AMD Memory Encryption

Secure Memory Encryption (SME) and Secure Encrypted Virtualization (SEV) are features found on AMD processors.

SME provides the ability to mark individual pages of memory as encrypted using the standard x86 page tables. A page that is marked encrypted will be automatically decrypted when read from DRAM and encrypted when written to DRAM. SME can therefore be used to protect the contents of DRAM from physical attacks on the system.

SEV enables running encrypted virtual machines (VMs) in which the code and data of the guest VM are secured so that a decrypted version is available only within the VM itself. SEV guest VMs have the concept of private and shared memory. Private memory is encrypted with the guest-specific key, while shared memory may be encrypted with hypervisor key. When SME is enabled, the hypervisor key is the same key which is used in SME.

A page is encrypted when a page table entry has the encryption bit set (see below on how to determine its position). The encryption bit can also be specified in the cr3 register, allowing the PGD table to be encrypted. Each successive level of page tables can also be encrypted by setting the encryption bit in the page table entry that points to the next table. This allows the full page table hierarchy to be encrypted. Note, this means that just because the encryption bit is set in cr3, doesn't imply the full hierarchy is encrypted. Each page table entry in the hierarchy needs to have the encryption bit set to achieve that. So, theoretically, you could have the encryption bit set in cr3 so that the PGD is encrypted, but not set the encryption bit in the PGD entry for a PUD which results in the PUD pointed to by that entry to not be encrypted.

When SEV is enabled, instruction pages and guest page tables are always treated as private. All the DMA operations inside the guest must be performed on shared memory. Since the memory encryption bit is controlled by the guest OS when it is operating in 64-bit or 32-bit PAE mode, in all other modes the SEV hardware forces the memory encryption bit to 1.

Support for SME and SEV can be determined through the CPUID instruction. The CPUID function 0x8000001f reports information related to SME:

```
0x8000001f[eax]:
        Bit[0] indicates support for SME
        Bit[1] indicates support for SEV
0x8000001f[ebx]:
        Bits[5:0]  pagetable bit number used to activate memory
                   encryption
        Bits[11:6] reduction in physical address space, in bits, when
                   memory encryption is enabled (this only affects
                   system physical addresses, not guest physical
                   addresses)
```

If support for SME is present, MSR 0xc00100010 (MSR_AMD64_SYSCFG) can be used to determine if SME is enabled and/or to enable memory encryption:

```
0xc0010010:
        Bit[23]   0 = memory encryption features are disabled
                  1 = memory encryption features are enabled
```

If SEV is supported, MSR 0xc0010131 (MSR_AMD64_SEV) can be used to determine if SEV is active:

```
0xc0010131:
        Bit[0]    0 = memory encryption is not active
                  1 = memory encryption is active
```

Linux relies on BIOS to set this bit if BIOS has determined that the reduction in the physical address space as a result of enabling memory encryption (see CPUID information above) will not conflict with the address space resource requirements for the system. If this bit is not set upon Linux startup then Linux itself will not set it and memory encryption will not be possible.

The state of SME in the Linux kernel can be documented as follows:

- Supported: The CPU supports SME (determined through CPUID instruction).
- Enabled: Supported and bit 23 of MSR_AMD64_SYSCFG is set.
- Active: Supported, Enabled and the Linux kernel is actively applying the encryption bit to page table entries (the SME mask in the kernel is non-zero).

SME can also be enabled and activated in the BIOS. If SME is enabled and activated in the BIOS, then all memory accesses will be encrypted and it will not be necessary to activate the Linux memory encryption support. If the BIOS merely enables SME (sets bit 23 of the MSR_AMD64_SYSCFG), then Linux can activate memory encryption by default (CONFIG_AMD_MEM_ENCRYPT_ACTIVE_BY_DEFAULT=y) or by supplying mem_encrypt=on on the kernel command line. However, if BIOS does not enable SME, then Linux will not be able to activate memory encryption, even if configured to do so by default or the mem_encrypt=on command line parameter is specified.