

Public clouds and vulnerable CPUs: are we secure?

FOSDEM 2020

Vitaly Kuznetsov <vkuznets@redhat.com>

About myself

- Focusing (mostly) on Linux kernel
- My areas of interest include:
 - Linux as guest on public clouds (AWS, Azure, Aliyun,...)
 - Linux as guest on Hyper-V
 - Hyper-V Enlightenments in KVM
 - Running nested KVM on Hyper-V
 - Running nested Hyper-V on KVM

Speculative vulnerabilities discovered in the past few years:

- Spectre v1
 - SWAPGS
- Spectre v2
- Meltdown (Spectre v3)
- SSB (Spectre v4/NG)
- L1TF (AKA Foreshadow/Spectre v5)
- MDS & TAA

Speculative Execution Side Channel Methods

Intel:

“The concept behind speculative execution is that instructions are executed ahead of knowing that they are required. [...] By executing instructions speculatively, performance can be increased by minimizing latency and extracting greater parallelism.

....

While speculative operations do not affect the architectural state of the processor, they can affect the microarchitectural state, such as information stored in Translation Lookaside Buffers (TLBs) and caches.

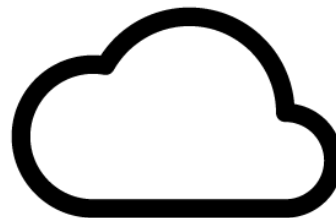
...

A side channel method works by gaining information through observing the system, such as by measuring microarchitectural properties about the system. Unlike buffer overflows and other vulnerability classes, side channels do not directly influence the execution of the program, nor do they allow data to be modified or deleted.

”

When running on a public cloud

... but my cloud provider tells me they've patched everything and I don't need to worry, is this so?



Types of attacks:

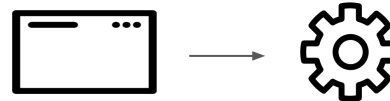
- VM to VM



- VM to Host (hypervisor)^[OBJ]





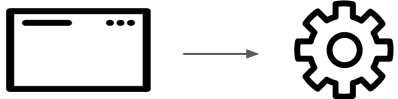

- in-VM: userspace to kernel



- in-VM: userspace to userspace

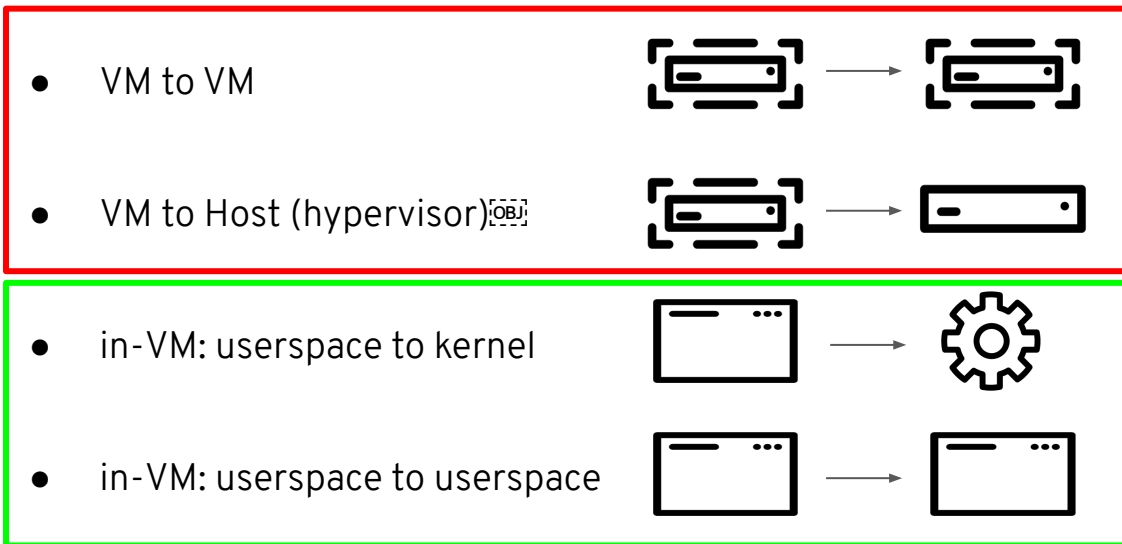


Types of attacks:

- VM to VM 
- VM to Host (hypervisor)^[OBJ] 
- in-VM: userspace to kernel 
- in-VM: userspace to userspace 

Of paramount importance to cloud providers!

Types of attacks:




Of paramount importance to cloud providers!

Outside of cloud providers' responsibility domain (but they need to provide the required tools for guests!)

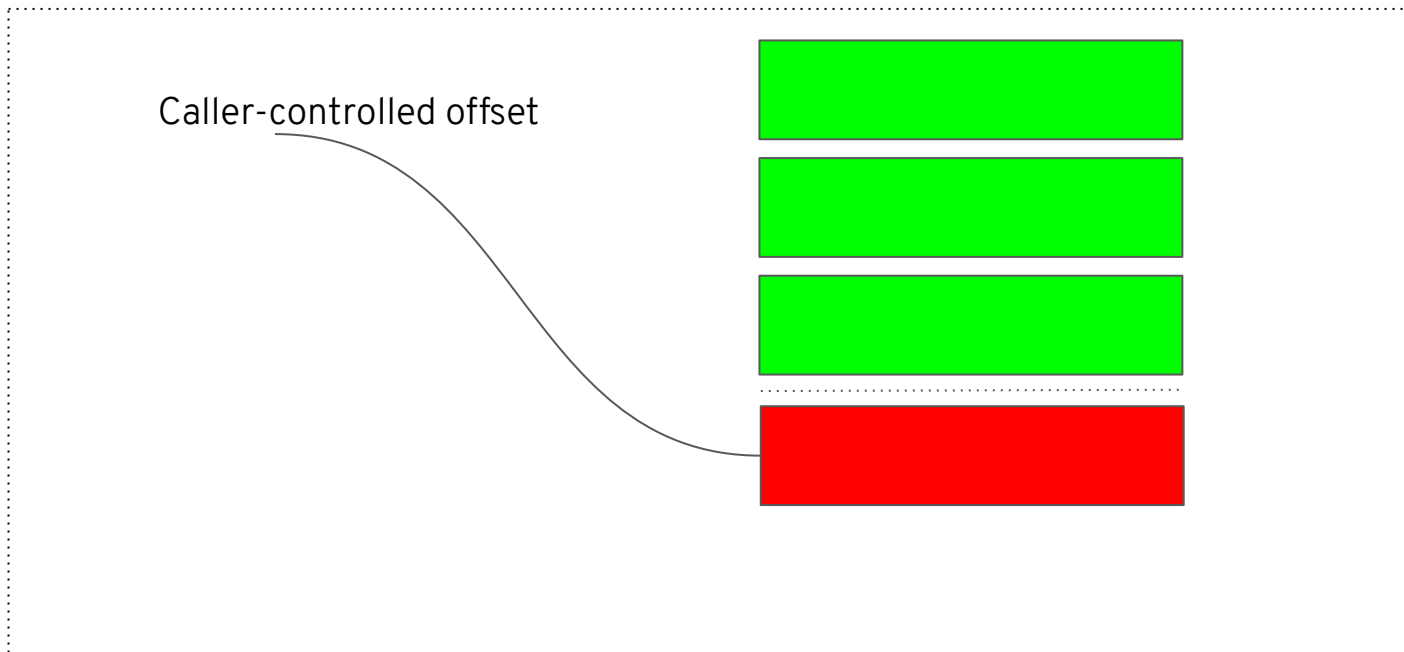
Things to consider for in-VM attacks

- Are you running a multi-tenant environment?
- Are you running (even in containers) untrusted code?
- Is this a ‘multi-task’ or a ‘single task’ VM?
- Are you relying on *language-based security* (e.g. running JITed untrusted code)?
- Is it acceptable to disable SMT (Hyperthreading)?
-



CPU vulnerabilities
caused by
“speculative
execution”

Spectre v1 (Bounds Check Bypass, CVE-2017-5753)



Spectre v1

- Hardware
 - No microcode update needed
- VM-to-VM and VM-to-hypervisor attacks:
 - Cloud provider fixes the hypervisor on a case by case basis (all potentially vulnerable places)

- In-VM attacks:

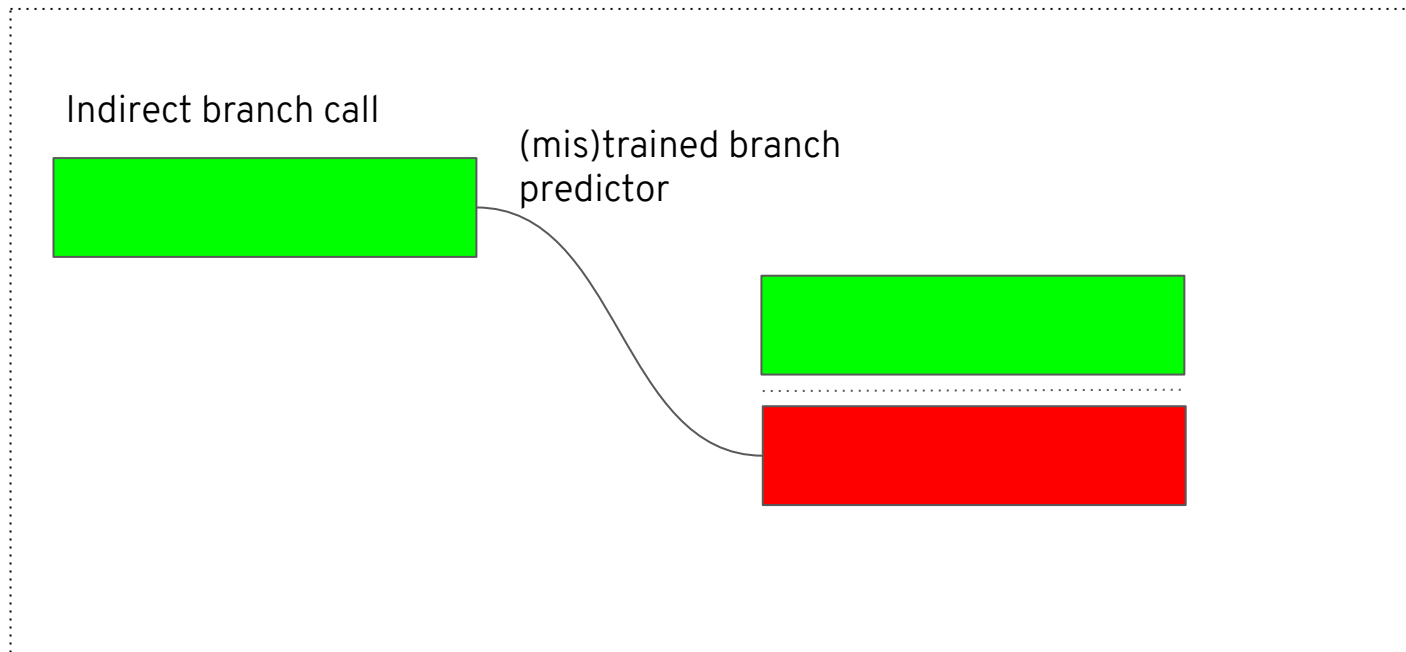
```
$ cat /sys/devices/system/cpu/vulnerabilities/spectre_v1  
Mitigation: usercopy/swapgs barriers and __user pointer sanitization
```

- Fixed in the kernel on a case by case basis. Keep yours updated!

SWAPGS (Spectre v1 variant, CVE-2019-1125)

- Speculation related to GS segment switch (per-CPU data)
- Software mitigation is needed for (almost) all current CPUs
- Future CPUs may fix the bug in hardware (NO_SWAPGS)

Spectre v2 (Branch Target Injection, CVE-2017-5715)



Spectre v2 (Branch Target Injection, CVE-2017-5715)

- Hardware support (microcode update) in combination with software techniques required for mitigations
 - Hardware:
 - **IBRS**: don't speculate in certain (privileged) contexts
 - **Enhanced IBRS**: tag branch target buffer (BTB)
 - **STIBP**: don't share BTB across CPU threads (for existing CPUs: just turn it off)
 - BTB may be shared across hyperthreads!
 - **IBPB**: clear BTB when switching between tasks
 - Software: **retpoline, RSB filling**

Spectre v2: hypervisor protection

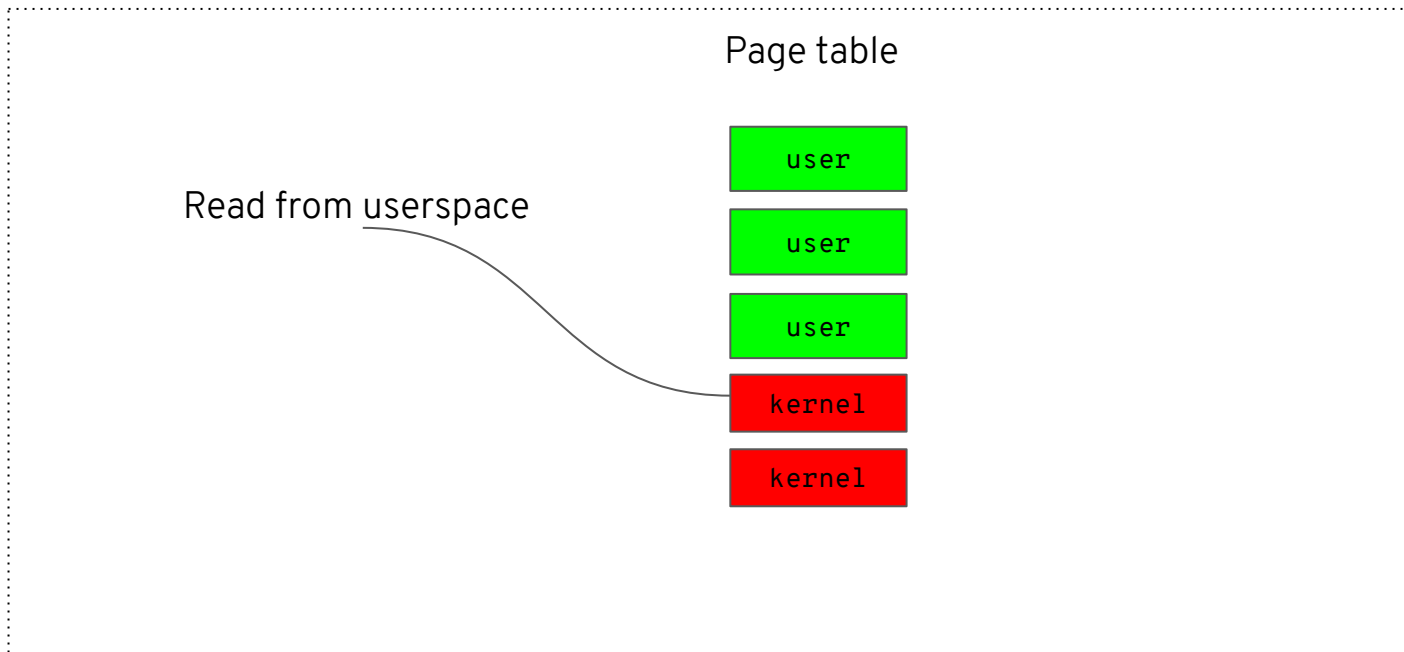
- VM-to-VM attacks:
 - Not possible with fully dedicated cores (most instance types)
 - SMT: core scheduling for VMs
- VM-to-hypervisor attacks:
 - Hypervisor itself needs to be protected (**retpoline** or **IBRS**)
 - SMT: protect with hardware features (**IBRS/STIBP**) or parallel threads should be blocked while in hypervisor context

Spectre v2: guest protection

```
$ cat /sys/devices/system/cpu/vulnerabilities/spectre_v2  
Mitigation: Full generic retpoline, IBPB: conditional, IBRS_FW, STIBP:  
conditional, RSB filling
```

- Userspace-to-kernel attacks:
 - **Enhanced IBRS (IBRS_ALL)** or **retpoline**
- Userspace-to-userspace attacks:
 - ***spectre_v2_user=on/off/prctl/prctl,ibpb/seccomp/seccomp,ibpb/auto*** depending on your needs!
 - Hardware features: **STIBP** and **IBPB** need to be exposed to the guest (check your */proc/cpuinfo*)
 - “***nosmt***” can be used instead of **STIBP** (may give a better performance in some cases)

Meltdown (Rogue data cache load, CVE-2017-5754)

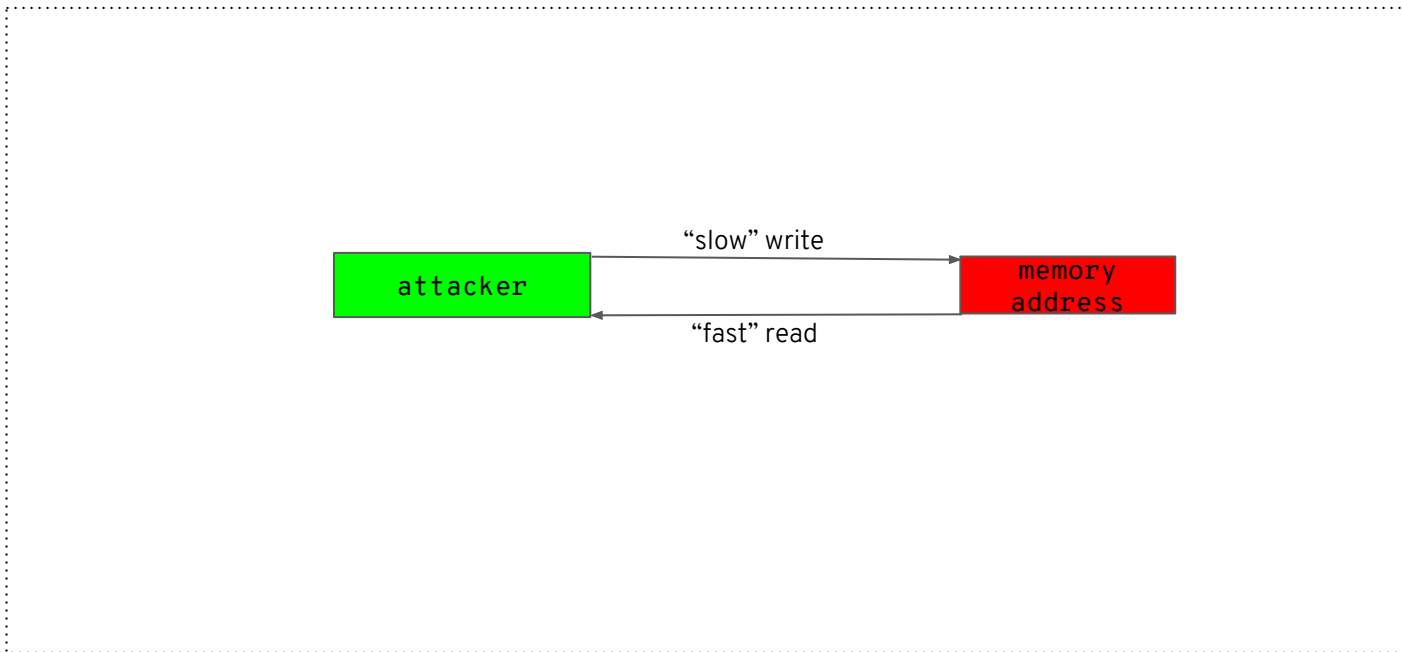


Meltdown

- Hardware
 - No microcode update needed for mitigation
 - Future CPUs may get fixed (**RDCL_NO**)
- VM-to-VM and VM-to-hypervisor attacks:
 - Only Xen PV is (was) vulnerable
- In-VM attacks:

```
$ cat /sys/devices/system/cpu/vulnerabilities/meltdown  
Mitigation: PTI
```

Speculative Store Bypass (SSB, CVE-2018-3639)



Speculative Store Bypass: hypervisor protection

- Hardware:
 - Microcode update required for mitigations (**SSBD**, **VIRT_SSBD** - AMD only)
 - SMT: SSB may be per-core!
 - Future CPUs may get fixed (**NO_SSB**)
- VM-to-VM and VM-to-hypervisor attacks:
 - Don't seem to be possible

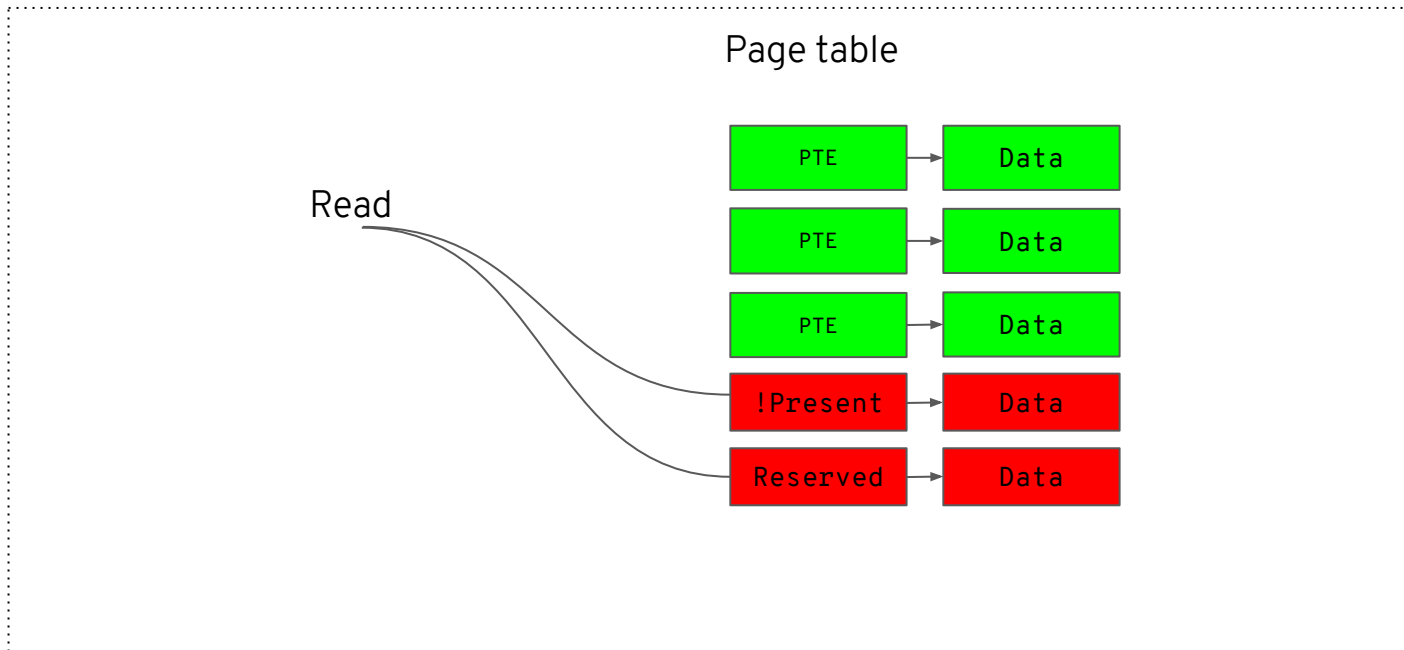
Speculative Store Bypass: guest protection

- In-VM-attacks:
 - “*Language-based security environments*” (e.g. JIT) are at highest risk, e.g. javaws executing untrusted code.
 - ‘**ssbd**’ cpu feature required for mitigation

```
$ cat /sys/devices/system/cpu/vulnerabilities/spec_store_bypass  
Mitigation: Speculative Store Bypass disabled via prctl and seccomp
```

- ***ssbd=force-on/force-off/kernel***

L1 Terminal Fault (L1TF, Foreshadow, CVE-2018-3615, CVE-2018-3620, CVE-2018-3646)



L1 Terminal Fault: hypervisor protection

- Hardware
 - Microcode update required for more effective mitigation on hypervisors (“**flush_l1d**”)
 - Future CPUs may get fixed (**RDCL_NO**)
- VM-to-VM attacks:
 - Not possible with dedicated cores (most instance types)
 - Core scheduling + L1D flush should be utilized when cores are shared (L1D is per core)
- VM-to-hypervisor attacks:
 - Core scheduling/simultaneous exit + L1D flush

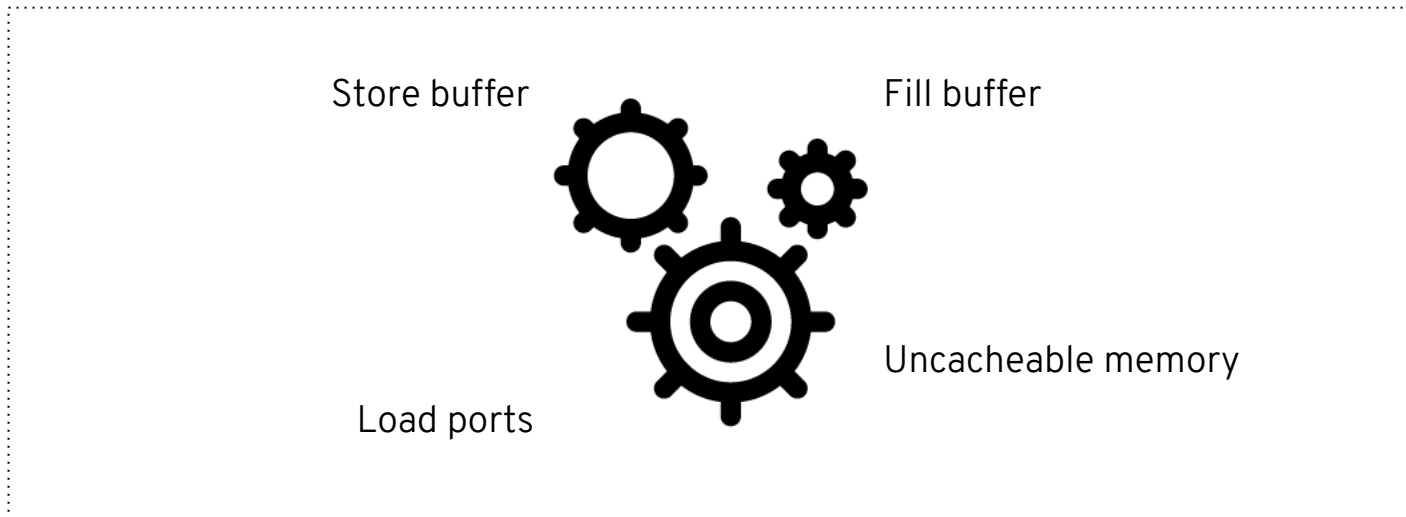
L1 Terminal Fault: guest protection

- In-VM attacks:

```
$ cat /sys/devices/system/cpu/vulnerabilities/l1tf  
Mitigation: PTE Inversion; VMX: conditional cache flushes, SMT  
vulnerable
```

- Software-based mitigation (PTE inversion) against userspace-to-kernel attacks, always enabled
- **l1tf=full/full,force/flush/flush,nosmt/flush,nowarn/off** only applies if you're running VMs (on public clouds: likely nested)

Microarchitectural Data Sampling (MDS, CVE-2018-12126, CVE-2018-12127, CVE-2018-12130, CVE-2019-11091) and TSX Asynchronous Abort (CVE-2019-11135)



MDS & TAA

- Hardware
 - Future hardware is expected to get fixed:
 - MFBDS: **RDCL_NO**
 - MFBDS/MSBDS/MLPDS/MDSUM: **MDS_NO**
 - TAA: **TAA_NO**
 - Existing hardware: microcode update required for mitigating (check for “**md_clear**”)
- VM-to-VM attacks:
 - Not possible with dedicated cores (most instance types)
 - Core scheduling + MD_CLEAR should be utilized when cores are shared
- VM-to-hypervisor attacks:
 - Core scheduling/simultaneous exit + MD_CLEAR

MDS & TAA

- In-VM attacks:

```
$ cat /sys/devices/system/cpu/vulnerabilities/mds  
Mitigation: Clear CPU buffers; SMT vulnerable
```

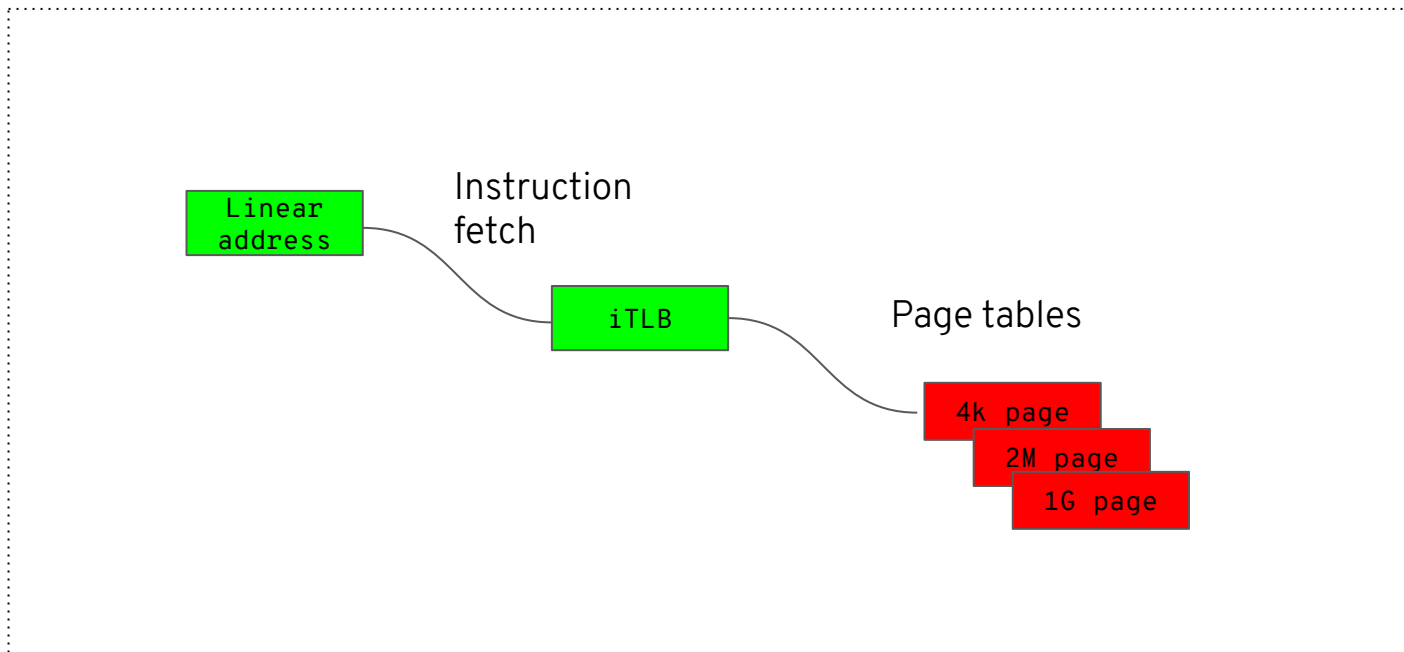
```
$ cat /sys/devices/system/cpu/vulnerabilities/tsx_async_abort  
Mitigation: Clear CPU buffers; SMT vulnerable
```

- ‘**md_clear**’ feature needed for effective mitigation
 - State is unknown if absent, Linux still tries
- SMT: no protection
 - Manual CPU pinning or “core scheduler” (not yet upstream) may help to certain extent (*userspace-to-userspace but not userspace-to-kernel*)
 - Use “**nosmt**” for ultimate protection



Other CPU vulnerabilities

ITLB_MULTIHIT (MCE on Page Size change, CVE-2018-12207)



ITLB_MULTIHIT

- Hardware
 - No microcode update needed
 - Future hardware is expected to get fixed (**PSCHANGE_MC_NO**)
- VM-to-VM and VM-to-hypervisor attacks:
 - Malicious guest can cause DoS
 - Cloud provider fixes the hypervisor by disabling huge pages (EPT) or by making them non-executable
- In-VM attacks:
 - Not possible, userspace can not trigger page size change
 - Nested hypervisors don't need additional mitigations.



Examples

AWS

- Instance type: **r5n.large**, CPU: Intel(R) Xeon(R) Platinum 8259CL CPU @ 2.50GHz
- /sys/devices/system/cpu/vulnerabilities:

```
itlb_multihit: KVM: Vulnerable
l1tf: Mitigation: PTE Inversion
mds: Vulnerable: Clear CPU buffers attempted, no microcode; SMT Host state
    unknown
meltdown: Mitigation: PTI
spec_store_bypass: Vulnerable
spectre_v1: Mitigation: usercopy/swapgs barriers and __user pointer
    sanitization
spectre_v2: Mitigation: Full generic retpoline, STIBP: disabled, RSB
    filling
tsx_async_abort: Not affected
```

AWS

- Instance type: **r5n.large**, CPU: Intel(R) Xeon(R) Platinum 8259CL CPU @ 2.50GHz
- /sys/devices/system/cpu/vulnerabilities:

itlb_multihit: KVM: Vulnerable	<- Irrelevant (no support for KVM)
lltf: Mitigation: PTE Inversion	<- Mitigated
mds: Vulnerable: Clear CPU buffers attempted, no microcode; SMT Host state unknown	<- Unknown, presumably mitigated
meltdown: Mitigation: PTI	<- Mitigated
spec_store_bypass: Vulnerable	<- No mitigation available!
spectre_v1: Mitigation: usercopy/swapgs barriers and __user pointer sanitization	<- Mitigated
spectre_v2: Mitigation: Full generic retpoline, STIBP: disabled, RSB filling	<- No mitigation (for userspace) available!
tsx_async_abort: Not affected	<- CPU not affected (features turned off)

Azure

- Instance type: **F8s_v2**, CPU: Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz
- `/sys/devices/system/cpu/vulnerabilities:`

```
itlb_multihit: KVM: Mitigation: Split huge pages
l1tf: Mitigation: PTE Inversion; VMX: conditional cache flushes, SMT
vulnerable
mds: Vulnerable: Clear CPU buffers attempted, no microcode; SMT Host state
unknown
meltdown: Mitigation: PTI
spec_store_bypass: Vulnerable
spectre_v1: Mitigation: usercopy/swaps barriers and __user pointer
sanitization
spectre_v2: Mitigation: Full generic retpoline, STIBP: disabled, RSB
filling
tsx_async_abort: Vulnerable: Clear CPU buffers attempted, no microcode; SMT
Host state unknown
```

Azure

- Instance type: **F8s_v2**, CPU: Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz
- /sys/devices/system/cpu/vulnerabilities:

```
itlb_multihit: KVM: Mitigation: Split huge pages <- Superfluous mitigation
lltf: Mitigation: PTE Inversion; VMX: conditional cache flushes, SMT
vulnerable <- Mitigated (but SMT still vulnerable)
mds: Vulnerable: Clear CPU buffers attempted, no microcode; SMT Host state
unknown <- Unknown, presumably mitigated
meltdown: Mitigation: PTI <- Mitigated
spec_store_bypass: Vulnerable <- No mitigation available!
spectre_v1: Mitigation: usercopy/swaps barriers and __user pointer
sanitization <- Mitigated
spectre_v2: Mitigation: Full generic retpoline, STIBP: disabled, RSB
filling <- No mitigation (for userspace) available!
tsx_async_abort: Vulnerable: Clear CPU buffers attempted, no microcode; SMT
Host state unknown <- Unknown, presumably mitigated
```

GCE

- Instance type: N2, CPU: Intel(R) Xeon(R) Cascade Lake @ 2.80GHz
- /sys/devices/system/cpu/vulnerabilities:

```
itlb_multihit: KVM: Vulnerable
l1tf: Not affected
mds: Mitigation: Clear CPU buffers; SMT Host state unknown
meltdown: Not affected
spec_store_bypass: Mitigation: Speculative Store Bypass disabled via prctl
and seccomp
spectre_v1: Mitigation: usercopy/swapgs barriers and __user pointer
sanitization
spectre_v2: Mitigation: Enhanced IBRS,
IBPB: conditional, RSB filling
tsx_async_abort: Mitigation: Clear CPU buffers; SMT Host state unknown
```

GCE

- Instance type: N2, CPU: Intel(R) Xeon(R) Cascade Lake @ 2.80GHz
- /sys/devices/system/cpu/vulnerabilities:

itlb_multihit: KVM: Vulnerable	<- Most likely mitigated on the host
lltf: Not affected	<- Unaffected
mds: Mitigation: Clear CPU buffers; SMT Host state unknown	<- Mitigated
meltdown: Not affected	<- Unaffected
spec_store_bypass: Mitigation: Speculative Store Bypass disabled via prctl and seccomp	<- Mitigation available
spectre_v1: Mitigation: usercopy/swapgs barriers and __user pointer sanitization	<- Mitigated
spectre_v2: Mitigation: Enhanced IBRS, IBPB: conditional, RSB filling	<- Mitigation available
tsx_async_abort: Mitigation: Clear CPU buffers; SMT Host state unknown	<- Mitigated

Thank you!



[linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)



[youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)



[facebook.com/redhatinc](https://www.facebook.com/redhatinc)



twitter.com/RedHat