AMD EPYC

USER GUIDE
AMD EPYC
9004, 7003, 7002, 7001

Using SEV with AMD EPYC™ Processors

Publication	58207
Revision	1.0
Issue Date	Mar, 2023

Date	Version	Changes
Mar, 2023	1.0	Initial release

Audience

This tuning guide is intended for a technical audience such as production deployment, virtualization developers, firmware engineers, and performance engineering teams with:

- A background in configuring servers.
- Access to the system BIOS.

Author

Brent Hollingsworth, Diego Gonzalez Villalobos, Anthony Hernandez

Table of Contents

Chapter 1 Security Features by Processor Generation

AMD EPYC processors have the following security features by generation:

1.1 4th Gen (9xx4)

- Secure Encrypted Virtualization (SEV)
- Secure Encrypted Virtualization – Encrypted State (SEV-ES)
- Secure Nested Paging (SEV-SNP)
- 1006 ASID keys
- Transparent Secure Memory Encryption (TSME)

1.2 3rd Gen (7xx3)

- Secure Encrypted Virtualization (SEV)
- Secure Encrypted Virtualization-Encrypted State (SEV-ES)
- Secure Nested Paging (SEV-SNP)
- Either:
  - 509 ASID keys (in systems equipped with up to 8TB DRAM)
  - 253 ASID keys (in systems equipped with up to 16TB DRAM)
- TSME

1.3 2nd Gen (7xx2)

- Secure Encrypted Virtualization (SEV)
- Secure Encrypted Virtualization-Encrypted State (SEV-ES)
- Either:
  - 509 ASID keys (in systems equipped with up to 8TB DRAM)
  - 253 ASID keys (in systems equipped with up to 16TB DRAM)
- TSME

1.4 1st Gen (7xx1)

- Secure Encrypted Virtualization (SEV)
- Secure Encrypted Virtualization-Encrypted State (SEV-ES)
- 15 ASID keys
- TSME

Chapter 2 Enabling/Disabling SMEE

This chapter describes how to enable the AMD Secure Memory Encryption (SMEE) feature. SMEE must be enabled in order to use all SEV features. All of the instructions shown in this chapter are based on AMD Custom Reference Boards (CRBs). The exact steps and images may vary by OEM and BIOS version..

2.1 Enabling SMEE in BIOS

This section describes to enable SMEE on AMD EPYC processors.

2.1.1 AMD EPYC 9004 Series Processors

SMEE is disabled by default on systems powered by AMD EPYC 9004 Series Processors because of incompatibility with certain Linux kernels. To enable SMEE:

1. Access your system BIOS.

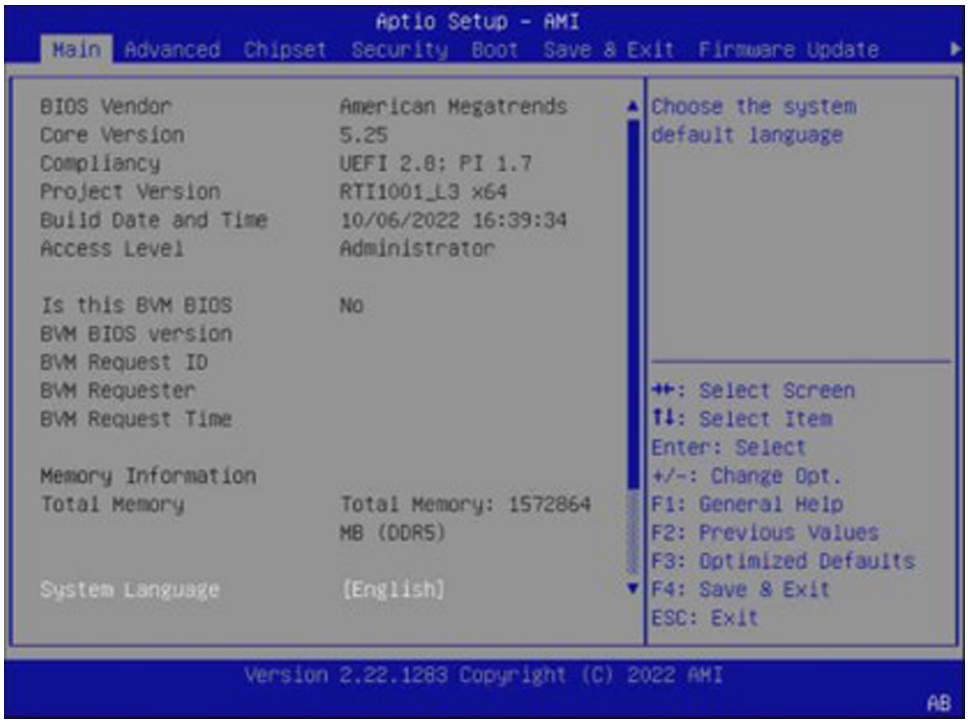


Figure 2-1: System BIOS (AMD EPYC 9004 Series Processors)

2. Select the **Advanced** tab.

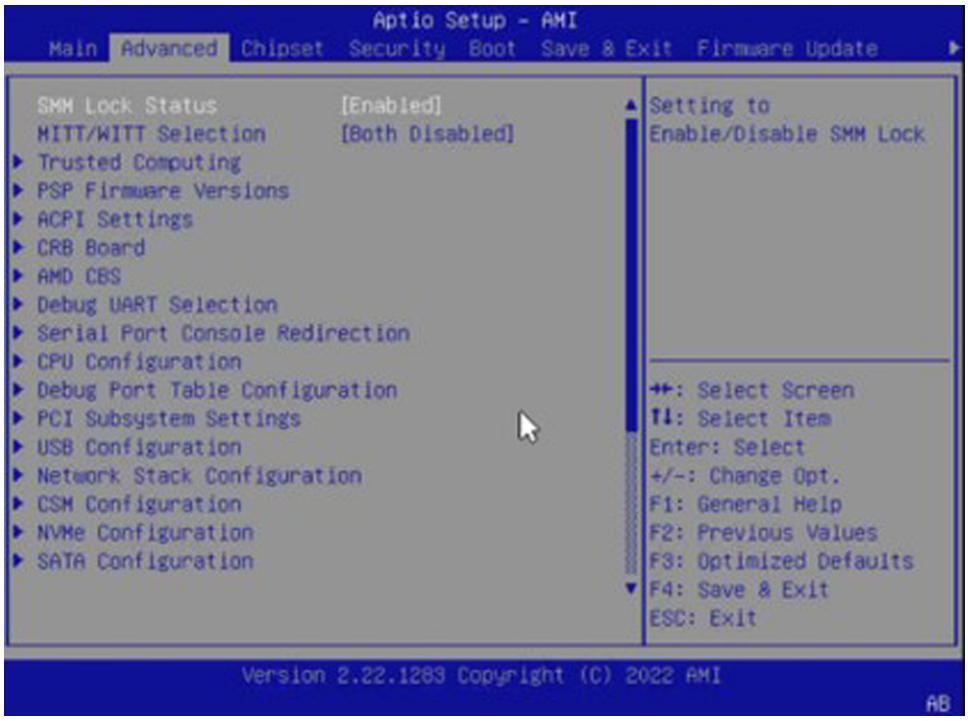


Figure 2-2: BIOS Advanced tab (AMD EPYC 9004 Series Processors)

3. Select **AMD CBS**.

Figure 2-3: AMD CBS tab (AMD EPYC 9004 Series Processors)

4. Select **CPU Common Options**.

Figure 2-4: CPU Common Options tab (AMD EPYC 9004 Series Processors)

5. Scroll down this tab, then select **SMEE**, and then set it to **Enable**.

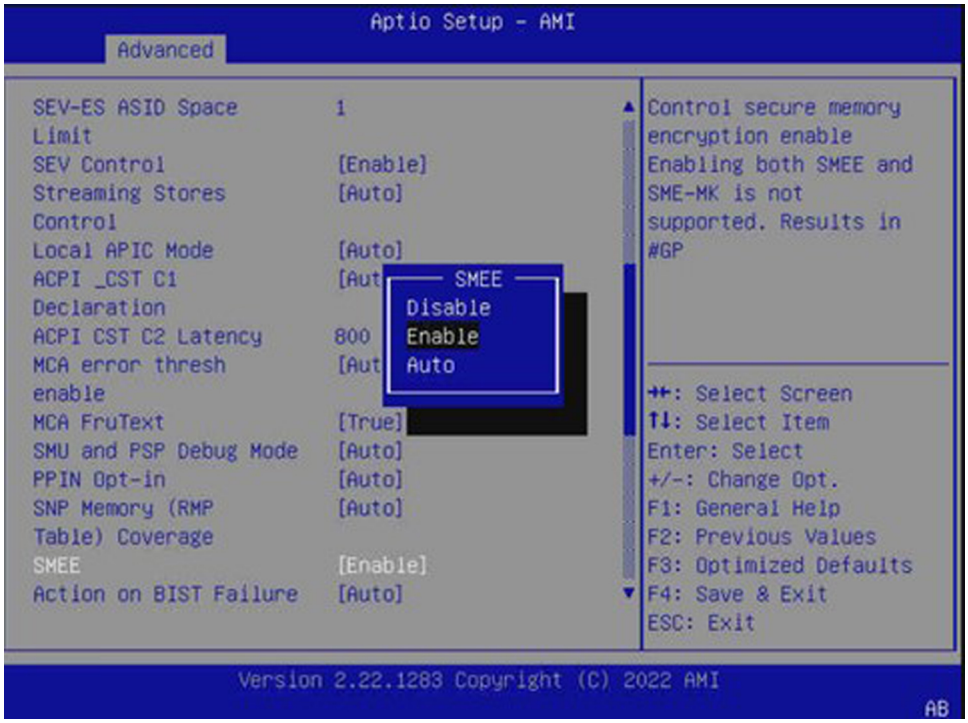


Figure 2-5: SMEE enabled (AMD EPYC 9004 Series Processors)

2.1.2 AMD EPYC 7003 Series Processors

SMEE is disabled by default on systems powered by AMD EPYC 7003 Series Processors because of incompatibility with certain Linux kernels. To enable SMEE:

1. Access your system BIOS.

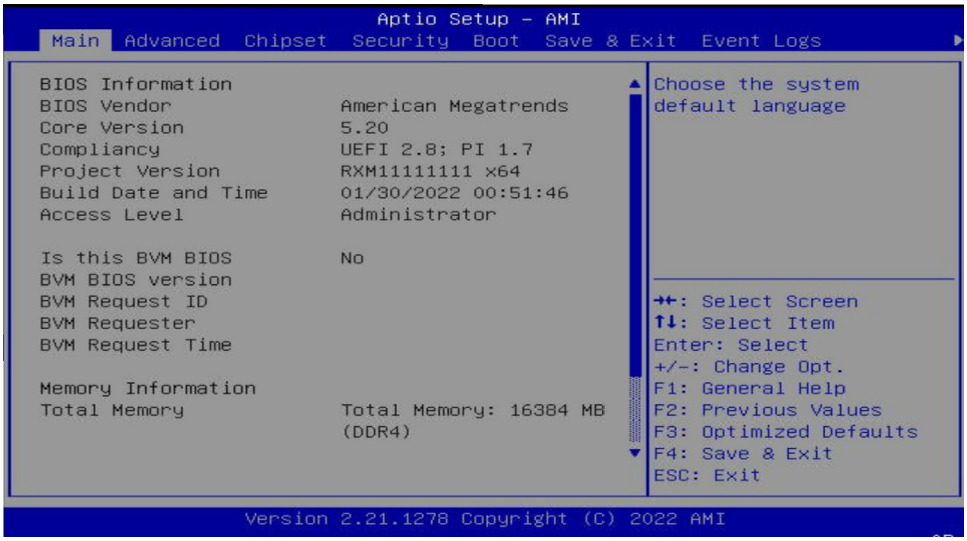


Figure 2-6: System BIOS (AMD EPYC 7003 Series Processors)

2. Select the **Advanced** tab.

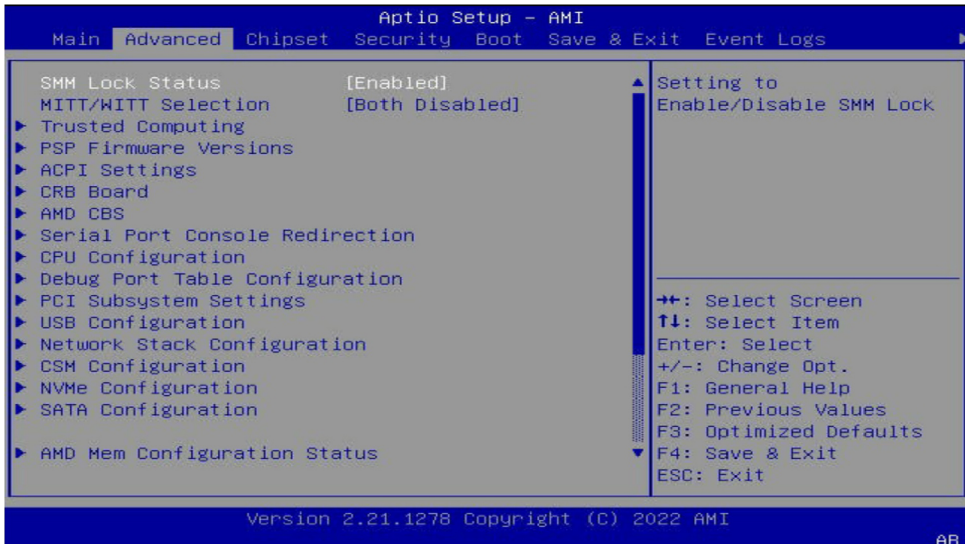


Figure 2-7: BIOS Advanced tab (AMD EPYC 7003 Series Processors)

3. Select **AMD CBS**.

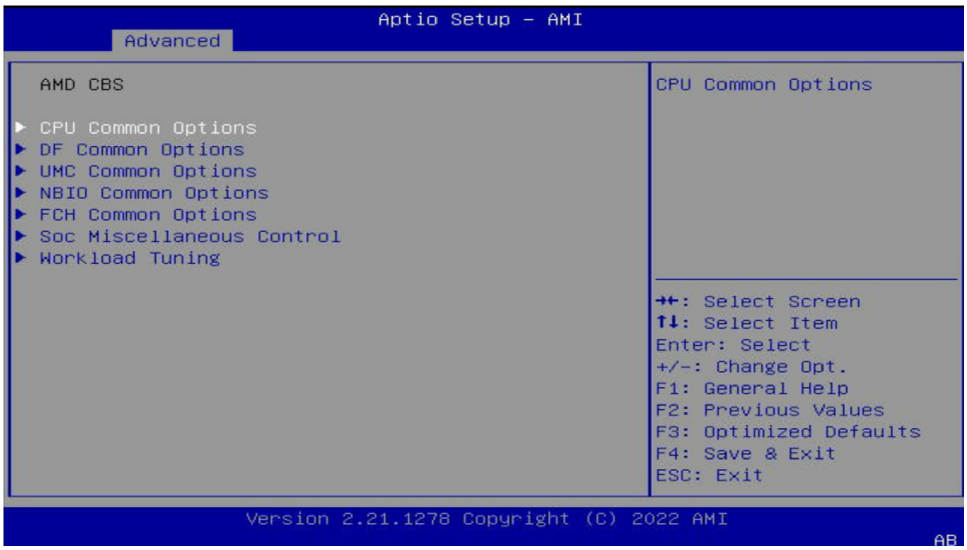


Figure 2-8: AMD CBS tab (AMD EPYC 7003 Series Processors)

4. Select **CPU Common Options**.

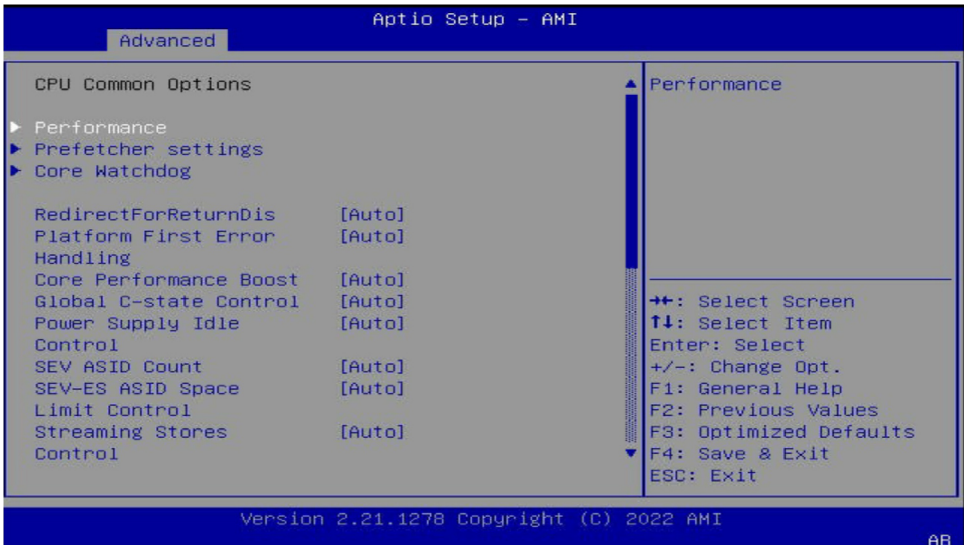


Figure 2-9: CPU Common Options tab (AMD EPYC 7003 Series Processors)

5. Scroll down this tab, then select **SMEE**, and then set it to **Enable**.

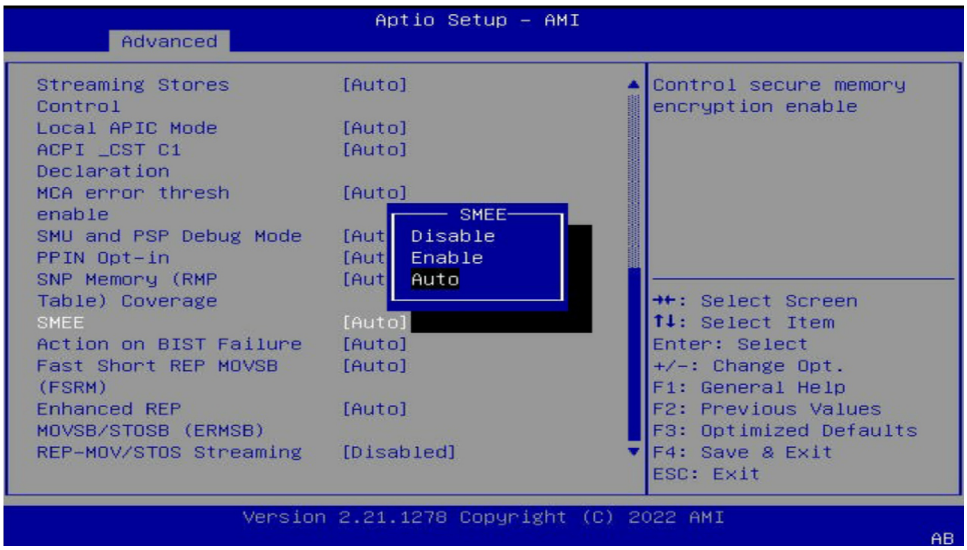


Figure 2-10: SMEE enabled (AMD EPYC 7003 Series Processors)

2.1.3 AMD EPYC 7002 and 7001 Series Processors

SMEE is **Enabled** by default on system powered by AMD EPYC 7002 or 7001 Series Processors.

2.2 Enabling SMEE via SMR

To enable SMEE via the processor MSR:

- x86 can set the SMEE bit (bit 23) in the `SYS_CFG MSR` before OS boot.
- `MSRC001_0010 [System Configuration] (Core::X86::Msr::SYS_CFG)`
- EDK2-based BIOS (non-CBS users) should specifically toggle this bit to enable/disable SEV if a reciprocal PCD method is not available for that processor family.

Note: This bit must be set on every CPU in the system.

Note: The bit is Write-1-Only, which (cannot be cleared once set, and which is set to 0 on system reset.

Note: AMD EPYC 7001 and 7002 Series Processors have SMEE enabled automatically. If SMEE is disabled in BIOS, then you can use MSR to reenable SMEE in the system.

2.3 Disabling SMEE in BIOS

This section describes disabling SMEE on AMD EPYC processors.

2.3.1 AMD EPYC 9004 Series Processors

To disable SMEE on a system with an AMD EPYC 9004 Series Processor:

1. Access your system BIOS.
2. Select the **Advanced** tab.
3. Select **AMD CBS**.
4. Select **CPU Common Options**.
5. Scroll down this tab, then select **SMEE**, and then set it to either **Auto** or **Disabled**.

2.3.2 AMD EPYC 7003 Series Processors

To disable SMEE on a system with an AMD EPYC 7003 Series Processor:

1. Access your system BIOS.
2. Select the **Advanced** tab.
3. Select **AMD CBS**.
4. Select **CPU Common Options**.
5. Scroll down this tab, then select **SMEE**, and then set it to either **Auto** or **Disabled**.

2.3.3 AMD EPYC 7002 and 7001 Series Processors

You cannot disable SMEE on a system with an AMD EPYC 7002 or 7001 Series Processor.

2.4 Disabling SMEE via MSR

SMEE cannot be disabled in the MSR; the bit is Write-1-Only. You must either reset the system or disable SMEE in BIOS.

Note: Disabling SEV will allow the use of more than 16TB of system physical address space (DRAM + PCIe + MMIO, etc.) because x bits of physical address space will not be used for ASIDs/c-bit.

- **AMD EPYC 9004 Series Processors:** 52-bit addressing with no c-bit, SMEE/SEV off.
  - 46-bit address with SEV (1006 keys).
- **AMD EPYC 7003 Series Processors:** 48-bit addressing with no c-bit, SMEE/SEV off.
  - 43-bit address with SEV in 509-key mode, 44-bit in 253 key mode.
- **AMD EPYC 7002 Series Processors:** 48-bit addressing with no c-bit, SMEE/SEV off.
  - 43-bit address with SEV in 509-key mode, 44-bit in 253 key mode.
- **AMD EPYC 7001 Series Processors:** 48-bit addressing with c-bit, SMEE/SEV off.
  - 43-bit address with SME/SEV (16 keys).

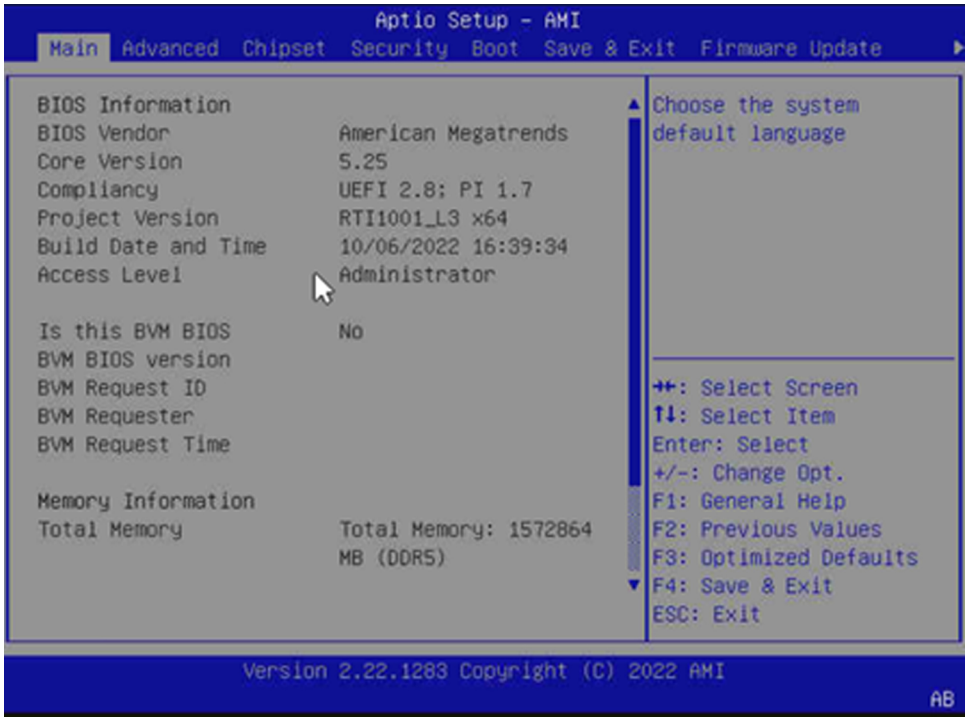
2.5 Enabling TSME on All Processors:

Transparent Secure Memory Encryption (TSME, also known as Secure Memory Encryptio) uses a single key to encrypt system memory. The AMD Secure Processor generates this key at boot. TSME requires enablement in the system BIOS and offers transparent memory encryption that can run with any operating system. TSME is separate from SEV, and you need not run SEV in order to benefit from TSME. TSME is disabled by default.

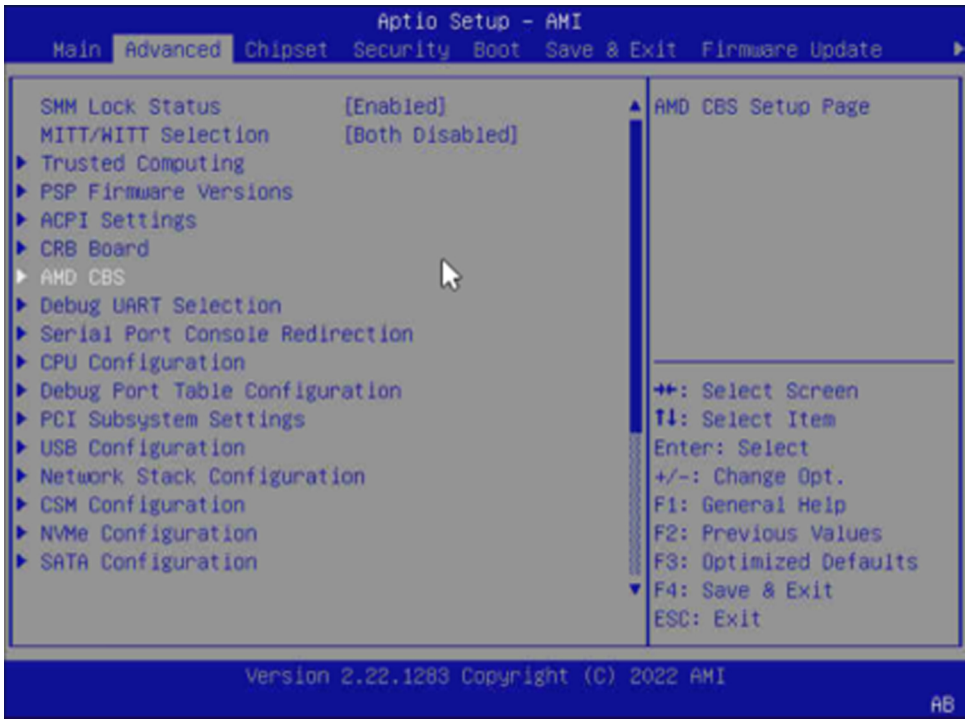
2.5.1 Enabling TSME on All Processors

To enable TSME on an AMD CRB:

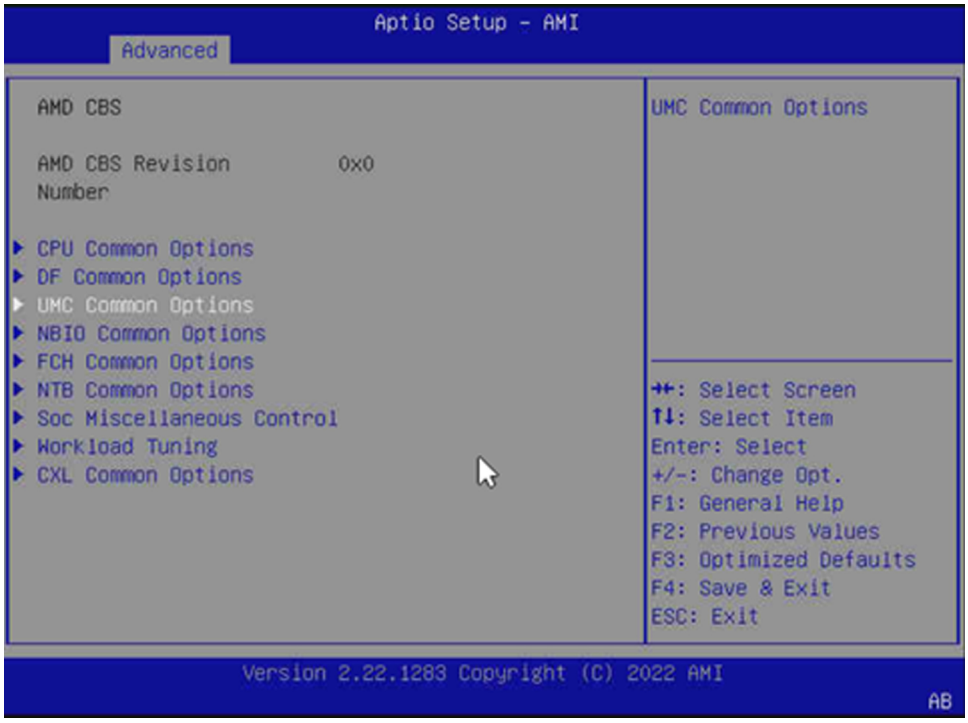
1. Access the system BIOS.
2. Select **Advanced**.



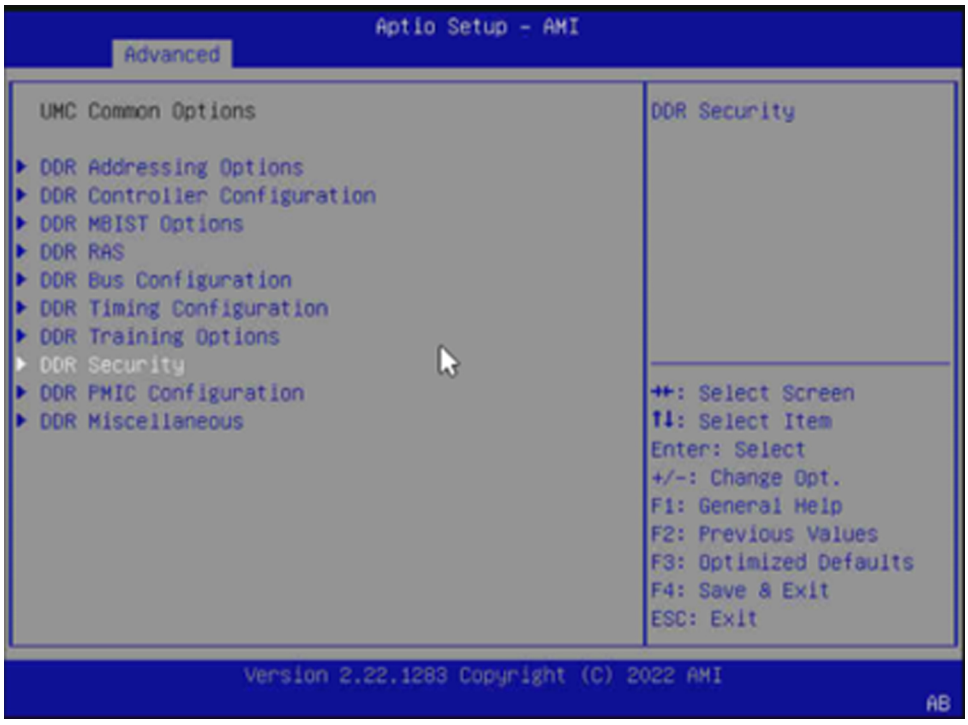
3. Select **AMD CBS**.



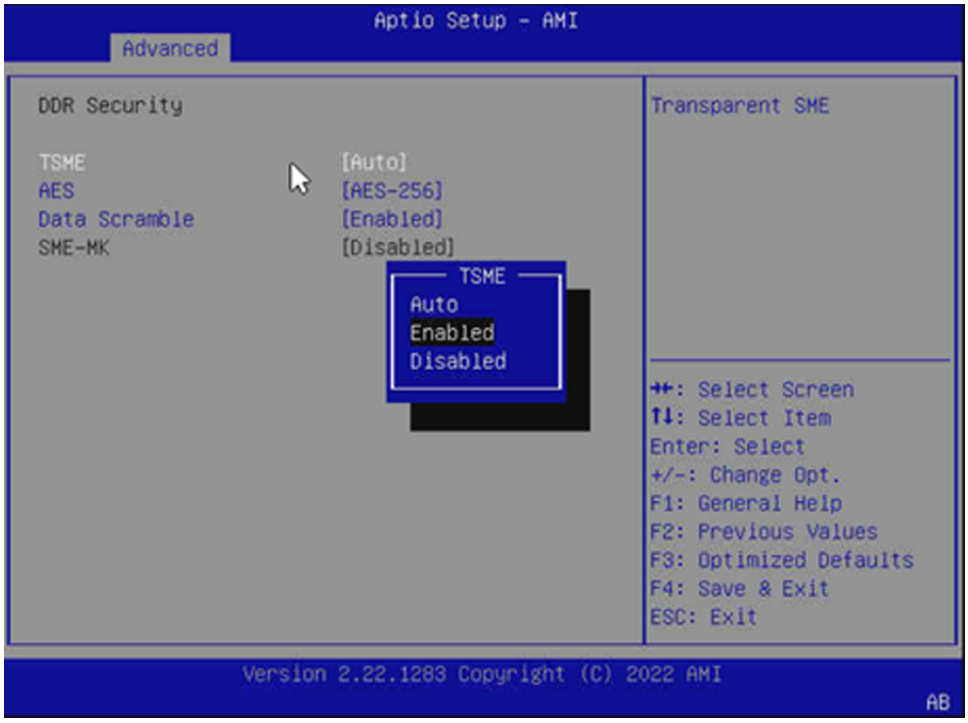
4. Select **UMC Common Options**.



5. Select **DDR Security**.



6. Set **TSME** to **Enabled**.



2.5.2 Disabling TSME on All Processors

To disable TSME on an AMD CRB:

1. Access the system BIOS.
2. Select **Advanced > AMD CBS > UMC Common Options > DDR Security**.
3. Set **TSME** to either **Disabled** or **Auto**.

Chapter 3 Configuring SEV

This chapter describes how to configure the Secure Encrypted Virtualization (SEV) feature.

3.1 AMD EPYC 9004 Series Processors

1. In BIOS, select **Advanced > AMD CBS > CPU Common Options**, and then set the **SEV Control** parameter to **Enable**.

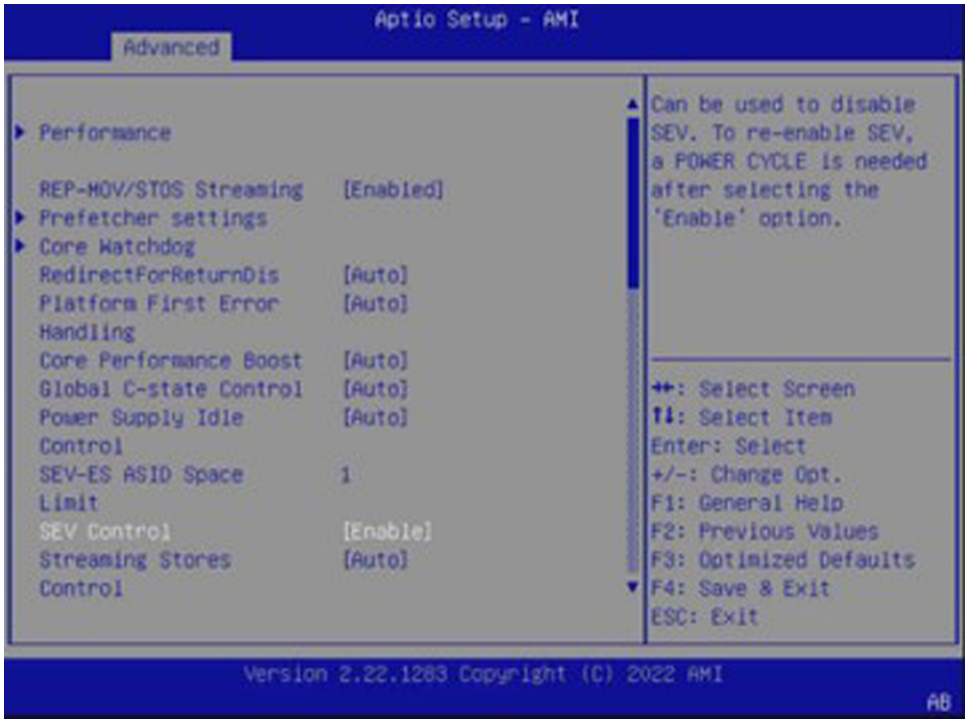


Figure 3-1: Setting SEV-ES Control to Enabled (AMD EPYC 9004 Series Processors)

2. Select **Advanced > AMD CBS > CPU Common Options**, and then change the **SEV-ES ASID Count** from **Auto (1006)** to 1006 or below to change the maximum number of ASIDs and the maximum amount of addressable DRAM. Set SEV-ES **ASID Space Limit** to the desired value based on the types of VMs you will be running. ASIDs less than 'x' are for SEV-ES, and ASIDs greater than or equal to 'x' are for SEV. For example, if 5 is input in the field, then there will be 4 available SEV-ES ASIDs and the rest will be SEV only. If the field is set to 1, then SEV-ES will be disabled because

there are no available ASIDs for SEV-ES.See the **minSEVSASID** question in "Frequently Asked Questions" on page 37 for more detailed information.

Figure 3-2: Configuring SEV-ES ASID Space Limit (AMD EPYC 9004 Series Processors)

3.2 AMD EPYC 7003 and 7002 Series Processors

To configure SEV on a system powered by an AMD EPYC 7003 or 7002 Series Processor:

1. In BIOS, select **Advanced > AMD CBS > CPU Common Options**, and then set the **SEV-ES ASID Space Limit Control** parameter to **Manual**.

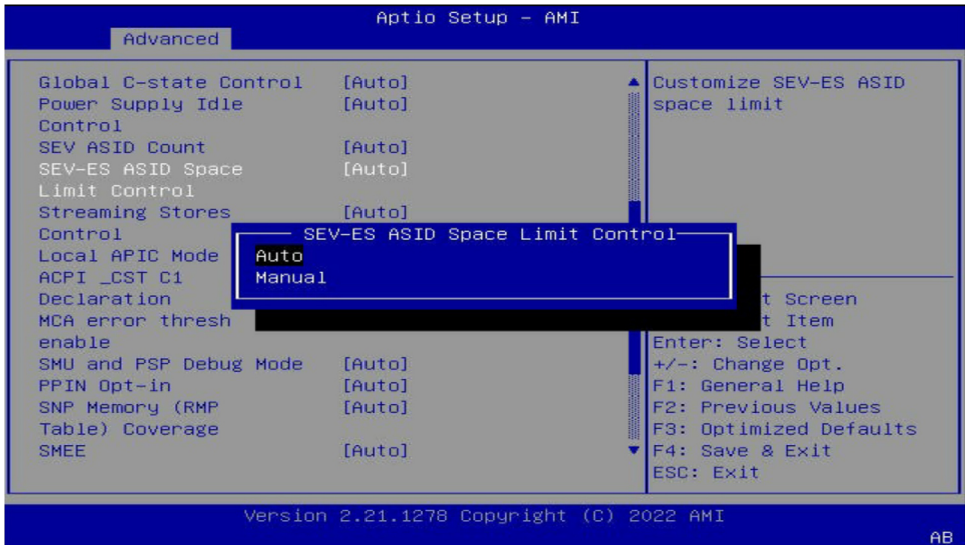


Figure 3-3: Setting SEV-ES Space Limit Control to Manual (AMD EPYC 7003 and 7002 Series Processors)

2. Select **Advanced > AMD CBS > CPU Common Options**, and then change the **SEV-ES ASID** Count from **Auto (509/253)** to **509/253** or less to change the maximum number of ASIDs and the maximum amount of addressable DRAM. Set **SEV-ES ASID Space Limit** to the desired value based on the types of VMs you will be running. ASIDs less than 'x' are for SEV-ES, and ASIDs greater than or equal to 'x' are for SEV. For example, if 5 is input in the field, then there will be 4 available SEV-ES ASIDs and the rest will be SEV only. If the field is set to 1, then SEV-ES will be disabled because there are no available ASIDs for SEV-ES. See the **minSEVSASID** question in "Frequently Asked Questions" on page 37 for more detailed information.

Note: If the system detects 8TB or more of DRAM, then BIOS will automatically switch this to 253 ASIDs.

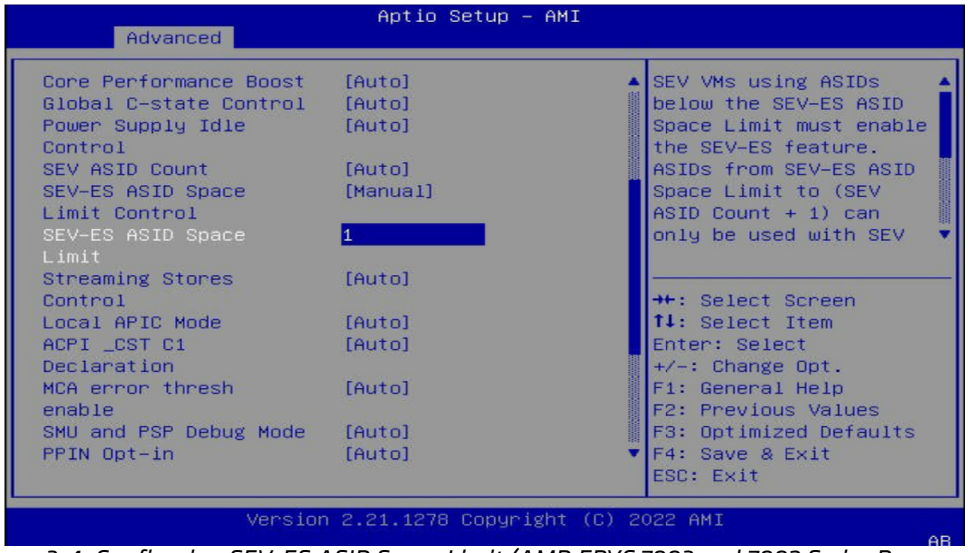


Figure 3-4: Configuring SEV-ES ASID Space Limit (AMD EPYC 7003 and 7002 Series Processors)

3.3 AMD EPYC 7001 Series Processors

1. In BIOS, select **Advanced > AMD CBS > CPU Common Options**, and then set the **SEV-ES ASID Space Limit Control** parameter to **Manual**.

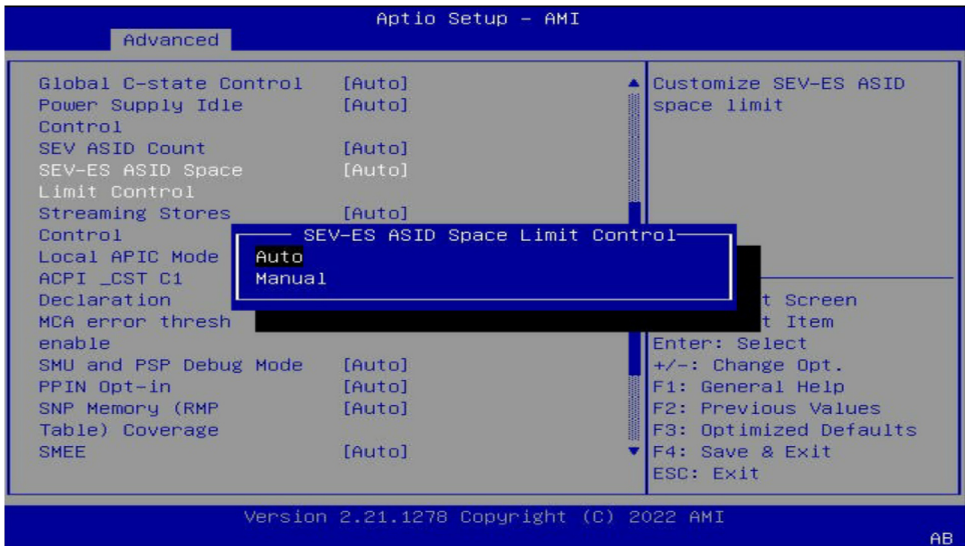


Figure 3-5: Setting SEV-ES Space Limit Control to Manual

2. Set **SEV-ES ASID Space Limit (16)** to the desired value based on the types of VMs you will be running. ASIDs less than 'x' are for SEV-ES and ASIDs greater than or equal to 'x' are for SEV. See the **minSEVSASID** question in "Frequently Asked Questions" on page 37 for more detailed information. AMD recommends leaving this setting at either **Auto** or **1**.

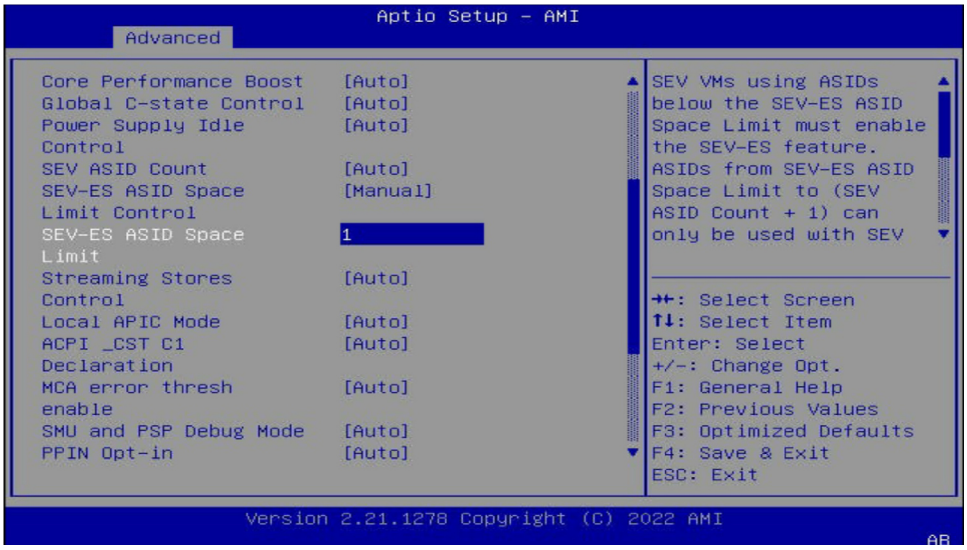


Figure 3-6: Configuring SEV-ES ASID Space Limit

Chapter 4

Enabling/Disabling SNP

This chapter describes how to enable and disable the AMD Secure Nested Paging (SNP) feature. This only applies to AMD EPYC 7003 Series Processors and above.

4.1 Enabling SNP

To enable SNP:

1. Enable and configure SEV and SEV-ES, as described in "Configuring SEV" on page 15.

Note: SNP only works on ASIDs that are SEV-ES capable (below MinSEVASID).

2. In the system BIOS, select **Advanced > AMD CBS > CPU Common Options**.
3. Change **SNP Memory (RMP Table) Coverage** from **Auto** (which means **Disabled**) to **Enabled**. This will reserve memory for SNP and create the RMP that covers all of memory. If needed, you can select **Custom** to set the RMP to not cover all of memory.

Note: This only required for Linux hosts. Microsoft hosts do not require this when using SEV-SNP under Hyper-V.

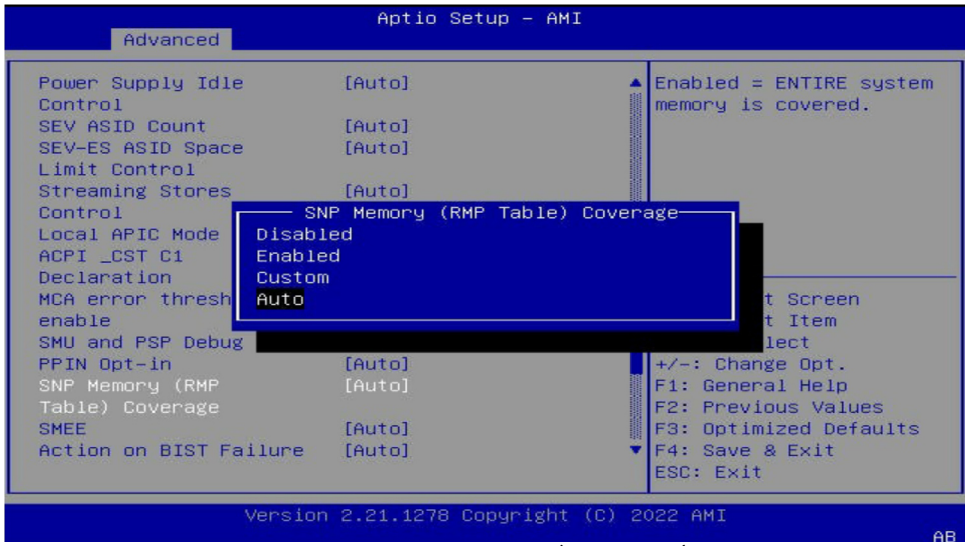


Figure 4-1: Changing SNP Memory (RMP Table) Coverage

You can also do this using MSRs. Before enabling SNP, first zero the RMP memory, and then write the address of the memory into the MSRs.

- `MSRC001_0132 [RMP Base] (Core::X86::Msr::LS_RMP_BASE)`
- `MSRC001_0133 [RMP End] (Core::X86::Msr::LS_RMP_END)`
- Enable SNP by setting the following MSR to 1:
  `MSRC001_0010 [System Configuration] (Core::X86::Msr::SYS_CFG) bit 25 VmplEn set to 1`

Please see Sections 15.26.4 and 15.36.1 in Volume 2 of the AMD Architecture Programmer's Manual for more information on RMP programming.

Next, configure the IOMMU to disable the vIOMMU:

1. In BIOS, select **Advanced > AMD CBS > NBIO Common Options**.

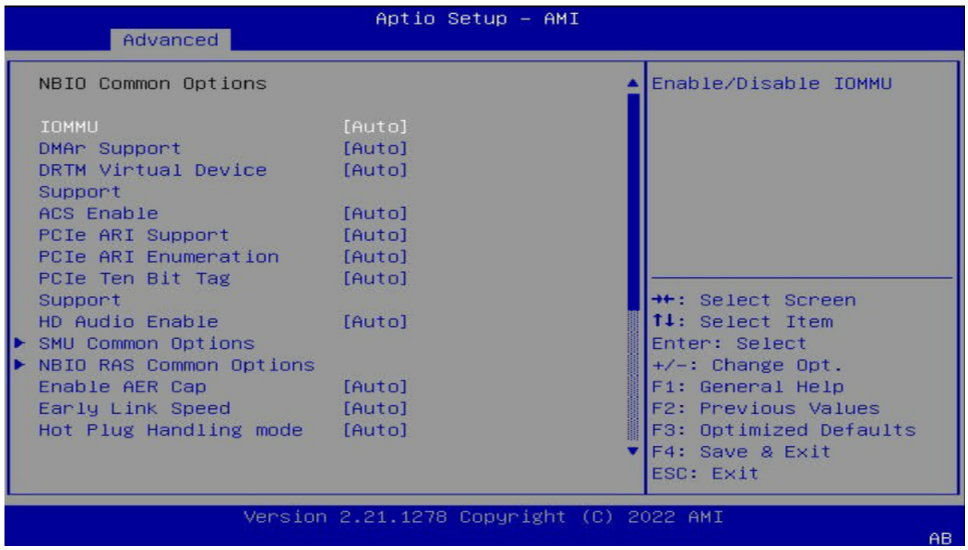


Figure 4-2: NBIO Common Options

2. Set **SEV-SNP Support** to **Enabled** (default is **Disabled**).

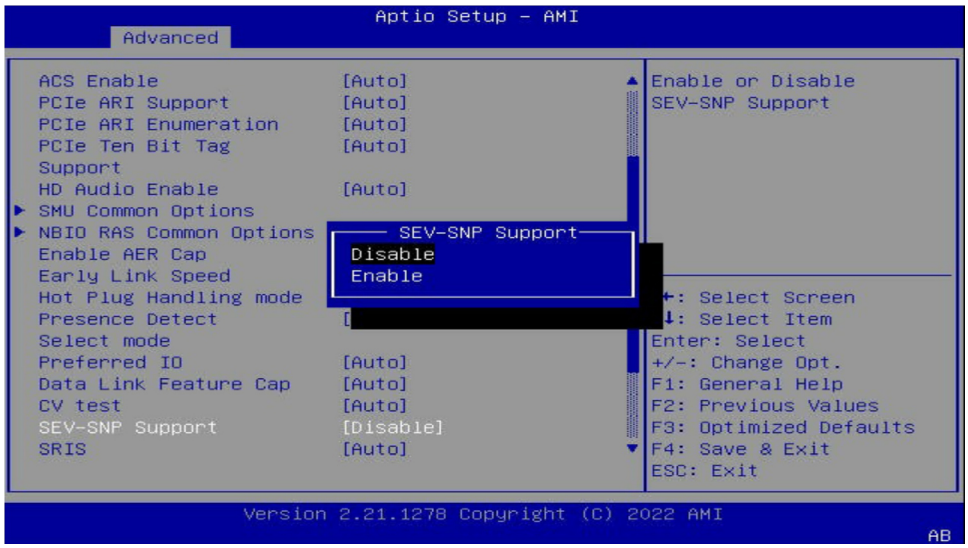


Figure 4-3: Enabling SEV-SNP support

4.2 Disabling SNP

Do not enable the `SecureNestedPagingEn` MSR bit: `MSRC001_0010 [System Configuration] (Core::X86::Msr::SYS_CFG)` bit 24 via x86.

Note: The system BIOS will never enable SecureNestedPagingEn. It always must be enabled by x86.

AMD recommends to leaving **SNP Memory (RNP Table) Coverage** set to **Auto/Disabled** in the BIOS, but there is no harm in leaving it **Enabled** if the hypervisor eventually wants to enable SNP. Leaving SNP memory coverage enabled will only remove some usable memory from the system.

The `RMPBase` and `RMPEnd` settings do not matter because RMP protection is not in effect since the `SYS_CFG` MSR for `SecureNestedPagingEn` (bit 24) is disabled.

This page intentionally left blank.

Chapter 5 OS Requirements

For SEV or SEV-ES, verify that your OS supports SEV as a hypervisor and/or SEV as a guest, as shown in the following tables.

Note: TSME is OS-independent and only needs enablement in the BIOS.

5.1 SEV

The following kernels/OS support SEV:

OS/KERNEL	HOST	GUEST
Linux 4.15		☑
Linux 4.16	☑	☑
RHEL 7.6		☑
RHEL 8	☑	☑
Fedora 28	☑	☑
SLES 15	☑	☑
Ubuntu 18.04		☑
Ubuntu 18.10	☑	☑
Oracle UEK 5	☑	☑

Table 5-1: SEV support

5.2 SEV-ES

The following kernels/OS support SEV-ES:

OS/KERNEL	HOST	GUEST
Linux 5.10		☑
Linux 5.11	☑	☑

Table 5-2: SEV-ES support

5.3 SEV-SNP

The following kernels/OS support SEV-SNP:

OS/KERNEL	HOST	GUEST
Linux 5.19	in development	☑

Table 5-3: SEV-SNP support

The SNP firmware requires IOMMU security protection, and a special OS kernel is required that knows how to configure the IOMMU. IOMMU must be enabled in BIOS. You can then use development kernels until the SNP patches have been merged into the main Linux kernel. See https://github.com/AMDESE/AMDSEV/blob/sev-snp-devel/stable-commits.

Chapter 6 OS Enablement

6.1 Checking SEV Enablement

Execute the following command to find all SEV kernel prompts:

```
$ sudo dmesg | grep SEV
```

- **SEV:** You should see either:
  - `[CCP VALUE] SEV supported`
  - `[CCP VALUE] SEV supported: 'xxx' ASIDs`

Note: Both are valid, depending on the kernel version.

- **SEV-ES:** You should see:
  - `[CCP VALUE] SEV-ES supported: 'xxx' ASIDs`

For example, when both SEV and SEV-ES are enabled::

```
root@  :~# dmesg | grep SEV
[   14.886391] ccp 0000:47:00.1: SEV firmware update successful
[   15.140921] ccp 0000:47:00.1: SEV API:1.51 build: 3
[   15.229519] SEV supported: 'xxx' ASIDs
[   15.229520] SEV-SEC supported: 'xxx' ASIDs
```

In the above example:

- The number before `ASIDs` is the number of available ASIDs for the given SEV feature. SEV-ES ASIDs are meant for both SEV-ES and SEV-SNP.
- If either of the prompts do not appear, then verify that SEV and SEV-ES have been correctly enabled in the system, as described in the previous chapters. If so, then you must enable SEV and SEV-ES in the kernel, as described in the following section.

6.2 Enabling SEV

If SEV still does not appear in the kernel message after enabling it in BIOS, then you might need to enable it at the kernel level. To enable SEV in the kernel:

1. Append the following to the kernel command line options:
   `kvm_amd.sev=1 kvm_amd.sev_es=1`

2. Update `grub` in the OS.
3. Reboot the machine. SEV should now be enabled in the host OS.
4. In any guest, check for enablement by executing the same command shown in "Checking SEV Enablement" on page 25:

```
root@localhost:~# dmesg|grep SEV
[ 0.145741] Memory Encryption Features active: SMD SEV SEV-ES
```

6.2.1 Additional Resources

Please see the following resources for additional information:

- **Kernel.org:** https://www.kernel.org/doc/html/v5.7/virt/kvm/amd-memory-encryption.html
- **RHEL:** https://access.redhat.com/articles/4491591
- **Oracle:** https://blogs.oracle.com/linux/post/using-amd-secure-memory-encryption-with-oracle-linux
- **SUSE:** https://documentation.suse.com/sles/15-SP1/html/SLES-amd-sev/index.html

6.3 Enabling SEV-SNP

DISCLAIMER: As of February 2023, SNP is still not supported upstream. You can follow these steps to build a demo kernel and get a look at an early version of SNP.

To enable SEV-SNP at the host level:

1. Follow the procedure described in "Enabling SEV" on page 25 to enable SEV.
2. Verify that the current firmware installed is the newest available (1.54 at the time of publication) for SNP-compatible AMD EPYC 7003 or 9004 Series Processor. If needed, update the firmware as described in "Updating SEV Firmware" on page 27.
3. Follow the steps listed in https://github.com/AMDESE/AMDSEV/tree/sev-snp-devel to build and install newest SNP kernel.
4. Execute the command described in "Checking SEV Enablement" on page 25 to verify that SEV-SNP is enabled. For example:

```
[     0.720169] SEV-SNP: RMP table physical address 0x000000003a00000 - 0x00000000568fffff
[     6.560584] ccp 0000:47.00.1: SEV firmware update successful
[     8.151665] ccp 0000:47.00.1: SEV API:1.51 build:3
[     8.151674] ccp 0000:47.00.1:
[     8.161364] SEV supported: 410 ASIDs
[     8.161364] SEV-ES and SEV-SNP supported: 99 ASIDs
```

Chapter 7 Updating SEV Firmware

You should always use the latest SEV firmware supported by your BIOS to have the latest features and security protection. To update SEV firmware:

1. Update your system BIOS.
2. Execute the SEV `DownloadFirmware (DLFW)` command. See "DownloadFirmware" on page 27.
3. Execute the SNP `DownloadFirmwareEX (DLFW_EX)` command. See "DownloadFirmwareEX" on page 29.

The `DownloadFirmware` and `DownloadFirmwareEX` commands replace the local copy of SEV in DRAM with the new image. Calling the next SEV command loads that new copy into SRAM and runs it. The BIOS copy remains in SpiRom; rebooting the system will run the older BIOS image until you execute these commands again to update to the latest version.

7.1 DownloadFirmware

The `DownloadFirmware` command allows system administrators to the version of SEV running on the platform without having to reboot the platform or update the BIOS, provided that:

- All SEV/SNP guests are shut down.
- The SEV/SNP platform state is UNINIT.

The Linux CCP driver will automatically check for a new SEV image when initialized. If it finds a new image, then it will execute the `DownloadFirmware` command.

1. Download the latest firmware version from https://developer.amd.com/sev/.
2. Check the /lib/firmware/amd/ directory to determine the system firmware format (`.sbin` or `.esbin`).
3. Copy the appropriate firmware filw (`.sbin` or `.esbin`) to `/lib/firmware/amd/`, and then name the file `amd_sev_fam[ family ]h_model[ model ]h.sbin` or `amd_sev_fam[ family ]h_model[ model ]h.esbin` (see Figure 5-1). If needed, you may create an `/amd` folder, as shown in Figure 5-2, then paste the `.sbin`/`.esbin` into this folder (see Figure 5-3), and then rename the firmware file as shown in Figure 5-4.

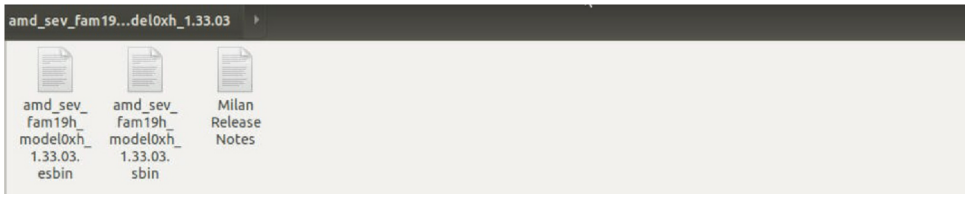


Figure 7-1: Firmware download example

Figure 7-2: lib_firmware folder, with the /amd subfolder created.

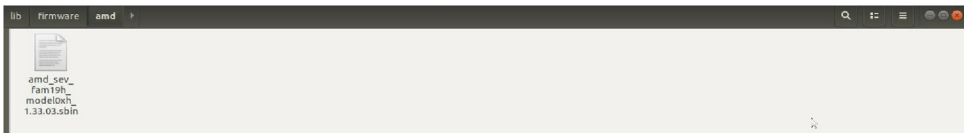
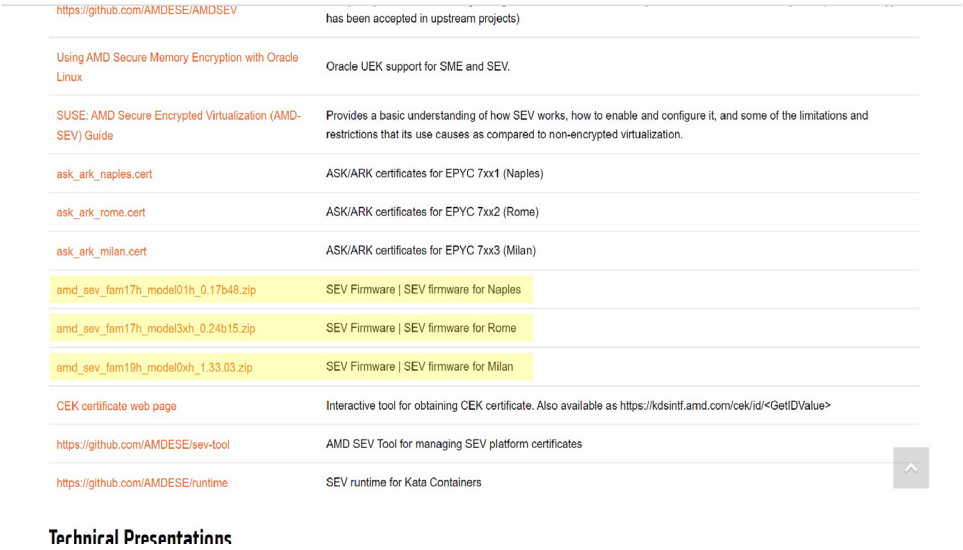


Figure 7-3: Pasted .sbin before renaming

Figure 7-4: Pasted .sbin after renaming

You can find the latest SEV firmware images at https://developer.amd.com/sev/. See Figure 5-5.



https://github.com/AMDESE/AMDSEV	has been accepted in upstream projects)
Using AMD Secure Memory Encryption with Oracle Linux	Oracle UEK support for SME and SEV.
SUSE: AMD Secure Encrypted Virtualization (AMD-SEV) Guide	Provides a basic understanding of how SEV works, how to enable and configure it, and some of the limitations and restrictions that its use causes as compared to non-encrypted virtualization.
ask_ark_naples.cert	ASK/ARK certificates for EPYC 7xx1 (Naples)
ask_ark_rome.cert	ASK/ARK certificates for EPYC 7xx2 (Rome)
ask_ark_milan.cert	ASK/ARK certificates for EPYC 7xx3 (Milan)
amd_sev_fam17h_model01h_0.17b48.zip	SEV Firmware | SEV firmware for Naples
amd_sev_fam17h_model3xh_0.24b15.zip	SEV Firmware | SEV firmware for Rome
amd_sev_fam19h_model0xh_1.33.03.zip	SEV Firmware | SEV firmware for Milan
CEK certificate web page	Interactive tool for obtaining CEK certificate. Also available as https://kdsintf.amd.com/cek/id/<GetIDValue>
https://github.com/AMDESE/sev-tool	AMD SEV Tool for managing SEV platform certificates
https://github.com/AMDESE/runtime	SEV runtime for Kata Containers

Technical Presentations

Figure 7-5: SEV firmware download links

AMD EPYC 7002 Series Processors and newer always support SEV. AMD EPYC 7001 Series Processors require SEV firmware version 0.16 or above to run SEV. Please see the SEV Specification for additional information.

7.2 DownloadFirmwareEX

The `DownloadFirmwareEX` command only applies to 3rd Gen AMD EPYC processors and later. This command allows system administrators to the version of SEV running on the platform without having to reboot the platform or update the BIOS. SNP guests may remain running during the update, but all SEV guests must be shut down. The exception is that you may be required to shut down the guests or or uninitialize the SNP platform in certain cases, such as if a security bug was found in a previous version and the running guests cannot be upgraded securely.

The minimum version requirements for this command are:

- **PSP Bootloader:** 00.13.00.60 (Milan PI 1004 BIOS).
- **SEV uapp version:** 1.2B.2B (around Milan PI 1007 BIOS).

If you are running a SEV version that does not support DLFW_EX, then you will have to first shut down your guests and then call the regular DLFW command (see "DownloadFirmware" on page 27) to upgrade to the SEV version that supports `DownloadFirmwareEX` and then use `DownloadFirmwareEX` going forward.

Please see the SEV Specification for additional information.

This page intentionally left blank.

Chapter 8 Launching an Encrypted VM

8.1 Launching a VM with SEV Encryption

To launch a VM with SEV encryption, enable SEV in the system as described in "Enabling SEV" on page 25, and then verify that you have the following minimum versions:

PROJECT	VERSION
Libvert	4.5
QEMU	2.12
OVMF	Commit newer than (75b7aa9528bd 2018-07-06)

Table 8-1: Minimum project versions to support SEV-encrypted VMs

8.1.1 Launching with QEMU

In the desired launch directory:

1. Create a new `qcow2` image:
   ```
   $ qemu-img create -f qcow2 encryptedImage.qcow2 30G
   ```
2. Copy the OVMF_VARS.fd file:
   ```
   $cp /usr/share/OVMF/OVMF_VARS.fd OVMF_VARS.fd
   ```
3. Launch your VM using your desired ISO image and the following commands as a minimum:
   ```
   $ qemu-system-x86_64 \
   -enable-kvm \
   -cpu EPYC \
   -machine q35 \
   -no-reboot \
   -vga std \
   -vnc :0
   -drive file=distro.iso=cdrom -boot d \
   -drive if=pflash,format=raw,unit=0,file=/usr/share/OVMF/OVMF_CODE.fd,readonly=on \
   -drive if=pflash,format=raw,unit=1,file=OVMF_VARS.fd \
   -drive file=encryptedImage.qcow2,if=none,id=disk0,format=qcow2
   -device virtio-scsi-pci,id=scsi0,disable-legacy=on,iommu_platform=on
   -device scsi-hd,drive=disk0
   -machine memory-encryption=sev0,vmport=off
   -object sev-guest,id=sev0,policy=0x3,cbitpos=47,reduced-phys-bits=1
   ```

In the preceding example, the `cbitpos` parameter in the line `- object sev-guest,id=sev0,policy=0x3,cbitpos=47,reduced-phys-bits=1` changes depending on the processor generation. AMD EPYC 7002 and 7001 Series Processors have a c-bit value of 47, and AMD EPYC 7003 Series Processors and newer have a c-bit value of 51.

If you are not sure what the appropriate cbit is, then you may check the EBX register on the `0x8000001f` CPUID function by executing the CPUID command:

```
$ cpuid -r -1 0x8000001f
I
```

In this example:

- The bits 0-5 make up the appropriate cbit value.
- EBX is a hex number; you may need a conversion to find this value.
  ```
  amdsev@amdsev:~$ cpuid -r -l -l 0x8000001f CPU:
    0x8000001f 0x00: eax=0x0101fd3f ebx=0x00004173 ecx=0x000001fd edx=0x00000064
  ```

Note: You may need to edit these commands to suit your particular needs and use cases. For example, different distros may have different QEMU launch commands. Please see the guides listed in "Additional Resources" on page 26 *for more information.*

4. Launch the VM, and then install the distro. You can now launch the VM using the `qcow2` image without using the ISO.
5. On the guest, execute the `dmesg | grep SEV` command to verify that SEV is enabled.
   ```
   root@localphost:!# dmesg | grep SEV
   [    0.150352] Memory ENcryption Features active: AMD SEV
   ```

8.1.2 Launching with Libvirt

Please see https://libvirt.org/kbase/launch_security_sev.html for instructions on launching encrypted VMs with Libvirt.

8.2 Launching a VM with SEV-ES Encryption

To launch a VM with SEV-ES encryption, enable SEV in the system as described in "Enabling SEV" on page 25, and then verify that you have the following minimum versions:

PROJECT	VERSION
Libvert	4.5
QEMU	6.0
OVMF	Commit newer than (EDK2-STABLE 2020-21-02)

Table 8-2: Minimum project versions to support SEV-encrypted VMs

1. If needed, install the correct versions.

2. Execute the launch command, which is very similar to the command used for "Launching a VM with SEV Encryption" on page 31, except for the following line:
`-object sev-guest,id=sev0,policy=0x3,cbitpos=47,reduced-phys-bits=1`, where the `policy` variable should be changed to reflect SEV-ES enablement, as shown in the following table:

OFFSET	BIT(S)	NAME	DESCRIPTION
000h	0	NODBG	Debugging of the guest is disallowed when set.
	1	NOKS	Sharing keys with other guests is disallowed when st.
	2	ES	SEV-ES is required when set.
	3	NOSEND	Sending the guest to another platform is disallowed when set.
	4	DOMAIN	The guest must not be transmitted to another platform that is not in the domain when set.
	5	SEV	The guest must not be transmitted to another platform that is not SEV-capable when set.
	15:6	Reserved; should be 0.	
002h	7:0	API_MAJOR	The guest must not be transmitted to another platform with a lower firmware version.
003h	7:0	API_MINOR	

Table 8-3: SEV policy bits

As shown in the previous table:

- The policy bit 2 must be set to launch SEV-ES. The policy is passed as a hexadecimal number.
- A valid SEV-ES configuration would look like this:
`-object sev-guest,id=sev0,policy=0x5,cbitpos=47,reduced-phys-bits=1`
- Everything else is the same as SEV.
- On the guest, execute the command `dmesg | grep SEV` to confirm SEV-ES enablement.

8.3 Launching a VM with SEV-SNP Encryption

As of publication, SEV-SNP does not yet have upstream QEMU or OVMF patches. The guest kernel is currently the only item with upstream support. See "OS Enablement" on page 25 for version information. You can build SNP-compatible OVMF and QEMU at https://github.com/AMDESE/AMDSEV/tree/sev-snp-devel.

Build the correct OVMF and QEMU, and then launch an SNP guest by executing a command similar to that used for regular SEV:

```
$ PATH-TO-SNP-QEMU/qemu-system-x86_64 \
-enable-kvm \
-cpu EPYC \
-machine q35 \
-no-reboot \
-vga std \
-vnc :0\
-drive if=pflash,format=raw,unit=0,file=PATH-TO-SNP-OVMF/OVMF_CODE.fd,readonly=on \
-drive if=pflash,format=raw,unit=1,file=OVMF_VARS.fd \ <- make sure you copy this file from
build ovmf
-drive file=SNPGUEST.qcow2,if=none,id=disk0,format=qcow2 \
-device virtio-scsi-pci,id=scsi0,disable-legacy=on,iommu_platform=on \
```

```
-device scsi-hd,drive=disk0 \
-machine memory-encryption=sev0,vmport=off \
-object sev-snp-guest,id=sev0,cbitpos=51,reduced-phys-bits=1
```

This command should allow you to launch an SNP-enabled VM if your guest has the correct kernel. You can execute the command `dmesg | grep SEV` on the guest to confirm that SNP is launched:

```
root@localhost:~# dmesg | grep SEV
[    0.150352] Memory Encryption Features avcoe: AMD SEV SEV-ES SEV-SNP
```

Note: LibVirt currently does not support SNP.

Chapter

9

Confidential Containers

SEV is now supported on confidential containers via an open-source that allows you to launch SEV encrypted kata-containers. Please visit https://github.com/confidential-containers/documentation/blob/main/quickstart.md for information and instructions on how to set-up confidential containers.

This page intentionally left blank.

Chapter 10 Frequently Asked Questions

What is MinSEVASID?

MinSEVAsid is the minimum ASID that lets you run SEV guests, everything below that is for SEV-ES and SEV-SNP guests. For example, if MinSEVAsid is set to 8, then ASIDs 1-7 can only be assigned to SEV-ES or SEV-SNP guests, and ASIDs 8-(max) can only be used for SEV guests.

How do I map more than 8TB/16TB of physical address space?

To map to more than 8TB of physical address space (DRAM + PCIe + MMIO, etc), change **SEV ASID Count** to 253 in the BIOS. AMD EPYC 7003 and 7002 AGESA will automatically change this setting to 253 if more than 8TB of physical address space is detected during boot. You must disable SME (which also disables SEV) to map to more than 16TB of physical address space. See "Disabling SMEE in BIOS" on page 9.

How many bits are being used by ASIDs and where is the C-bit on my generation of platform?

See "Disabling SMEE via MSR" on page 10 to find the number of bits being used by ASIDs.

- 1st and 2nd Gen AMD EPYC processors have the c-bit in bit 47.
- 3rd and 4th Gen AMD EPYC processors have the c-bit in bit 51.

If in doubt, check `CPUID_Fn8000001F_EBX [AMD Secure Encryption EBX] (Core::X86::Cpuid::SecureEncryptionEbx)` to find the c-bit position. See "Launching a VM with SEV Encryption" on page 31 for additional information.

3rd and 2nd Gen AMD EPYC with 256 ASIDs (8 bits) and 16TB DRAM	3rd and 2nd Gen AMD EPYC with 512 ASIDs (9 bits) and 8TB DRAM
64:52 reserved	64:52 reserved
51:44 asids/cbit cbit=51	51:43 asids/cbit cbit=51
43:0 PhysAddr	42:0 PhysAddr

Table 10-1: ASID bit usage

Where is the SEV documentation?

See https://developer.amd.com/sev/.

Does the APM vol 2 support SEV and SNP?

Yes. See https://www.amd.com/system/files/TechDocs/24593.pdf.

What is SEV 2?

This is not an official AMD term but may refer to the second implementation of SEV (on 2nd Gen AMD EPYC processors that have 509 ASID keys).

I have questions about PCDs

Please contact the AMD BIOS support team. The firmware team does not know about PCDs. The BIOS documentation on SEV-related options should be good enough on its own or will need to be updated.

How big will my RMP be for a given amount of memory?

Each RMP entry is 16 bytes, and 256 RMP entries can fit in a 4K page. So, for 512 GB of DRAM:

- 512*1024*1024*1024 bytes / 4096 = 134,217,728 4K pages
- 134,217,728 4K pages * 16 Bytes per RMP entry = 2,147,483,648 Bytes for all RMP entries
- 2,147,483,648 Bytes for all RMP entries / (1024*1024) = 2,048 MB = 2GB (approx.)

How do I disable SEV?

The easiest way is to disable SMEE in the BIOS (see "Disabling SMEE in BIOS" on page 9). If you want to still use SME but not SEV, then you can blacklist the `ccp` kernel driver so it doesn't load SEV. The last option is to remove the SEV binary from the BIOS, but that is not recommended.

How do I disable SNP?

Don't reserve memory for the RMP in the BIOS, and don't set the SNP_EN MSR from x86. See "Enabling/Disabling SNP" on page 19 for more information.

Can we request new security features in later-generation AMD EPYC processors or to SEV?

The SEV spec is generally thought to be final, except for any security issues. Any new features will go into SNP.

How do I check if TSME is enabled?

You can check the kernel message to see if TSME is enabled by executing the command `dmesg | grep SME`, which should return a message similar to `AMD Memory Encryption Features active: SME`.

Additionally, a SNP guest can send a `MSG_REPORT_REQ` guest message to the PSP to get the SNP attestation report. Bit 1 (`tsme_en`) of the `PLATFORM_INFO` field contains the `tsme_en` info.

Which version of SEV firmware did 'x' support get added?

Feature	SEV Firmware Version
• DownloadFirmware • GetID	0.16
• ActivateEX • Enhanced DownloadFirmware (PSP firmware dependency checking) • InitEX	0.18
• SwapIn/SwapOut • NOP • SendCancel • Attestation	0.23
RingBuffer	0.24

Table 10-2: SEV firmware versions with 'x' support

This page intentionally left blank.

Chapter

11

Performance Data

Please see Application Note: AMD SEVSNP Workloads Performance And Best Practices for AMD EPYC™ 7003 Series Processors (login required) for information about SEV performance.

This page intentionally left blank.