Enabling AMD Security Features (SME, SEV and SEV-ES) on ThinkSystem Servers

Planning / Implementation

Prior 2016, server administrators concerned with data security focused mainly on disk encryption, however with the availability of non-volatile memory technology, the desire to also focus on protecting data in memory also became commonplace. AMD first introduced Secure Memory Encryption (SME) and the first generation Secure Encrypted Virtualization (SEV) technology with the release of the EYPC 7001 processor. To continually enhance security, AMD announced SEV-ES (Encrypted State) to encrypt CPU register state of the virtual machine (VM) with EYPC 7002 series processors in 2017.

Secure Memory Encryption (SME) is mainly for main memory encryption against a variety of attacks such as Coldboot.

It is not only full memory encryption, but also partial memory encryption for the flexible usage and better performance. The other benefit is that no application changes are required. The encryption and decryption process are shown in the figure below.

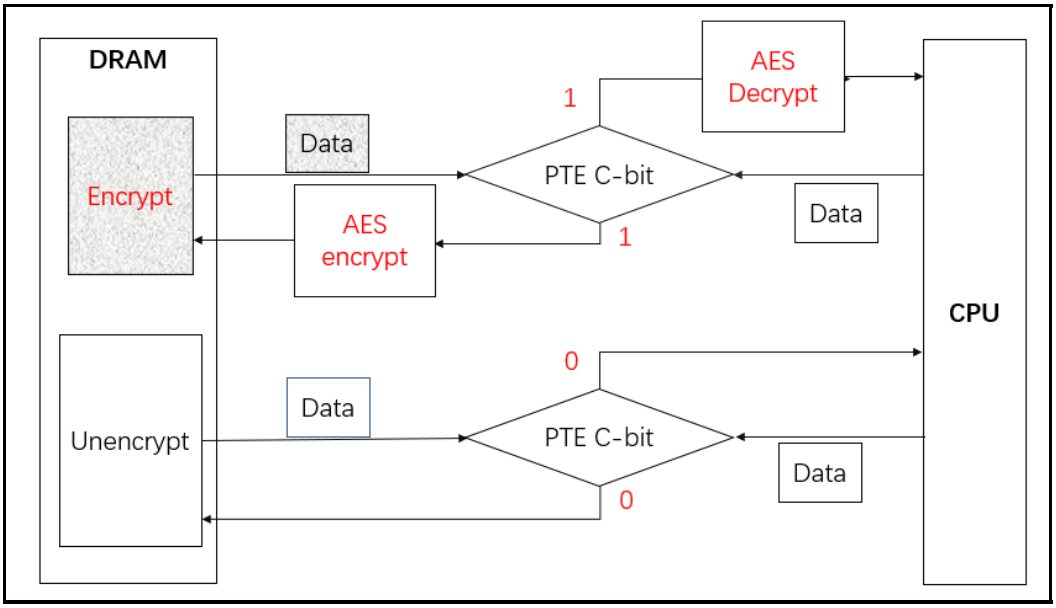

Figure 1. Memory Encryption process

Secure Encrypted Virtualization (SEV) is used to protect virtual machines against the threats from higher privileged code such as the hypervisor. SEV integrates main memory encryption capabilities with the existing AMD-V virtualization architecture. Like SME, SEV also can be used without any application modification to VMs. The SEV architecture is shown in the figure below.

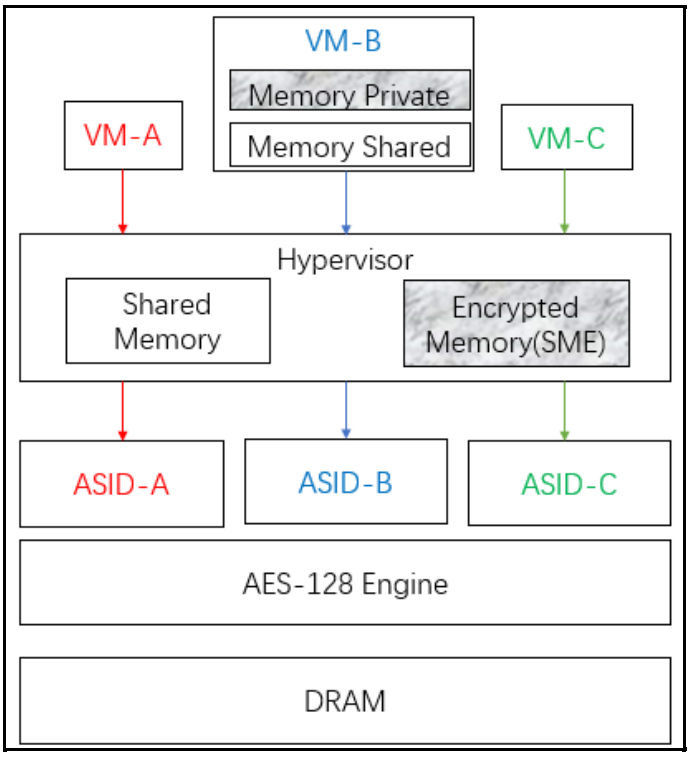

Figure 2. SEV Architecture

For a VM, CPU register protection is very important as well as memory, because the attackers can obtain some information by reading and replacing the register value from a hypervisor. To prevent VM from register's threat, SEV-ES was made available in 2017. In the architecture, it added Virtual Machine Control Block (VMCB) for CPU registers as shown in the figure below, that makes a guest VM easily protects required CPU registers and decreases the attack surface from hypervisor.

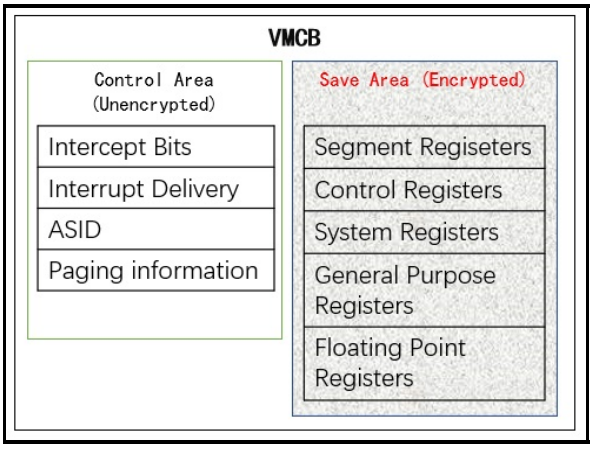

Figure 3. Virtual Machine Control Block (VMCB)

Linux support

The following table shows the minimum supported Linux version for each feature.

Table 1. Linux support

OS	SME	SEV	SEV-ES
Red Hat Enterprise Linux	7.5 and later	8.6 and later	8.6 and later
SUSE Linux Enterprise Server	12 SP4 and later	12 SP4 and later	15 SP3 and later
Ubuntu	18.04 and later	18.04 HWE kernel (v5.4)	20.04 HWE kernel (v5.11)

Enabling SME

Follows these steps to enable SME:

1. To use SME, you need to enable it in UEFI. One method is to use F1 at boot to enter System Setup, go to the Memory section, and set SMEE to Enable as shown below.

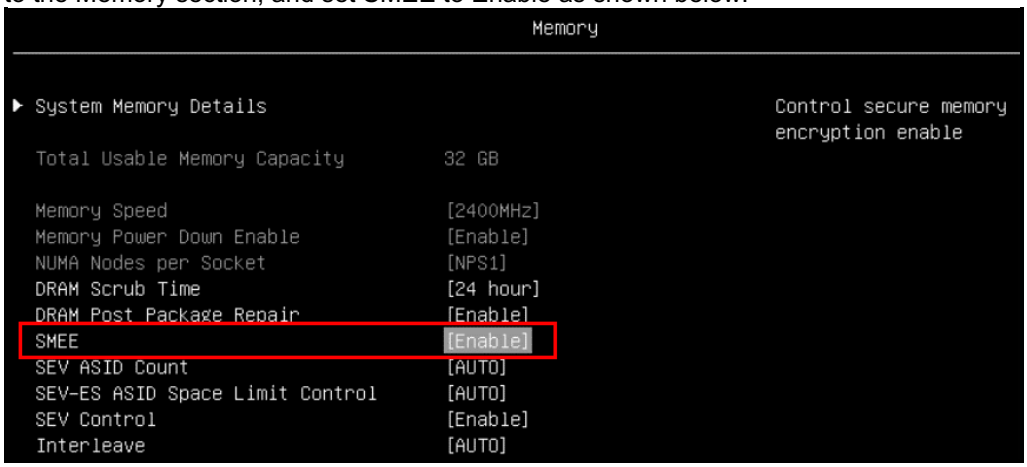

Figure 4. Memory settings in System Setup

Alternatively, you can use the OneCLI command line tool to enable SME using the following command:

```
./lnvgy_utl_lxceb_onecli01l-4.3.0_linux_x86-64.bin config set memory.smee
enabled
```

```
[root@sme ~]# ./lnvgy_utl_lxceb_onecli01l-4.3.0_linux_x86-64.bin config set memory.smee enabled
Extracting...
Executing...


Start to connect current machine's BMC inband to apply config set
Connected to BMC at IP address fe80::0a94:efff:feaa:94a4%enp35s0f3u1u6 by REDFISH
Invoking SET command ...
memory.smee=enabled
Changes completed successfully, but these changes will not take effect until next restart the system.
Succeed.
[root@sme ~]#
```

Figure 5. Enabling SME via OneCLI

2. Restart the server after the command completes.
3. To enable SME in Linux, append boot parameter `mem_encrypt=on` to enable memory encryption:

```
[root@sme ~]# grubby --args="mem_encrypt=on" --update-kernel ALL
[root@sme ~]#
[root@sme ~]# grubby --info DEFAULT
```

```
[root@sme ~]# grubby --args="mem_encrypt=on" --update-kernel ALL
[root@sme ~]#
[root@sme ~]# grubby --info DEFAULT
index=3
kernel="/boot/vmlinuz-5.14.0-284.11.1.el9_2.x86_64"
args="ro crashkernel=1G-4G:192M,4G-64G:256M,64G-:512M resume=/dev/mapper/rhel-swap rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap rhgb quiet $tuned_param
s mem_encrypt=on"
root="/dev/mapper/rhel-root"
initrd="/boot/initramfs-5.14.0-284.11.1.el9_2.x86_64.img $tuned_initrd"
title="Red Hat Enterprise Linux (5.14.0-284.11.1.el9_2.x86_64) 9.2 (Plow)"
id="b6099645bc7c448a9c4807f81bbb0b1d-5.14.0-284.11.1.el9_2.x86_64"
[root@sme ~]#
```

Figure 6. Enabling SME in Linux

4. Restart the OS to activate SME.

To verify that SME is enabled, you can check the dmesg log to see what memory features are active:

```
[root@sme ~]# dmesg | grep SME
[    0.066635] Memory Encryption Features active: AMD SME
[root@sme ~]#
```

Enabling SEV

Follows these steps to enable and verify SEV:

1. To use SEV, you need to enable it in UEFI. One method is to use F1 at boot to enter System Setup, go to the Memory section, and set SMEE to Enable as shown below.
   Tip: SMEE is the only setting you need to enable for SEV.

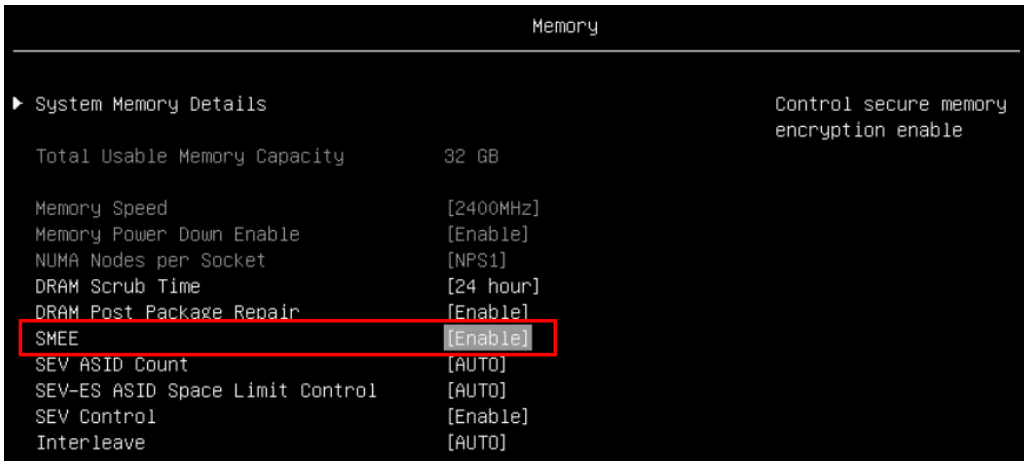

Figure 7. Memory settings in System Setup

Alternatively, you can use the OneCLI command line tool to enable SME using the following command:

```
./lnvgy_utl_lxceb_onecli01l-4.3.0_linux_x86-64.bin config set memory.smee
enabled
```

```
[root@sme ~]# ./lnvgy_utl_lxceb_onecli01l-4.3.0_linux_x86-64.bin config set memory.smee enabled
Extracting...
Executing...


Start to connect current machine's BMC inband to apply config set
Connected to BMC at IP address fe80::0a94:efff:feaa:94a4%enp35s0f3u1u6 by REDFISH
Invoking SET command ...
memory.smee=enabled
Changes completed successfully, but these changes will not take effect until next restart the system.
Succeed.
[root@sme ~]#
```

Figure 8. Enabling SME via OneCLI

2. Restart the server after the command completes.
3. In Linux, install the required packages using the following command:

```
# yum install -y virt-install qemu-kvm qemu-img edk2-ovmf sevctl
```

4. Add boot parameters `mem_encrypt=on kvm_amd.sev=1` as shown below

```
[root@sev ~]# grubby --args="mem_encrypt=on kvm_amd.sev=1" --update-kernel
ALL
[root@sev ~]#
[root@sev ~]# grubby --info DEFAULT
```

```
[root@sev ~]# grubby --args="mem_encrypt=on kvm_amd.sev=1" --update-kernel ALL
[root@sev ~]#
[root@sev ~]# grubby --info DEFAULT
index=3
kernel="/boot/vmlinuz-5.14.0-284.11.1.el9_2.x86_64"
args="ro crashkernel=1G-4G:192M,4G-64G:256M,64G-:512M resume=/dev/mapper/rhel-swap rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap rhgb quiet $tuned_param
s mem_encrypt=on kvm_amd.sev=1"
root="/dev/mapper/rhel-root"
initrd="/boot/initramfs-5.14.0-284.11.1.el9_2.x86_64.img $tuned_initrd"
title="Red Hat Enterprise Linux (5.14.0-284.11.1.el9_2.x86_64) 9.2 (Plow)"
id="b6099645bc7c448a9c4807f81bbb0b1d-5.14.0-284.11.1.el9_2.x86_64"
[root@sev ~]#
```

Figure 9. Add boot parameters

5. Restart the host OS to active the SEV
6. Create SEV guest VM using the following command:

```
[rootsme-sev ~]# virt-install --name SEV \
> --machine q35 \
> --boot uefi \
> --launchsecurity sev,policy=0x1 \
> --memtune hard_limit=4563402 \
> --memory 4096 \
> --disk size=30 \
> --autoconsole text \
> --location /mnt/RHEL-9.2.0-20230414.17-x86_64-dvd1. iso \
> --install kernel_args="console=ttyS0"
```

Figure 10. Create a virtual machine

The install command includes the `--launchSecurity` parameter which allows you to specify the guest policy. The guest policy is specified as a hexadecimal value, the hex representation of the binary policy flags as listed in Table 2 below. As highlighted in red in the above figure, the policy is `policy=0x1` which means the guest VM will enable SEV in non-debug mode (bit 0 set to 1, as shown in Table 2).

Table 2. Guest Policy

Offset	Bit(s)	Name	Description
000h	**0**	**NODBG**	**Non-debug**
	1	NOKS	Not share the keys between guests
	2	ES	Enable SEV-ES
	3	NOSEND	Cannot send the guest to the other platform
	4	DOMAIN	Only transmit the guest to the platform in the domain
	5	SEV	Cannot transmit the guest to a platform without SEV capacity
	6-15	Reserved	Should be zero
002h	16-23	APL MAJOR	Cannot transmit the guest to a platform with lower firmware version
003h	24-32	API MIN	

To verify that SEV is enabled on the host, the parameter of the module kvm_amd should be “Y” or “1” as shown below:

```
[root@sme-sev ~]# cat /sys/module/kvm_amd/parameters/sev
Y
[root@sme-sev ~]#
```

To verify that SEV is enabled in the guest VM, the dmesg log should show the SEV support information:

```
[root@sev-guest ~]# dmesg | grep SEV
[    0.100517] Memory Encryption Features active: AMD SEV
[root@sev-guest ~]#
```

Enabling SEV-ES

Follows these steps to enable and verify SEV-ES:

1. To use SEV-ES, you need to enable it in UEFI.
   One method is to use F1 at boot to enter System Setup, go to the Processor and Memory sections. In the Processors section, set SVM Mode to Enable.

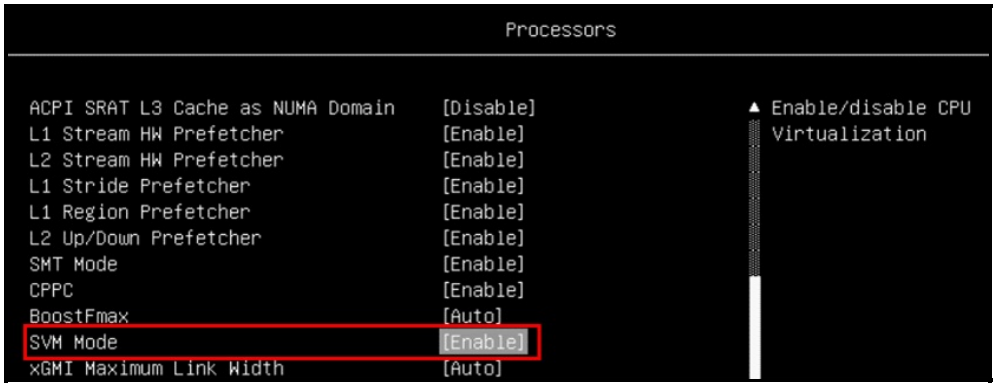

Figure 11. Processor settings in System Setup

In the Memory section, enable these items as shown in the figure below.

- SMEE: Enable
- SEV-ES ASID Count: AUTO
- SEV-ES ASID Space Limit Control: Manual
- SEV-ES ASID Space Limit: 10
- SEV Control: Enable

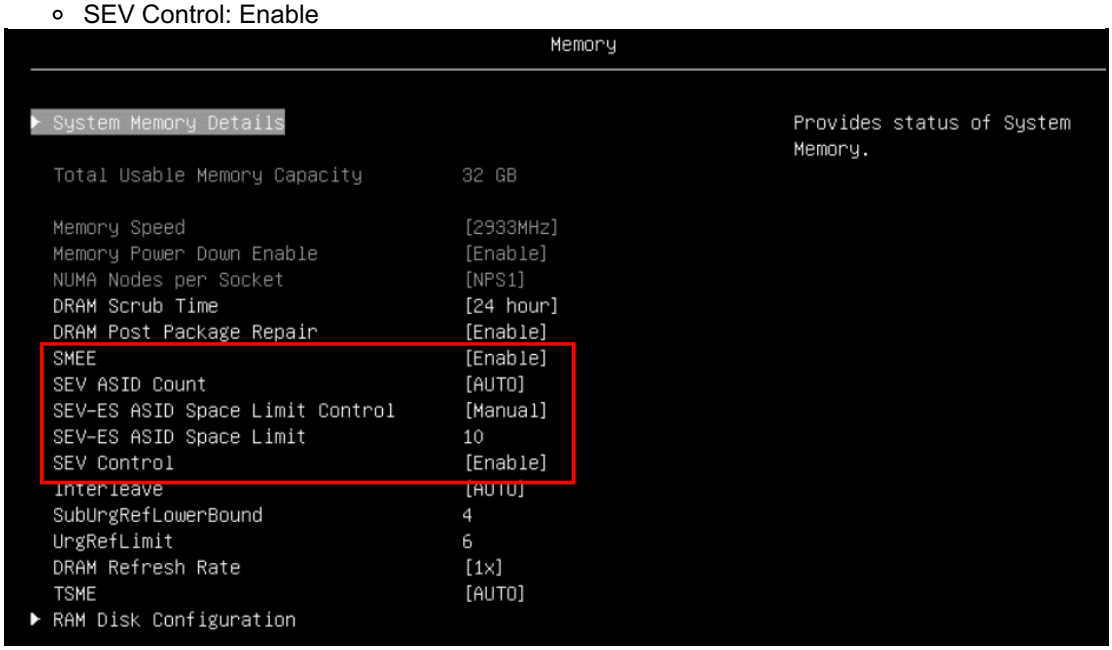

Figure 12. Memory settings in System Setup

Alternatively, you can use the OneCLI command line tool using the following command to build the configuration flie:

```
[root@sme-sev ~]# cat > sev-es.cfg << EOF
> set Memory.SMEE Enable
> set Memory.SEVASIDCount AUTO
> set Memory.SEV-ESASIDSpaceLimitControl Manual
> set Memory.SEV-ESASIDSpaceLimit 10
> set Memory.SEVControl Enable
> set Processors.SVMMode Enable
> EOF
[root@sme-sev ~]#
```

You then enable SEV-ES via a OneCLI command with the configuration file (sev-es.cfg):

```
./lnvgy_utl_lxceb_onecli01l-4.3.0_linux_x86-64.bin config batch --file sev
-es.cfg
```

```
[root@sme-sev ~]# ./lnvgy_utl_lxceb_onecli01l-4.3.0_linux_x86-64.bin config batch --file sev-es.cfg
Extracting...
Executing...


Start to connect current machine's BMC inband to apply config batch
Connected to BMC at IP address fe80::0a94:efff:feaa:94a4%enp35s0f3u1u6 by REDFISH
Invoking BATCH command ...

Processing....

All commands executed successfully !
Changes completed successfully, but these changes will not take effect until next restart the system.
Succeed.
[root@sme-sev ~]#
```

Figure 13. OneCLI command to enable SEV-ES

2. Restart the server to apply the change.
3. In Linux, install the required packages using the following command:

```
# yum install -y virt-install qemu-kvm qemu-img edk2-ovmf sevctl
```

4. Add boot parameters `mem_encrypt=on kvm_amd.sev=1 kvm_amd.sev_es=1` as shown below

```
[root@sme-sev ~]# grubby --args="mem_encrypt=on kvm_amd.sev kvm_amd.sev_es
=l" --update-kernel ALL
[root@sme-sev ~]#
[root@sme-sev ~]# grubby --info DEFAULT
```

```
[root@sme-sev ~]# grubby --args="mem_encrypt=on kvm_amd.sev=1 kvm_amd.sev_es=1" --update-kernel ALL
[root@sme-sev ~]#
[root@sme-sev ~]# grubby --info DEFAULT
index=3
kernel="/boot/vmlinuz-5.14.0-284.11.1.el9_2.x86_64"
args="ro crashkernel=1G-4G:192M,4G-64G:256M,64G-:512M resume=/dev/mapper/rhel-swap rd.lvm.lv=rhel/root rd.lvm.lv=
rhel/swap rhgb quiet $tuned_params mem_encrypt=on kvm_amd.sev=1 kvm_amd.sev_es=1"
root="/dev/mapper/rhel-root"
initrd="/boot/initramfs-5.14.0-284.11.1.el9_2.x86_64.img $tuned_initrd"
title="Red Hat Enterprise Linux (5.14.0-284.11.1.el9_2.x86_64) 9.2 (Plow)"
id="b6099645bc7c448a9c4807f81bbb0b1d-5.14.0-284.11.1.el9_2.x86_64"
[root@sme-sev ~]#
```

Figure 14. Add boot parameters

5. Restart the host OS to active the SEV-ES
6. Create SEV guest VM using the following command:

```
[root@sme-sev ~]# virt-install --name SEV-ES \
> --machine q35 \
> --boot uefi \
> --launchsecurity sev,policy=0x5 \
> --memtune hard_limit=4563402 \
> --memory 4096 \
> --disk size=30 \
> --autoconsole text \
> --location /mnt/RHEL-9.2.0-20230414.17-x86_64-dvd1.iso \
> --install kernel_args="console=ttyS0"
```

```
[root@sme-sev ~]# virt-install --name SEV-ES \
> --machine q35 \
> --boot uefi \
> --launchSecurity sev,policy=0x5 \
> --memtune hard_limit=4563402 \
> --memory 4096 \
> --disk size=30 \
> --autoconsole text \
> --location /mnt/RHEL-9.2.0-20230414.17-x86_64-dvd1.iso \
> --install kernel_args="console=ttyS0"

Starting install...
Retrieving 'vmlinuz'                                              |  12 MB  00:00:00
Retrieving 'initrd.img'                                           |  96 MB  00:00:00
Allocating 'SEV-ES.qcow2'                                         |  30 GB  00:00:03
Creating domain...                                                |         00:00:00
Running text console command: virsh --connect qemu:///system console SEV-ES
Connected to domain 'SEV-ES'
Escape character is ^] (Ctrl + ])
```

Figure 15. Create a virtual machine

The install command includes the `--launchSecurity` parameter which allows you to specify the guest policy. The guest policy is specified as a hexadecimal value, the hex representation of the binary policy flags as listed in the table below. As highlighted in red in the above figure, the policy is `policy=0x5` which means (converting 0x5 hex to 0101 binary), the guest VM will enable SEV in non-debug mode (bit 0 set to 1, as shown in the table) and with SEV-ES enabled (bit 2 set to 1, as shown in the table).

Table 3. Guest Policy

Offset	Bit(s)	Name	Description
000h	**0**	**NODBG**	**Non-debug**
	1	NOKS	Not share the keys between guests
	2	**ES**	**Enable SEV-ES**
	3	NOSEND	Cannot send the guest to the other platform
	4	DOMAIN	Only transmit the guest to the platform in the domain
	5	SEV	Cannot transmit the guest to a platform without SEV capacity
	6-15	Reserved	Should be zero
002h	16-23	APL MAJOR	Cannot transmit the guest to a platform with lower firmware version
003h	24-32	API MIN	

To verify that SEV is enabled on the host, the parameter of the module kvm_amd should be “Y” or “1” as shown below:

```
[root@sme-sev ~]# cat /sys/module/kvm_amd/parameters/sev_es
Y
[root@sme-sev ~]#
```

To verify that SEV is enabled in the guest VM, the dmesg log should show the SEV support information:

```
[root@seves-guest ~]# dmesg | grep SEV-ES
[    0.211789] Memory Encryption Features active: AMD SEV SEV-ES
[root@seves-guest ~]#
```

Special considerations

The following are limitations regarding the use of SME and SEV technologies:

- For some 32-bit legacy devices, it cannot issue DMA to encrypt memory directly, thus needs IOMMU (Input–Output Memory Management Unit) to re-map device request addresses with the C-bit set.
- Full cache flashing is a must to ensure all data has been written to DRAM before accessing a page via a different c-bit.
- SEV and SEV-ES VMs cannot be compatible with Secure Boot.

Resources

For more information, see these resources:

- AMD Memory Encryption
  https://www.amd.com/content/dam/amd/en/documents/epyc-business-docs/white-papers/memory-encryption-white-paper.pdf
- Protecting VM Register State with SEV-ES
  https://www.amd.com/content/dam/amd/en/documents/epyc-business-docs/white-papers/Protecting-VM-Register-State-with-SEV-ES.pdf
- SEV on Github
  https://github.com/AMDESE/AMDSEV
- Secure Encrypted Virtualization API
  https://www.amd.com/content/dam/amd/en/documents/epyc-technical-docs/programmer-references/55766_SEV-KM_API_Specification.pdf
- AMD Secure Encrypted Virtualization (AMD-SEV) Guide under SLES 15 SP5
  https://documentation.suse.com/sles/15-SP5/html/SLES-amd-sev/article-amd-sev.html
- Launch security with AMD SEV
  https://libvirt.org/kbase/launch_security_sev.html

Author

Song Shang is a Linux Engineer in Lenovo Infrastructure Solutions Group, based in TianJin, China.

Thanks to the following people for their contributions to this project:

- David Watts, Lenovo Press
- Adrian Huang, Lenovo Linux Engineer
- Gary Cudak, Lenovo Lead Architect

Related product families

Product families related to this document are the following:

- Processors

Notices

Lenovo may not offer the products, services, or features discussed in this document in all countries. Consult your local Lenovo representative for information on the products and services currently available in your area. Any reference to a Lenovo product, program, or service is not intended to state or imply that only that Lenovo product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any Lenovo intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any other product, program, or service. Lenovo may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

Lenovo (United States), Inc.
8001 Development Drive
Morrisville, NC 27560
U.S.A.
Attention: Lenovo Director of Licensing

LENOVO PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Lenovo may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

The products described in this document are not intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Lenovo product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Lenovo or third parties. All information contained in this document was obtained in specific environments and is presented as an illustration. The result obtained in other operating environments may vary. Lenovo may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any references in this publication to non-Lenovo Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Lenovo product, and use of those Web sites is at your own risk. Any performance data contained herein was determined in a controlled environment. Therefore, the result obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

This document, LP1894, was created or updated on February 20, 2024.

Send us your comments in one of the following ways:

- Use the online Contact us review form found at:
  https://lenovopress.lenovo.com/LP1894
- Send your comments in an e-mail to:
  comments@lenovopress.com

This document is available online at https://lenovopress.lenovo.com/LP1894.

Trademarks

Lenovo and the Lenovo logo are trademarks or registered trademarks of Lenovo in the United States, other countries, or both. A current list of Lenovo trademarks is available on the Web at https://www.lenovo.com/us/en/legal/copytrade/.

The following terms are trademarks of Lenovo in the United States, other countries, or both:
Lenovo®
ThinkSystem®

The following terms are trademarks of other companies:

AMD, AMD EPYC™, and AMD-V™ are trademarks of Advanced Micro Devices, Inc.

Linux® is the trademark of Linus Torvalds in the U.S. and other countries.

Other company, product, or service names may be trademarks or service marks of others.