEE Symposium on Security and Privacy, 2013. http://doi.org/10.1109/SP.2013.23
- Jang, Y., et al., 2016. DrK: Breaking Kernel Address Space Layout Randomization with Intel TSX. In 23rd ACM Conference on Computer and Communications Security. https://doi.org/10.1145/2976749.2978321
- Larabel, M., 2018. Further Analyzing The Intel CPU "x86 PTI Issue" On More Systems. Retrieved from https://www.phoronix.com/scan.php?page=article&item=linux-more-x86pti.