2024 NTU Virtual Machine HW3 Writeup

StudentID : R12922054

[Method 1. Using QUMU Monitor]

Step-by-Step Process

1. First, enter the host VM through the following command

```
./run-pkvm.sh -k pkvm_host_Image -i cloud.img
```

2. Next, set up the ssh connection of the host VM (this part was not recorded in the video because it takes too much time), and execute `dhclient`
3. Next, modify run-guest.sh and add the following command in it. The purpose is to connect to `qemu monitor` through `telnet` (the added location is shown in the video): `telnet` is the method learned through ChatGPT. You can also use `nc`.

```
-monitor telnet:127.0.0.1:7777,server,nowait
```

4. Execute the following command to enter the guest VM, and open another window ssh to enter the host VM (hereinafter called window B, the original window is called window A)

```
./run-guest.sh -k Image -i cloud-inner.img
```

5. Enter the following command in window B to enter `qemu monitor` through `telnet`

```
telnet localhost 7777(port num you set up in run-guest.sh)
```

6. Execute the following commands in `qemu monitor` to get the memory information

```
info mtree
```

7. Find the memory section named `mach-virt.ram`, and then access it through the following command.

```
xp /20x 0x40000000
```

The resulting behaviors of pKVM's host

And, in window A, you will find that the system will trigger a `segmentation fault` to protect the guest VM's memory from being accessed by the host process, as shown in Figure 1

```
root@ubuntu:~# ./run-guest.sh: line 98:   593 Segmentation fault      (core dumped) ./qemu-system-aarch
64 -nographic -machine virt -m ${MEMSIZE} -cpu host -smp ${SMP} -enable-kvm -kernel ${KERNEL} ${DTB} -d
rive if=none,file=$FS,id=vda,cache=none,format=raw -device virtio-blk-pci,drive=vda,indirect_desc=false
 -display none -serial $CONSOLE -append "console=ttyAMA0 root=/dev/vda rw $CMDLINE" -netdev user,id=net
0,hostfwd=tcp::2222-:22 -monitor telnet:127.0.0.1:7777,server,nowait
./run-guest.sh: line 99: -device: command not found
```

Figure 1: segmentation fault.

[Method 2. Using GDB]

Step-by-Step Process

1. First, enter the host VM through the following command

```
./run-pkvm.sh -k pkvm_host_Image -i cloud.img
```

2. Next, set up the ssh connection of the host VM (this part was not recorded in the video because it takes too much time), and execute `dhclient`
3. Next, modify run-guest.sh and add the following command in it. The purpose is to connect to `qemu monitor` through `telnet` (the added location is shown in the video): `telnet` is the method learned through ChatGPT. You can also use `nc`.

```
-monitor telnet:127.0.0.1:7777,server,nowait
```

4. Execute the following command to enter the guest VM, and open another window ssh to enter the host VM (hereinafter called window B, the original window is called window A)

```
./run-guest.sh -k Image -i cloud-inner.img
```

5. Enter the following command in window B to enter `qemu monitor` through `telnet`

```
telnet localhost 7777(port num you set up in run-guest.sh)
```

6. Execute the following commands in `qemu monitor` to get the memory information

```
info mtree
```

7. Find the memory section named `mach-virt.ram`, and then execute the following commands in `qemu monitor` to get the guest VM's host virtual address. This command is known from this website[1].

```
gpa2hva 0x40000000(address you want to translate)
```

8. Next, enter the following command to exit `qemu monitor`:
   First enter the command mode of `telnet`

```
Ctrl + ]
```

   And close `telnet` to leave `qemu monitor`

```
close
```

9. Use the following command to find the PID of the QEMU process

```
pgrep -l qemu
```

10. After finding the corresponding PID, use gdb to access the process, as follows

```
gdb -p PID
```

11. After entering gdb, execute the following instructions

```
x /20x 0xffff57e00000(the address you get from gpa2hva command)
```

The resulting behaviors of pKVM's host

And, in window B, you will find that GDB triggers a "segmentation fault" to protect the guest VM's memory from being accessed by the host process, as shown in Figure 2. In window A, the guest VM shuts down directly due to an error, as shown in Figure 3.

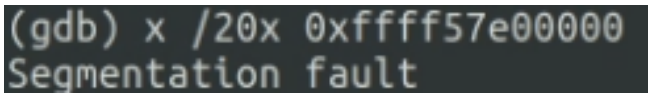

Figure 2: segmentation fault showing in GDB.

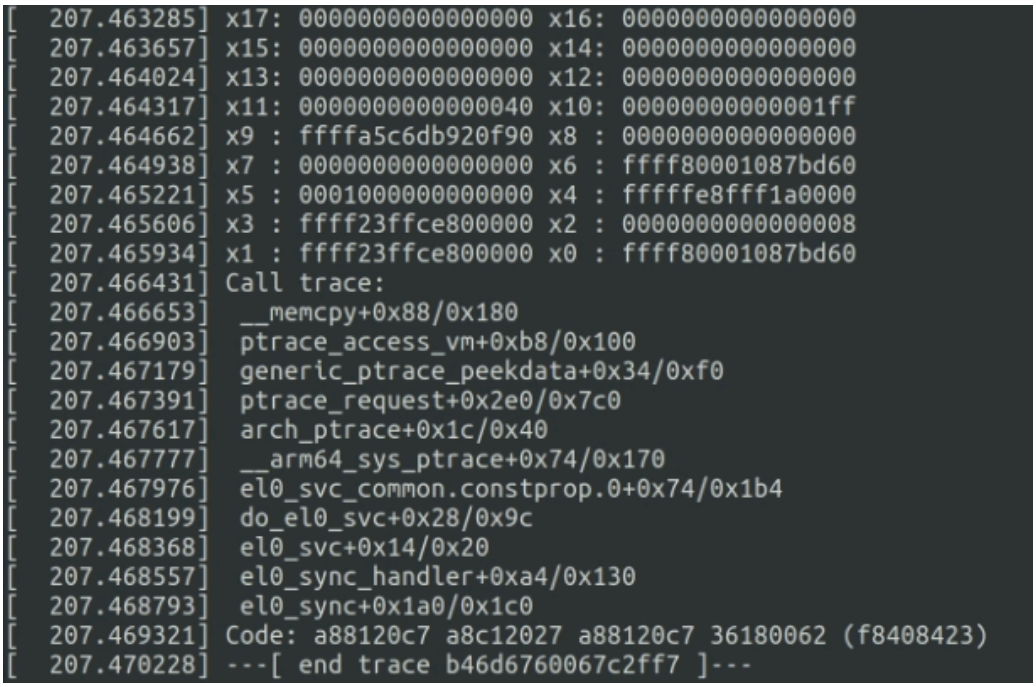

Figure 3: guest VM shuts down automatically due to error.

Discussion

In fact, it is not necessary to enter qemu monitor first and execute gpa2hva to get the address we want. We can directly enter gdb and list all memory mappings through the following command, and then find the section with offset 512MB (0x20000000). After finding it, and accessing it, we can get the same result.

```
info proc mappings
```

References

[1] "QEMU-Monitor-CMD," [Online]. Available: https://qemu-project.gitlab.io/qemu/system/monitor.html