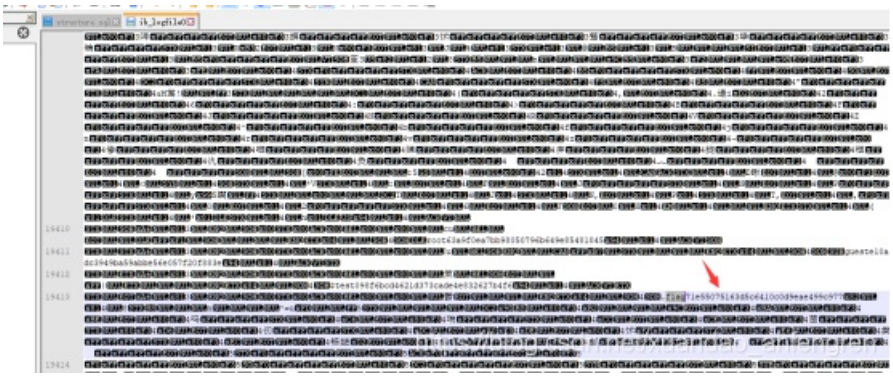
攻防世界misc高手进阶篇教程（5）

原创

锋刃科技 于 2020-05-28 19:05:53 发布 2665 收藏 5
文章标签： ctf攻防世界

Mysql

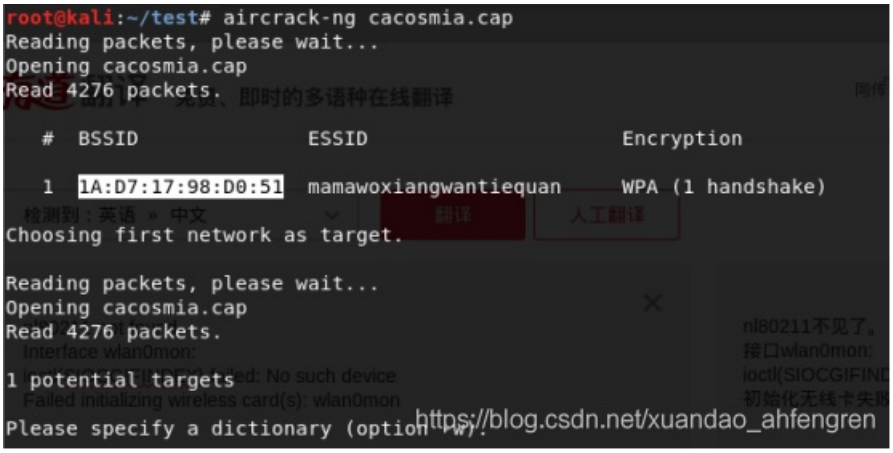
在mysql文件中ib_logfile0有flag



恶臭的数据包

Wireshark是看的有些懵

我们用aircrack-ng查看信息

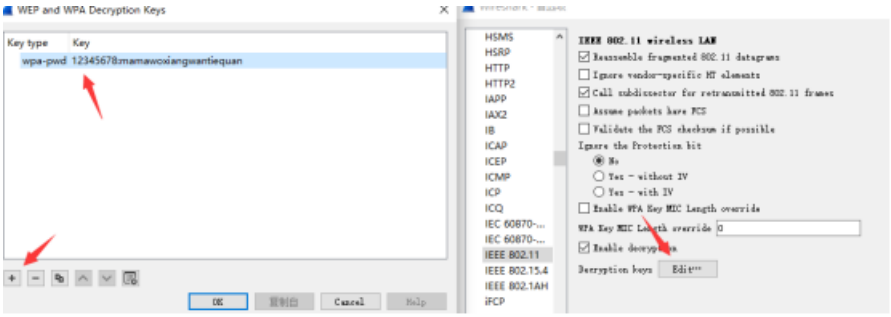


然后再破解WiFi密码

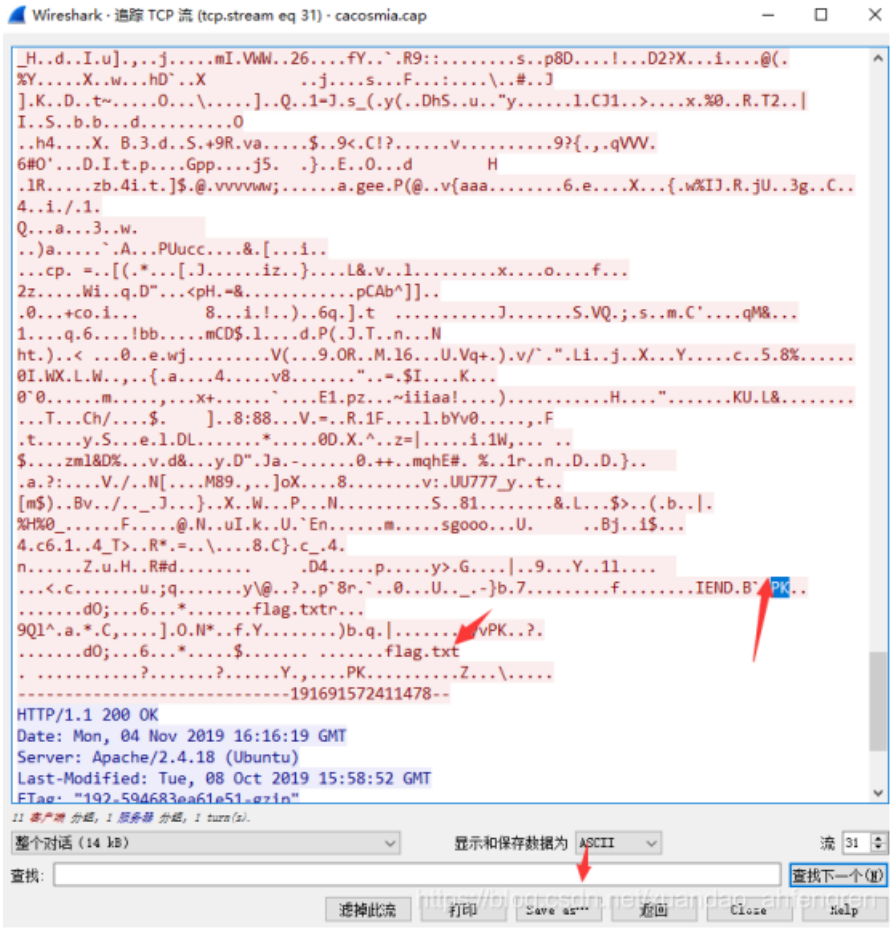
aircrack-ng cacosmia.cap -w /usr/share/wordlists/fasttrack.txt

```
root@kali:~/test# aircrack-ng cacosmia.cap -w /usr/share/wordlists/fasttrack.txt
Reading packets, please wait...
Opening cacosmia.cap
Read 4276 packets.

   #  BSSID              ESSID                     Encryption

   1  1A:D7:17:98:D0:51  mamawoxiangwantiequan     WPA (1 handshake)

Choosing first network as target.

Reading packets, please wait...
Opening cacosmia.cap
Read 4276 packets.

1 potential targets

                               Aircrack-ng 1.6

      [00:00:00] 191/222 keys tested (3049.96 k/s)

      Time left: 0 seconds                                      86.04%

                          KEY FOUND! [ 12345678 ]
```

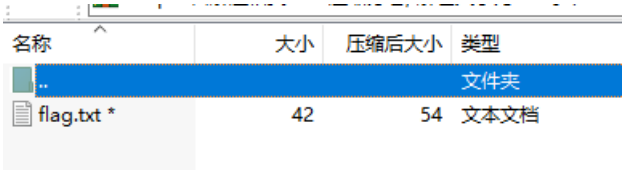

打开Wireshark 编辑->首选项->Protocols->IEEE 802.11



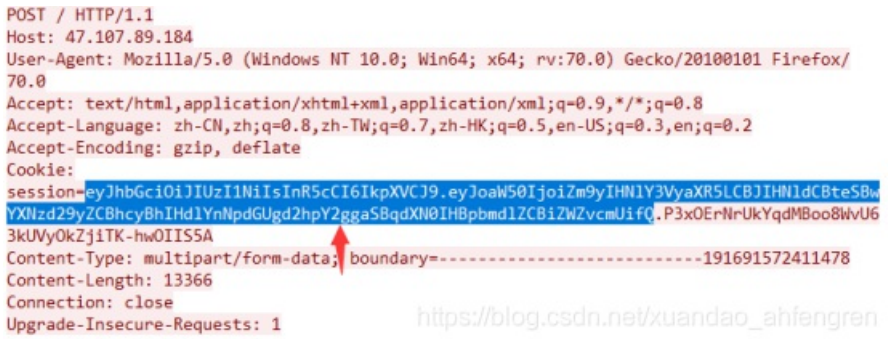
发现有个数据包里有PK就是zip，变成原始数据保存下来

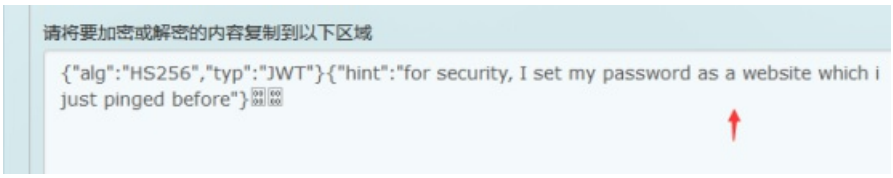


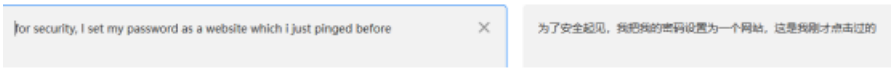
需要密码



发现cookie里有疑似base64加密的数据

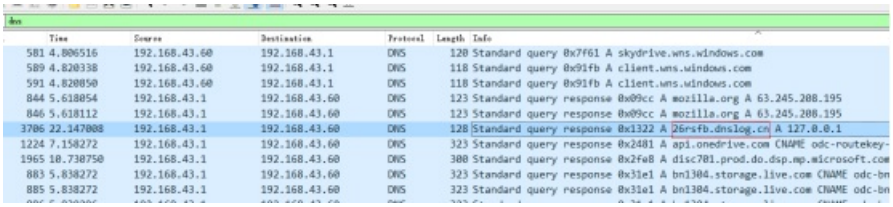


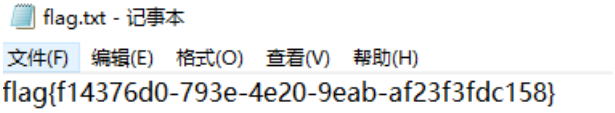




ping域名之前，一定要通过DNS来获取域名指向的ip，于是过滤DNS协议

发现26rsfb.dnslog.cn是对的





picture2

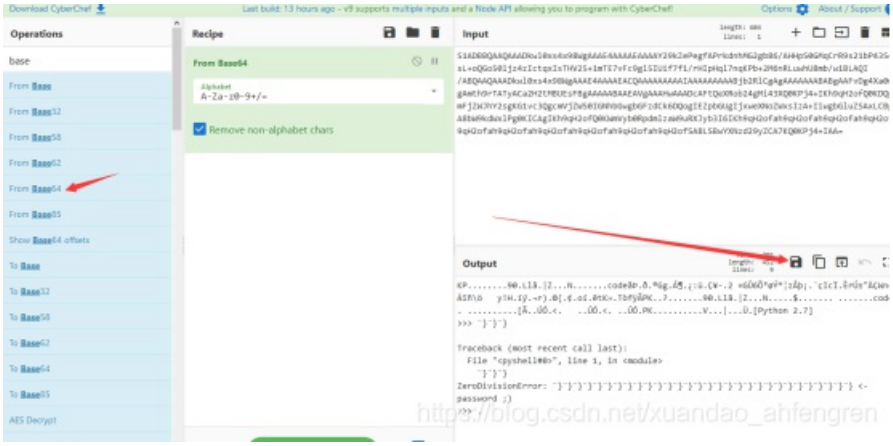
发现有zlib文件，分离出来

binwalk -e 直接分离

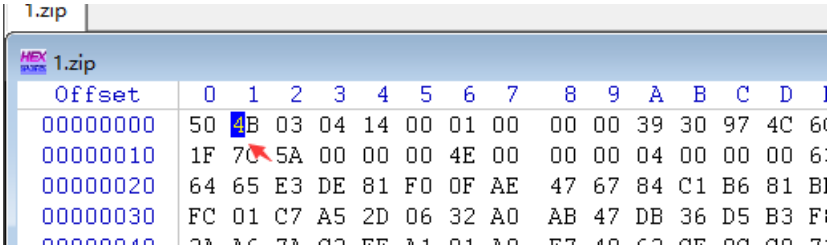
```
root@kali:~/test# binwalk e4103617b4a6476fb7aa8f862f2ee400.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
38884         0x97E4          Zlib compressed data, default compression
```

```
root@kali:~/test# binwalk -e e4103617b4a6476fb7aa8f862f2ee400.png

WARNING: The Python LZMA module could not be found. It is *strongly* recomme
 that you install this module for binwalk to provide proper LZMA identificat
and extraction results.

WARNING: The Python LZMA module could not be found. It is *strongly* recomme
 that you install this module for binwalk to provide proper LZMA identificat
and extraction results.

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
38884         0x97E4          Zlib compressed data, default compression
```


这文件是用base64加密的，我们直接解密后，后缀改成zip

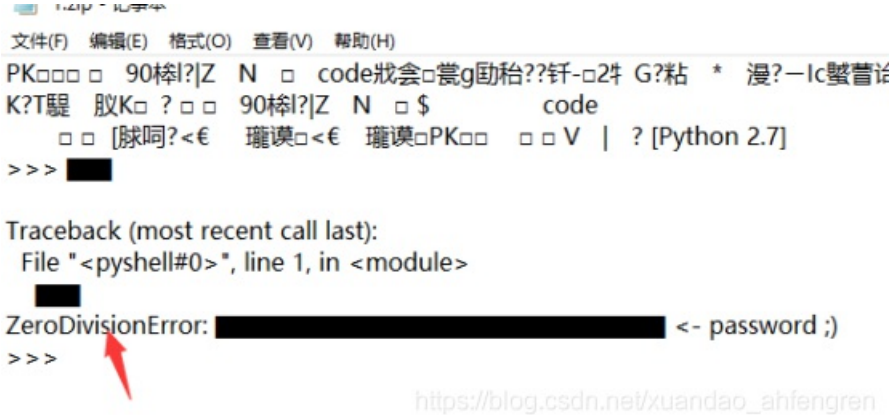


然后用winhex修改头为50、4B



解压密码是integer division or modulo by zero

因为这里说了错误，python2错误是integer division or modulo by zero



中间部分解码即可

Latlong

用kali发现这是音频文件

用multimon-ng直接获取flag

./multimon-ng -t raw -a AFSK1200 transmission.raw

multimon-ng (C) 1996/1997 by Tom Sailer HB9JNX/AE4WA (C) 2012-2014 by Elias

Oenal available demodulators: POCSAG512 POCSAG1200 POCSAG2400 FLEX EAS

UFSK1200 CLIPFSK FMSFSK AFSK1200 AFSK2400 AFSK2400_2 AFSK2400_3 HAPN4800

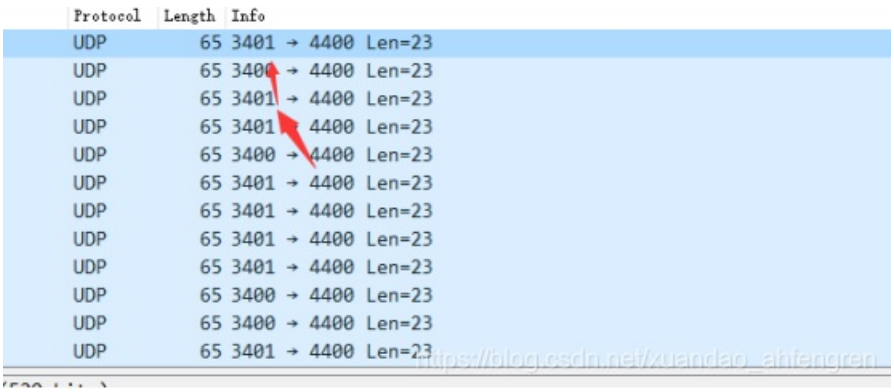
FSK9600 DTMF ZVEI1 ZVEI2 ZVEI3 DZVEI PZVEI EEA EIA CCIR MORSE_CW DUMPCSV SCOPE

Enabled demodulators: AFSK1200 AFSK1200: fm WDPX01-0 to APRS-0 UI pid=F0

!/;E'q/Sz'O /A=000000flag{f4ils4f3c0mms}

Hidden-Message

发现这里在0和0一直在变换

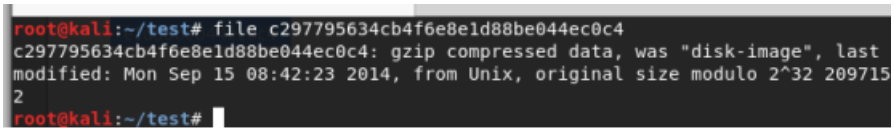


这样 前八个就为：**01001000** 即为大写“**H**”

最后flag就为：**Heisenberg**

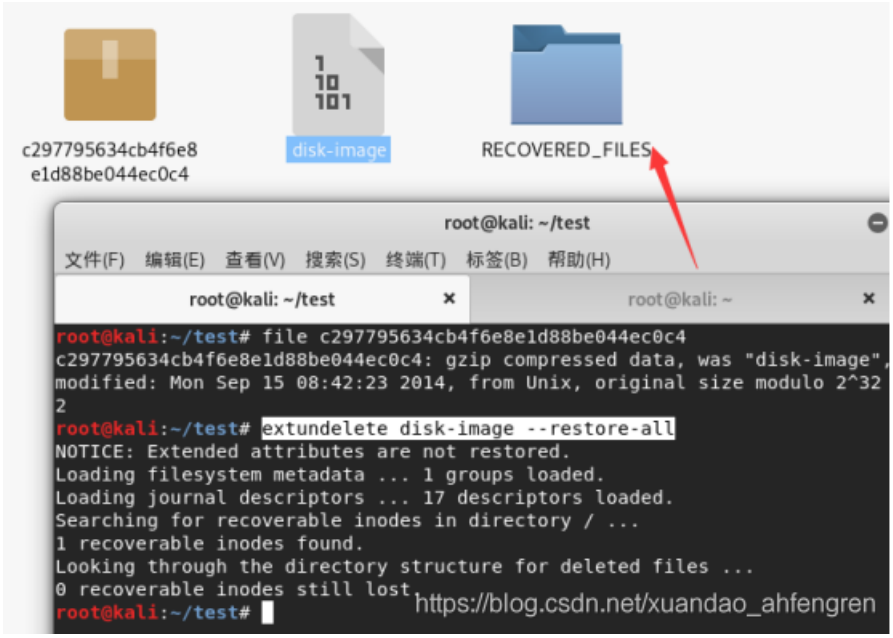
Recover-Deleted-File

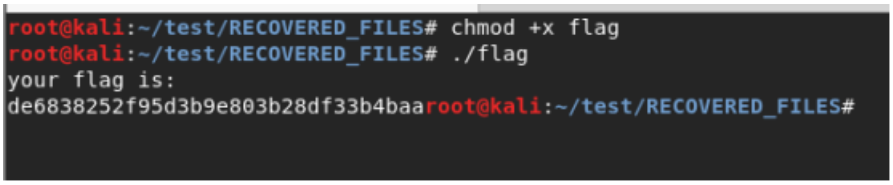
先改后缀再把文件解压出来



extundelete disk-image --restore-all

生成文件，里面flag加权限运行即可



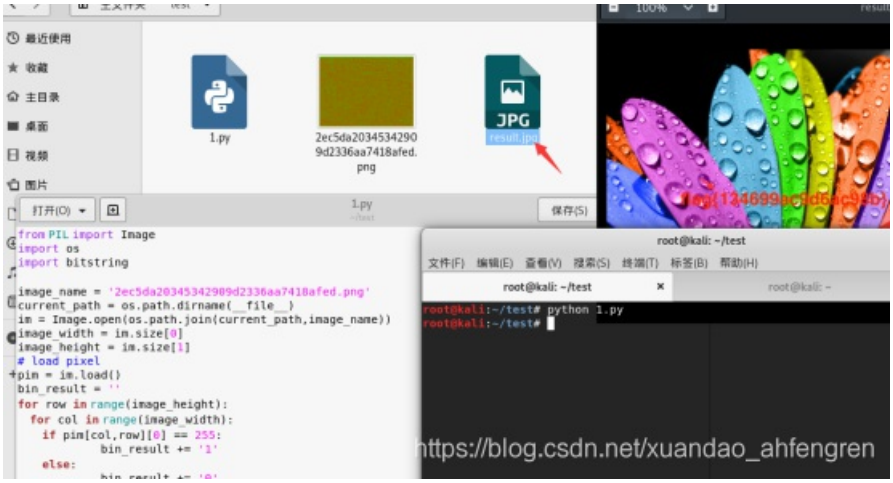


red_green

进行颜色识别转换为0和1

```
from PIL import Image
import os
import bitstring

image_name = '2ec5da20345342909d2336aa7418afed.png'
current_path = os.path.dirname(__file__)
im = Image.open(os.path.join(current_path,image_name))
image_width = im.size[0]
image_height = im.size[1]# load pixel
pim = im.load()
bin_result = ''for row in range(image_height):
    for col in range(image_width):
        if pim[col,row][0] == 255:
            bin_result += '1'
        else:
            bin_result += '0'
with open(os.path.join(current_path,'result.jpg'),'wb') as f:
    f.write(bitstring.BitArray(bin=bin_result).bytes)
```

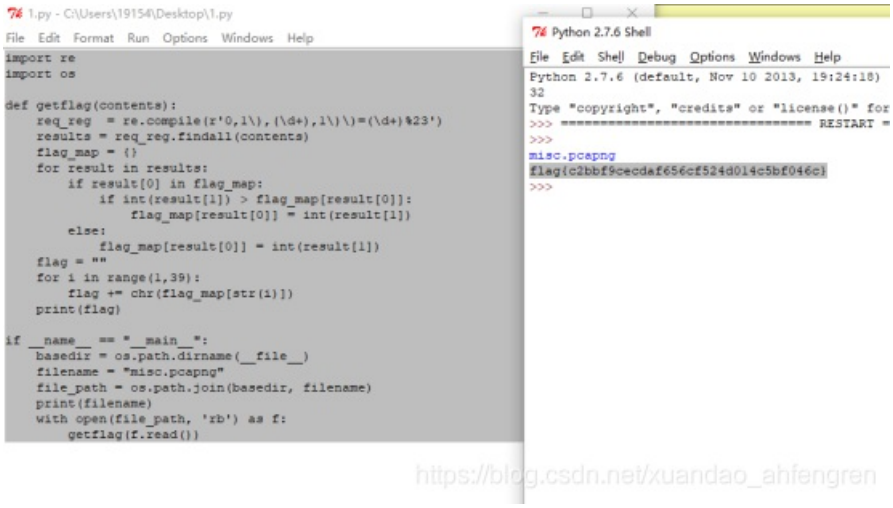
运行脚本就出来了



流量分析

通过=进行的sql注入，查看每个位置最后相等的ascii码即可

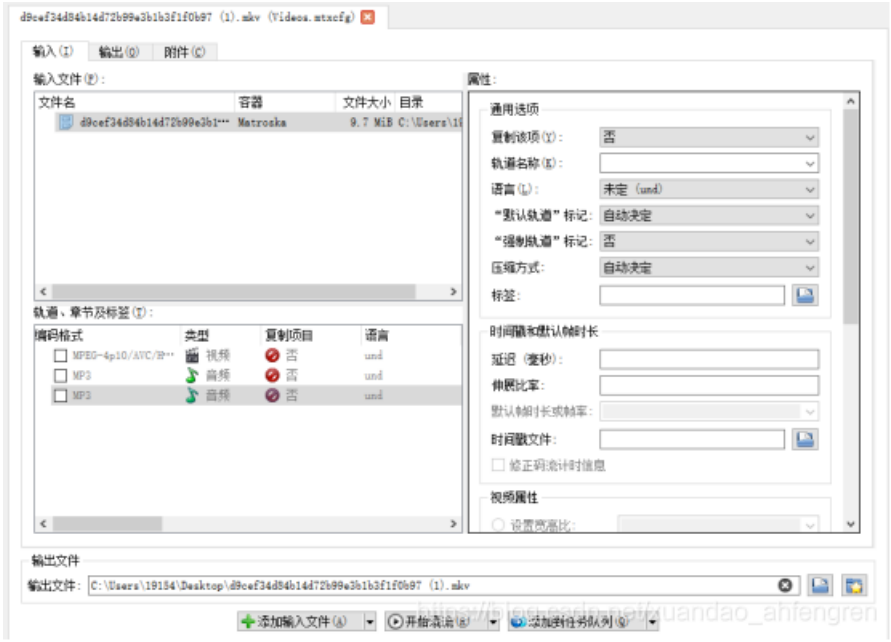
```
import re
import os

def getflag(contents):
    req_reg  = re.compile(r'0,1\),(\d+),1\)\)=(\d+)%23')
    results = req_reg.findall(contents)
    flag_map = {}
    for result in results:
        if result[0] in flag_map:
            if int(result[1]) > flag_map[result[0]]:
                flag_map[result[0]] = int(result[1])
        else:
            flag_map[result[0]] = int(result[1])
    flag = ""
    for i in range(1,39):
        flag += chr(flag_map[str(i)])
    print(flag)

if __name__ == "__main__":
    basedir = os.path.dirname(__file__)
    filename = "misc.pcapng"
    file_path = os.path.join(basedir, filename)
    print(filename)
    with open(file_path, 'rb') as f:
        getflag(f.read())
```



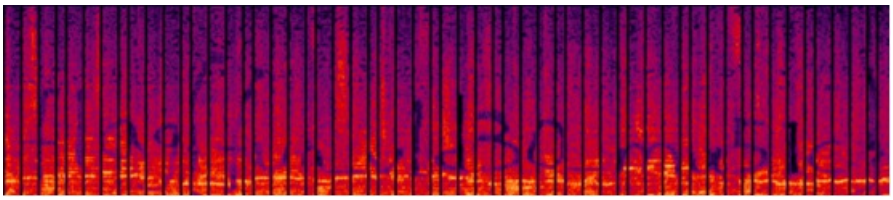
funny_video

播放视频可发现视频有多个音轨，并且两个音轨声音相似

用MKVToolnixPortable工具进行提取

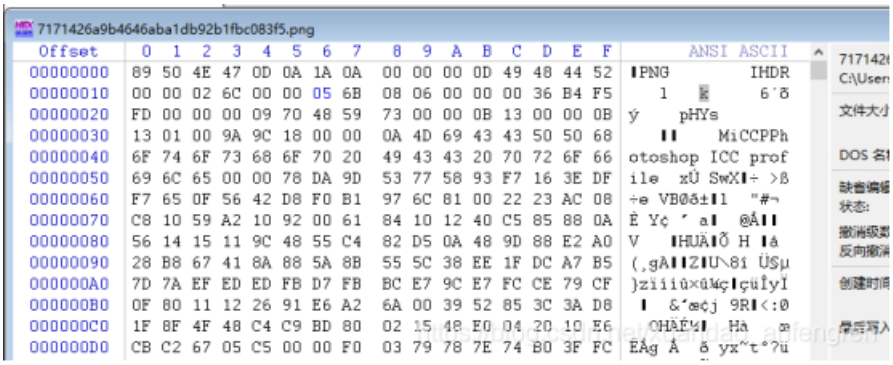


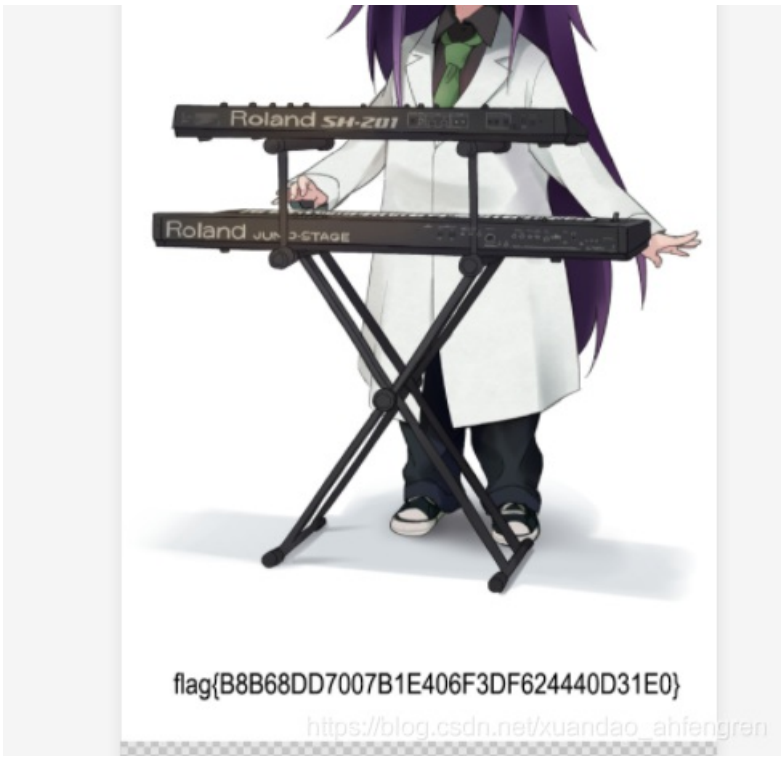
使用Audition打开提取的音频查看频谱，得到flag



normal_png

修改高度即可





flag{B8B68DD7007B1E406F3DF624440D31E0}

侧信道初探

If语句会增加复杂度，所消耗能量也增加，所以代表1不进行if语句的能量消耗就会少一些，所以代表0

flag:SCTF{0110111010}

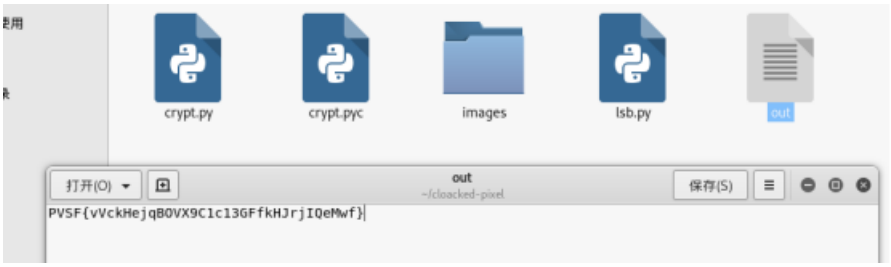
Keyword

先安装环境

git clone https://github.com/livz/cloacked-pixel.git

python lsb.py extract keyword.png out lovekfc

打开out后得到PVSF{vVckHejqBOVX9C1c13GFfkHJrjIQeMwf}



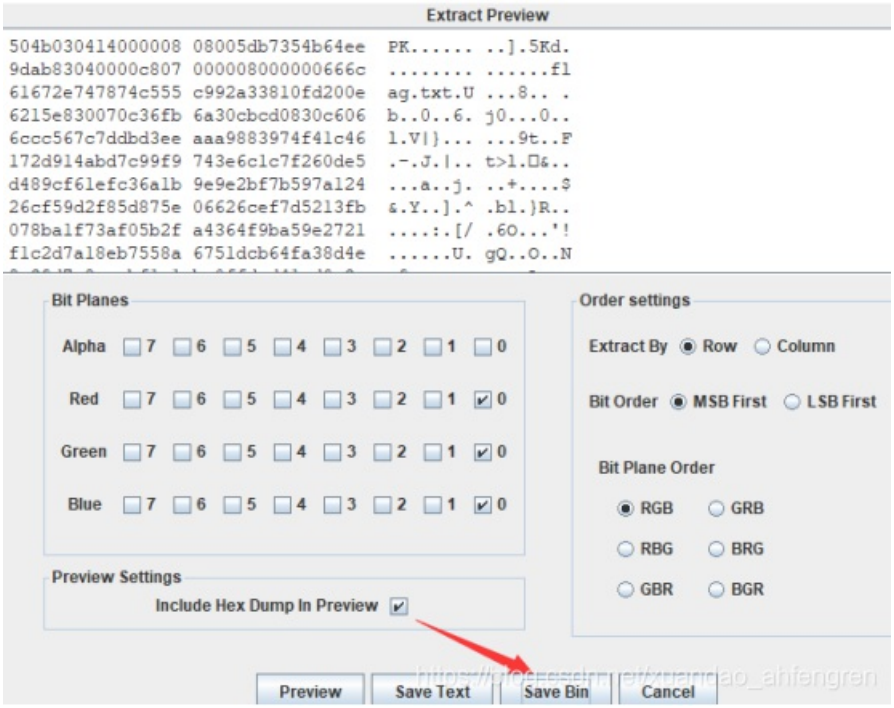
发现是 Nihilist 密码

解密后

flag QCTF{cCgeLdnrlBCX9G1g13KFfeLNsnMRdOwf}

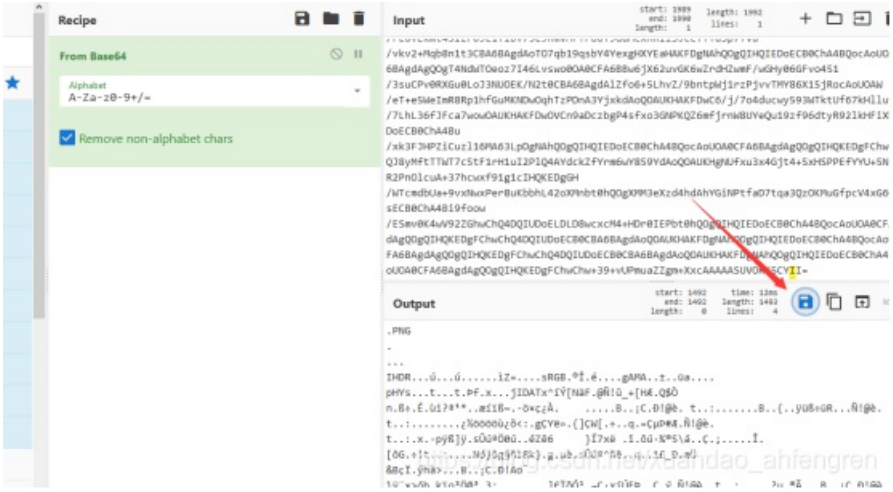
3-11

用Stegsolve.jar把压缩包解压



解压后发现base64加密的数据

保存为图片



FLAG{LSB_i5_SO_EASY}