VM HW3 B10902078

HackMD Link

Steps

- First, I added this flag `-monitor telnet:127.0.0.1:1234,server,nowait \` in `./qemu-system-aarch64 -nographic ...` inside `run-guest.sh`, so I can open qemu-monitor.
- Then, I launch the guest VM, and use another terminal to run `telnet localhost 1234` (in kvm) to connect to the qemu-monitor session.
- Then, I used three methods to test the pkvm protection (since method1 & method2 both uses qemu-monitor, so I added another method)

Method 1:

In qemu monitor, I use the `dump-guest-memory` command, which attempts to dump the guest memory. The output is

```
(qemu) dump-guest-memory guest-memory
KVM_SET_DEVICE_ATTR failed: Group 4 attr 0x0000000000000001: Bad address
Error: dump: failed to save memory: Bad address
```

This confirms the pkvm protection of the guest VM's memory, as we are not allowed to dump the guest memory out.

Method 2:

In qemu monitor, I use the `info mtree` command, which will display the memory tree of the GPA. The output consist of one special line (`mach-virt.ram`)

```
0000000040000000-000000005fffffff (prio 0, ram): mach-virt.ram
```

which contains the RAM of the guest VM that is allocated by QEMU through `mmap()`.

Then, I try to access this GPA using `xp/10wx 0x40000000`. Due to the pkvm protection, the host linux should not be able to access the guest VM's memory, thus, the qemu monitor will be automatically killed. Since the guest VM is associated with the same process of QEMU, so the guest VM will also be killed, showing a `segmentation fault (core dumped)` error.

Method 3:

In qemu monitor, I will also get the GPA of `mach-virt.ram` through `info mtree`, and use the `gpa2hva` command to translate GPA to HVA.

```
(qemu) gpa2hva 0x40000000
Host virtual address for 0x40000000 (mach-virt.ram) is 0xffff6fe00000
```

Let's just assume the corresponding HVA is `0xffff6fe00000` (it may not be the same everytime). I will first get the PID of the qemu process through `ps aux | grep qemu`, and verify this HVA exists in `/proc/[PID]/maps`. The output of this file is very long, so I will not display it here (check video).

Then, I run the program (`sudo ./memread [PID] [HVA]`) below to read the HVA from the `/proc/[PID]/mem` file. When I run this file in the host linux, it will result in `segmentation fault`, confirming the pkvm protection of the guest VM's memory.

```c
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdlib.h>

int main(int argc, char *argv[]){
        char buf[256];
        char path[64];
        sprintf(path, "/proc/%s/mem", argv[1]);
        int fd = open(path, O_RDONLY);
        lseek(fd, strtoull(argv[2], NULL, 16), SEEK_SET);
        read(fd, buf, 16);
        write(1, buf, 16);
        close(fd);
        return 0;
}
```