Brandenburg
University of Technology
Cottbus - Senftenberg

Seminar: Advanced Topics in Network and System Security

ZombieLoad

Tomi Jerenko

Supervisor: Eric Strehle

Cottbus, 15.7.2020

Table of contents

1. Introduction
2. Memory Mapping, Cache, Buffers
3. Related Work
4. ZombieLoad
5. Results
6. Countermeasures
7. Conclusion
8. References

Introduction

- CPU performance optimizations: out of order and speculative execution.
- Exploited by Meltdown and Spectre in 2018.
- Performance and security don’t go hand in hand.
- Extended by microarchitectural data sampling side channel attacks.

Virtual to Physical Memory Mapping

- Map memory to: isolate, access contiguously, save space, separate permissions.

- Memory Management Unit for translating, Translation Lookaside Buffer for caching.

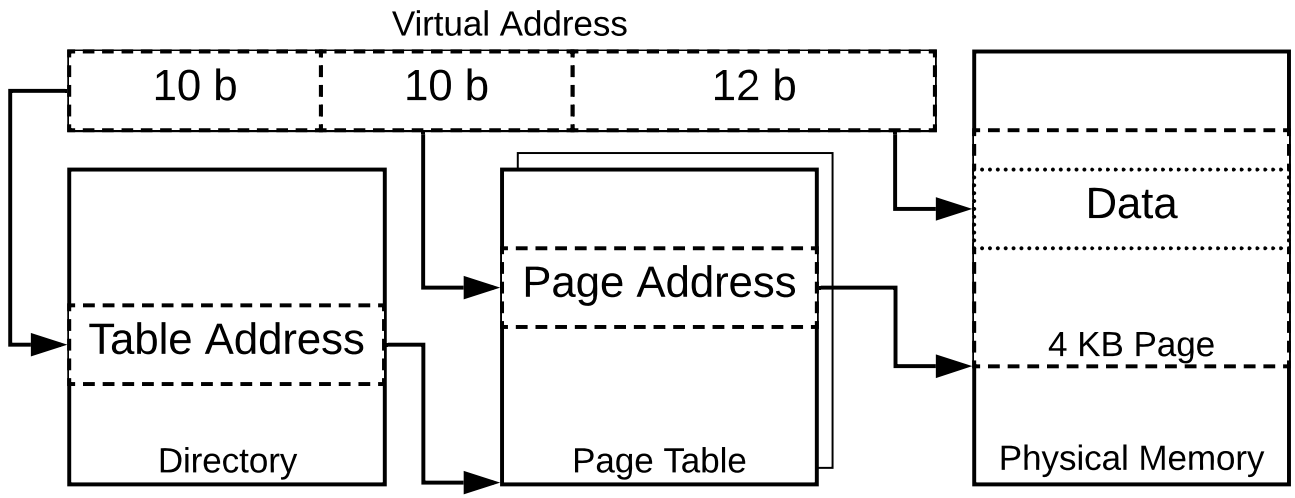

Fig. Virtual to Physical Address Translation

Cache

- Smaller and faster temporary storage.
- Can be used as a side channel (e.g. FLUSH+RELOAD).

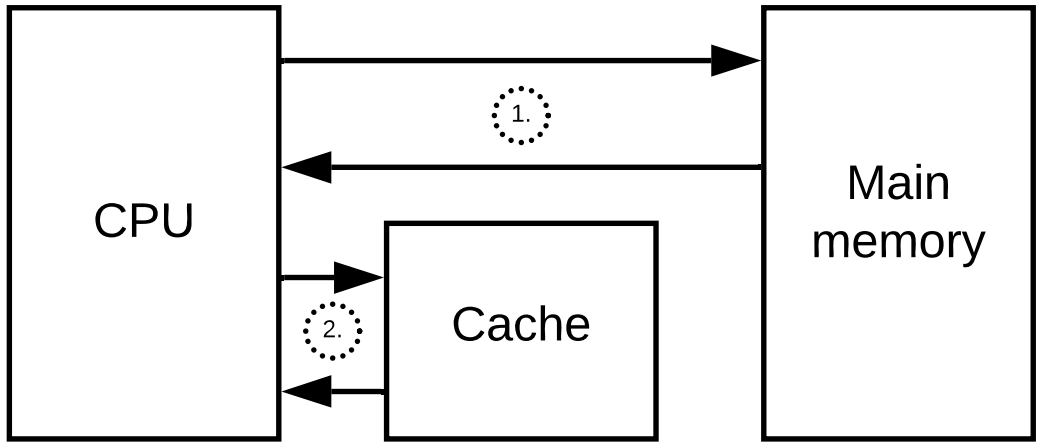

Fig. Main Memory vs. Cache Access Speed

Buffers

- Queue with extra performance optimizations.

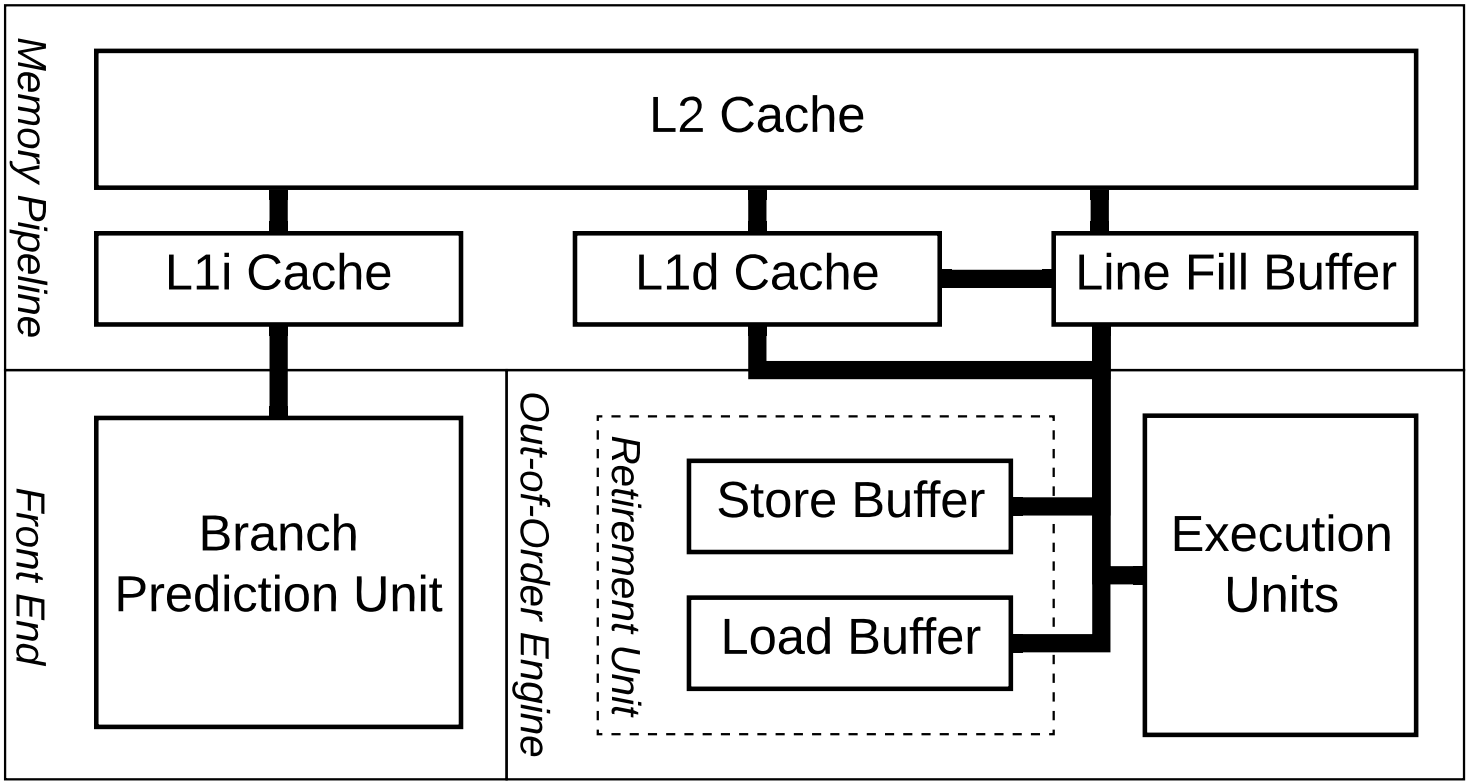

Fig. Simplified Overview of Intel Skylake Architecture [RIDL].

RIDL and Fallout

- Transient out of order execution.
- Leaks user and kernel memory.
- Builds upon the logic of Meltdown attack.
- Leaks in-flight data from line-fill buffer.
- Targets Intel CPUs.

- Speculative branch execution.
- Leaks user and kernel memory.
- Built from Meltdown and Spectre.
- Leaks pending writes from store buffer.
- Targets Intel CPUs.

ZombieLoad

- Transient out of order execution.
- Leaks in-flight data from line-fill buffer.
- Speculative matching LFB data to load instructions.
- 5 variants.
- Data sampling (very little control over leaked data).
- Cache side channel attack to extract secrets.

Data Leak Flows

- User process to user process.
- Kernel space to user space.
- SGX enclave to outside.
- Guest VM to guest VM.
- Host VM to guest VM.

Attack Variants: V1 - Kernel Mapping

- Shared pages: user address u and kernel address k translate to the same physical address.

- Flushing u and accessing k results in zombie load.

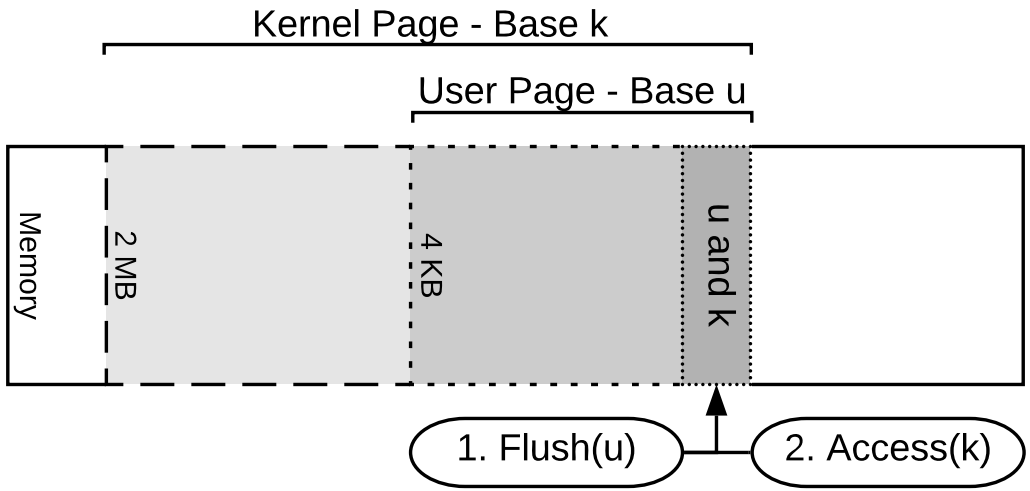

Fig. Kernel and user shared page.

Attack Variants: V2 - Intel TSX

- User virtual address u with valid mapping required.
- Flushing u within TSX transaction results in zombie load.
- No need to handle faults.

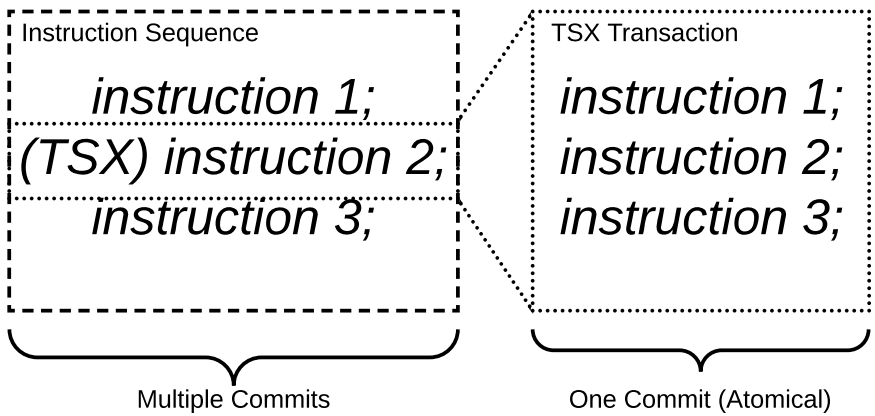

Fig. TSX transaction.

Attack Variants: V3 - Microcode Assisted Page Table Walk

- Two user virtual addresses v_1 and v_2 mapping to a same physical page.
- Set accessed bit of one to 0 then visit it in transient domain.
- Microcode assist performs page table walk to set the bit to 1.

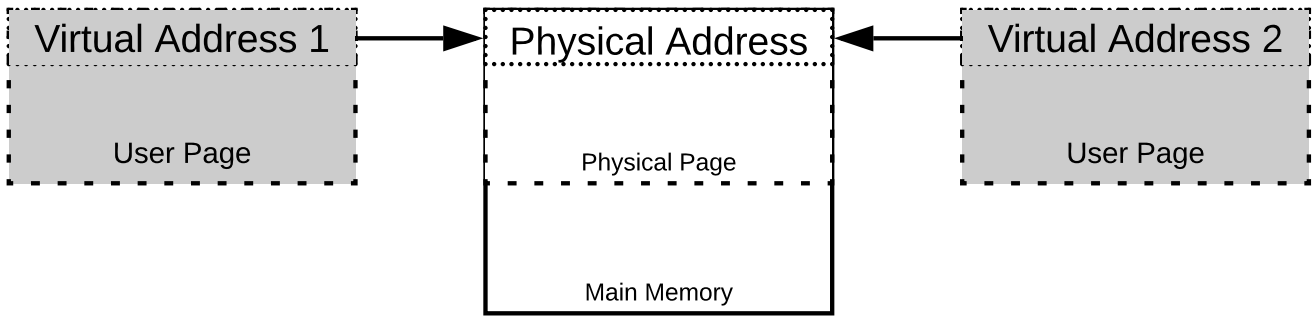

Fig. Two user shared pages.

Attack Variants: V4 and V5

V4 SGX abort page semantics: accessing virtual address which is reserved as SGX enclave triggers microcode assist.

V5 Uncacheable memory: similar to V4, but instead of SGX reserved memory uncacheable page is used.

Synchronization and Noise Filtering

- Get secret data in flight.
- Listen for specific code line to be cached.
- Use domino byte to connect detect sequential bytes.

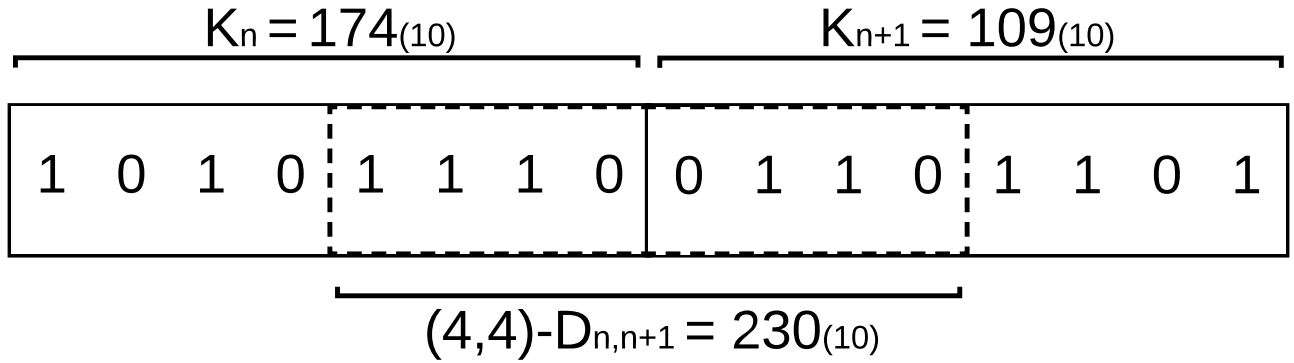

Fig. Domino Byte [ZombieLoad].

Results

- Leak AES-NI key.
- Extract SGX sealing key.
- Leak data across virtual machines.
- Monitor browsing behavior.

Version	Avg. Speed	Avg. Accuracy
V1	5.3 kB/s	85.74%
V2	39.66 kB/s	99,99%
V3	7.73 kB/s	76.28%

Environments

Setup	CPU	μ-arch	Variant 1	2	3
Lab	Core i7-33630QM	Ivy Bridge	Y	D	Y
Lab	Core i7-6700K	Skylake-S	Y	Y	Y
Lab	Core i5-7300U	Kaby Lake	Y	Y	Y
Lab	Core i7-7700	Kaby Lake	Y	Y	Y
Lab	Core i7-8650U	Kaby Lake-R	Y	Y	Y
Lab	Core i7-8565U	Whiskey Lake	N	D	N
Lab	Core i7-8700K	Coffee Lake-S	Y	Y	Y
Lab	Core i9-9900K	Coffee Lake-R	N	Y	N
Lab	Xeon E5-1630 v4	Broadwell-EP	Y	Y	Y
Cloud	Xeon E5-2670	Sandy Bridge-EP	Y	D	Y
Cloud	Xeon Gold 5120	Skylake-SP	Y	Y	Y
Cloud	Xeon Platinum 8175M	Skylake-SP	Y	D	Y
Cloud	Xeon Gold 5218	Cascade Lake-SP	N	Y	N

Y - Yes, N - No, D - Intel TSX is disabled

Fig. Tested Environments [ZombieLoad].

Countermeasures

- Disable hyperthreading with up to 30% performance.
- Co-scheduling: threads should enter kernel mode at the same time.
- Disable underlying instructions required to mount the attack.
- Flush L1 and buffers between context switches.
- Use combination of instructions $verw$ and $mfence$.
- Core pinning.

Conclusion

- Software mitigations are not enough, but switching hardware all the time is impossible.
- Chances that someone will attack your personal computer are extremely low.
- Keep your microcode updated.
- To prevent completely disable hyperthreading at cost of performance, or run untrusted code on separate physical core.
- Think about changing CPU manufacturer - Apple already did.

References

Meltdown Moritz Lipp, Michael Schwarz, Daniel Gruss, Thomas Prescher, Werner Haas, Anders Fogh, Jann Horn, Stefan Mangard, Paul Kocher, Daniel Genkin, YuvalYarom, and Mike Hamburg. 2018. Meltdown: Reading Kernel Memory from User Space. In27th USENIX Security Symposium (USENIX Security 18).

Spectre Paul Kocher, Jann Horn, Anders Fogh, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, Michael Schwarz, and Yuval Yarom. 2019. Spectre Attacks: Exploiting Speculative Execution. In 40th IEEE Symposium on Security and Privacy (S&P'19).

RIDL Stephan van Schaik, Alyssa Milburn, Sebastian Österlund, Pietro Frigo, GiorgiMaisuradze, Kaveh Razavi, Herbert Bos, and Cristiano Giuffrida. 2019. RIDL: Rogue In-flight Data Load. In S&P.

Fallout Claudio Canella, Daniel Genkin, Lukas Giner, Daniel Gruss, Moritz Lipp, Ma-rina Minkin, Daniel Moghimi, Frank Piessens, Michael Schwarz, Berk Sunar, JoVan Bulck, and Yuval Yarom. 2019. Fallout: Leaking Data on Meltdown-resistant CPUs. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS). ACM.

ZombieLoad Michael Schwarz, Moritz Lipp, Daniel Moghimi, Jo Van Bulck, Julian Steck-lina, Thomas Prescher, and Daniel Gruss. 2019. ZombieLoad: Cross-Privilege-Boundary Data Sampling. In CCS.

FlushReload Yuval Yarom and Katrina E. Falkner. 2013. FLUSH+RELOAD: A High Resolution, Low Noise, L3 Cache Side-Channel Attack. In USENIX Security Symposium.