SRBDS - Special Register Buffer Data Sampling

SRBDS is a hardware vulnerability that allows MDS Documentation/admin-guide/hw-vuln/mds.rst techniques to infer values returned from special register accesses. Special register accesses are accesses to off core registers. According to Intel's evaluation, the special register reads that have a security expectation of privacy are RDRAND, RDSEED and SGX EGETKEY.

When RDRAND, RDSEED and EGETKEY instructions are used, the data is moved to the core through the special register mechanism that is susceptible to MDS attacks.

Affected processors

Core models (desktop, mobile, Xeon-E3) that implement RDRAND and/or RDSEED may be affected.

A processor is affected by SRBDS if its Family_Model and stepping is in the following list, with the exception of the listed processors exporting MDS_NO while Intel TSX is available yet not enabled. The latter class of processors are only affected when Intel TSX is enabled by software using TSX_CTRL_MSR otherwise they are not affected.

common name	Family_Model	Stepping
IvyBridge	06_3AH	All
Haswell	06_3CH	All
Haswell_L	06_45H	All
Haswell_G	06_46H	All
Broadwell_G	06_47H	All
Broadwell	06_3DH	All
Skylake_L	06_4EH	All
Skylake	06_5EH	All
Kabylake_L	06_8EH	<= 0xC
Kabylake	06_9EH	<= 0xD

Related CVEs

The following CVE entry is related to this SRBDS issue:

CVE-2020-0543	SRBDS	Special Register Buffer Data Sampling

Attack scenarios

An unprivileged user can extract values returned from RDRAND and RDSEED executed on another core or sibling thread using MDS techniques.

Mitigation mechanism

Intel will release microcode updates that modify the RDRAND, RDSEED, and EGETKEY instructions to overwrite secret special register data in the shared staging buffer before the secret data can be accessed by another logical processor.

During execution of the RDRAND, RDSEED, or EGETKEY instructions, off-core accesses from other logical processors will be delayed until the special register read is complete and the secret data in the shared staging buffer is overwritten.

This has three effects on performance:

1. RDRAND, RDSEED, or EGETKEY instructions have higher latency.
2. Executing RDRAND at the same time on multiple logical processors will be serialized, resulting in an overall reduction in the maximum RDRAND bandwidth.
3. Executing RDRAND, RDSEED or EGETKEY will delay memory accesses from other logical processors that miss their core caches, with an impact similar to legacy locked cache-line-split accesses.

The microcode updates provide an opt-out mechanism (RNGDS_MITG_DIS) to disable the mitigation for RDRAND and RDSEED instructions executed outside of Intel Software Guard Extensions (Intel SGX) enclaves. On logical processors that disable the mitigation using this opt-out mechanism, RDRAND and RDSEED do not take longer to execute and do not impact performance of sibling logical processors memory accesses. The opt-out mechanism does not affect Intel SGX enclaves (including execution of RDRAND or RDSEED inside an enclave, as well as EGETKEY execution).

IA32_MCU_OPT_CTRL MSR Definition

Along with the mitigation for this issue, Intel added a new thread-scope IA32_MCU_OPT_CTRL MSR, (address 0x123). The presence of this MSR and RNGDS_MITG_DIS (bit 0) is enumerated by CPUID.(EAX=07H,ECX=0).EDX[SRBDS_CTRL =

9]==1. This MSR is introduced through the microcode update.

Setting IA32_MCU_OPT_CTRL[0] (RNGDS_MITG_DIS) to 1 for a logical processor disables the mitigation for RDRAND and RDSEED executed outside of an Intel SGX enclave on that logical processor. Opting out of the mitigation for a particular logical processor does not affect the RDRAND and RDSEED mitigations for other logical processors.

Note that inside of an Intel SGX enclave, the mitigation is applied regardless of the value of RNGDS_MITG_DS.

Mitigation control on the kernel command line

The kernel command line allows control over the SRBDS mitigation at boot time with the option "srbds=". The option for this is:

off	This option disables SRBDS mitigation for RDRAND and RDSEED on affected platforms.

SRBDS System Information

The Linux kernel provides vulnerability status information through sysfs. For SRBDS this can be accessed by the following sysfs file: /sys/devices/system/cpu/vulnerabilities/srbds

The possible values contained in this file are:

Not affected	Processor not vulnerable
Vulnerable	Processor vulnerable and mitigation disabled
Vulnerable: No microcode	Processor vulnerable and microcode is missing mitigation
Mitigation: Microcode	Processor is vulnerable and mitigation is in effect.
Mitigation: TSX disabled	Processor is only vulnerable when TSX is enabled while this system was booted with TSX disabled.
Unknown: Dependent on	
hypervisor status	Running on virtual guest processor that is affected but with no way to know if host processor is mitigated or vulnerable.

SRBDS Default mitigation

This new microcode serializes processor access during execution of RDRAND, RDSEED ensures that the shared buffer is overwritten before it is released for reuse. Use the "srbds=off" kernel command line to disable the mitigation for RDRAND and RDSEED.