

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: [cf](#) [攻防世界](#)

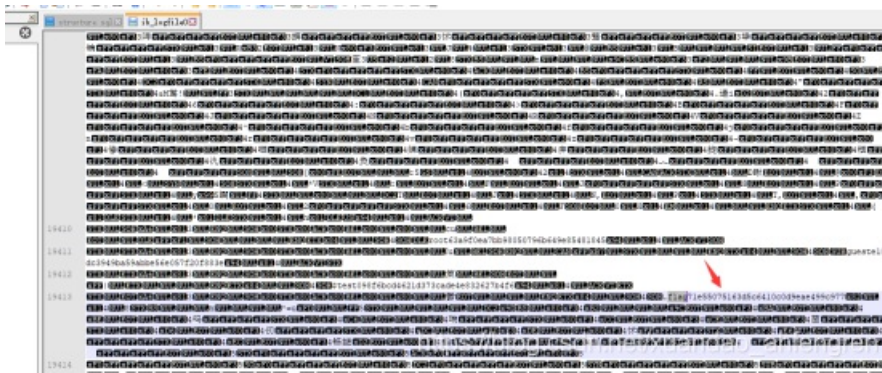
版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/xuandao_ahfengren/article/details/106411298

版权

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

```
root@kali:~/test# aircrack-ng cacosmia.cap
Reading packets, please wait...
Opening cacosmia.cap
Read 4276 packets.

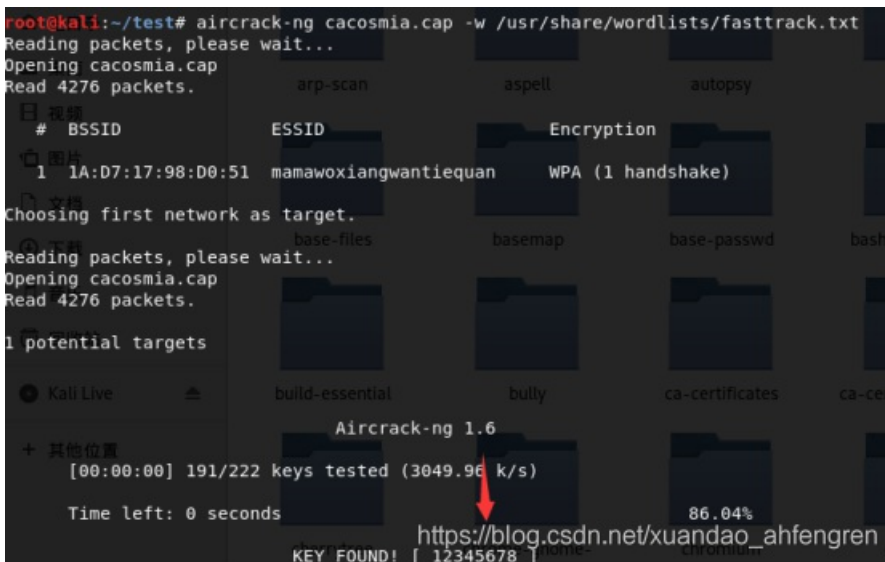
# BSSID      ESSID      Encryption
1 1A:D7:17:98:D0:51  mamawoxiangwantiequan  WPA (1 handshake)

Choosing first network as target.

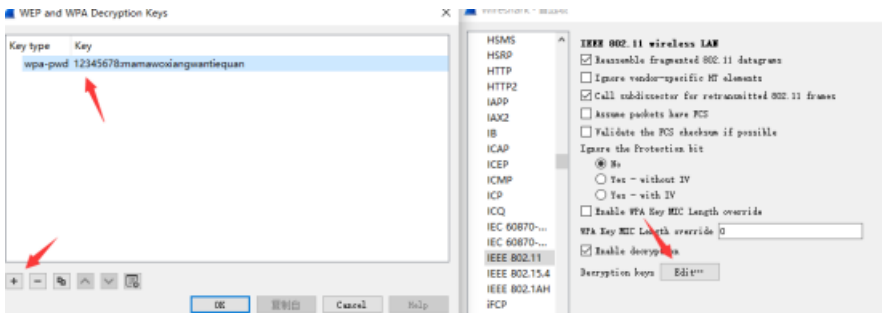
Reading packets, please wait...
Opening cacosmia.cap
Read 4276 packets.
Interface wlan0mon:
1 potential targets:
Failed initializing wireless card(s): wlan0mon
Please specify a dictionary (option -w):
```

然后再破解WiFi密码

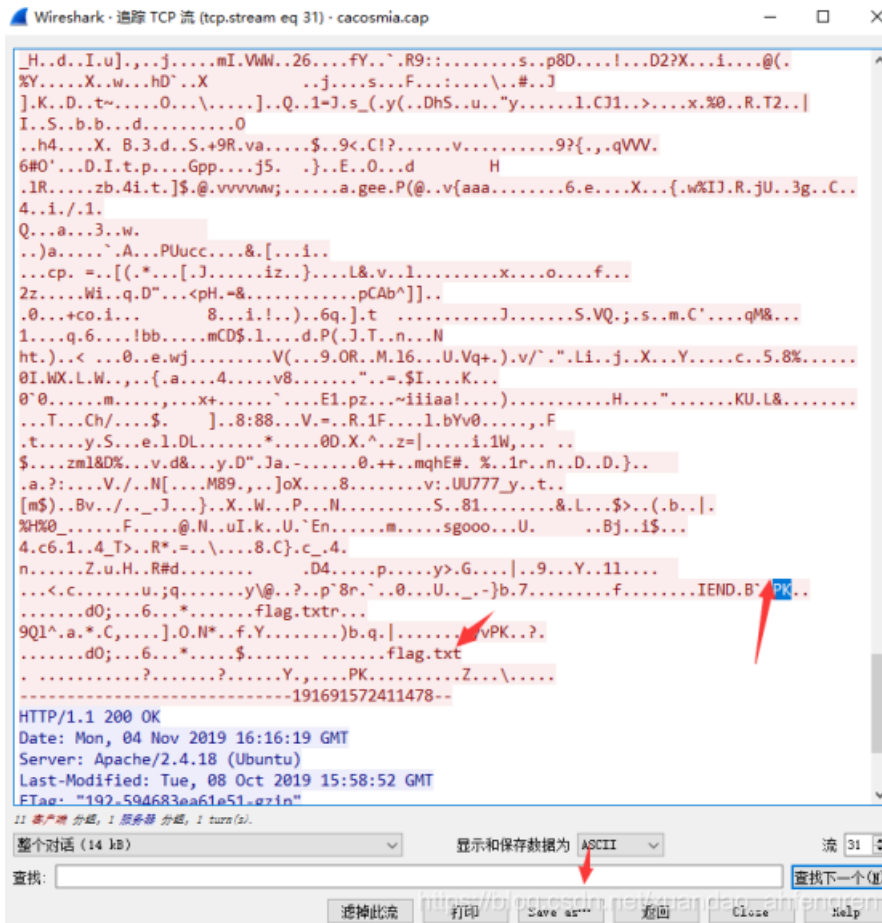
`aircrack-ng cacosmia.cap -w /usr/share/wordlists/fasttrack.txt`

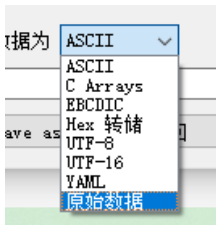


打开Wireshark 编辑->首选项->Protocols->IEEE 802.11



发现有个数据包里有PK就是zip，变成原始数据保存下来





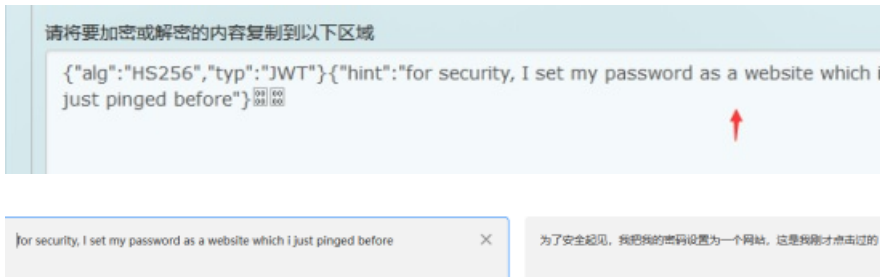
需要密码

名称	大小	压缩后大小	类型
..			文件夹
flag.txt *	42	54	文本文档

发现cookie里有疑似base64加密的数据

```
POST / HTTP/1.1
Host: 47.107.89.184
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:70.0) Gecko/20100101 Firefox/70.0
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
Cookie:
session=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0aW50IjoiZm9yIHNhY3VyaXR5LCBjIHh1dCBteSBwYXNzd29yZCBhcyBhIHdlYnNpdGUgd2hpY2ggaSBqdXN0IHhpbmdlZCBiZWVvcmluZi0uP3x0ErNrUkYqdMBoo8WvU63kUVyOkZj1TK-hw0IIS5A
Content-Type: multipart/form-data; boundary=-----191691572411478
Content-Length: 13366
Connection: close
Upgrade-Insecure-Requests: 1
```

https://blog.csdn.net/xuandao_ahfengren



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

Time	Source	Destination	Protocol	Length	Info
581 4.806516	192.168.43.60	192.168.43.1	DNS	128	Standard query 8x7f61 A skydrive.wns.windows.com
589 4.820338	192.168.43.60	192.168.43.1	DNS	118	Standard query 8x91fb A client.wns.windows.com
591 4.829850	192.168.43.60	192.168.43.1	DNS	118	Standard query 8x91fb A client.wns.windows.com
844 5.618054	192.168.43.1	192.168.43.60	DNS	123	Standard query response 8x09cc A mozilla.org A 63.245.208.195
846 5.618112	192.168.43.1	192.168.43.60	DNS	123	Standard query response 8x09cc A mozilla.org A 63.245.208.195
3706 22.147008	192.168.43.1	192.168.43.60	DNS	128	Standard query response 8x1322 A 26rsfb.dnslog.cn A 127.0.0.1
1224 7.158272	192.168.43.1	192.168.43.60	DNS	323	Standard query response 8x2401 A api.onedrive.com CNAME odc-routekey-
1965 10.730750	192.168.43.1	192.168.43.60	DNS	308	Standard query response 8x2fa8 A disc781.prod.do.dsp.mp.microsoft.com
883 5.838272	192.168.43.1	192.168.43.60	DNS	323	Standard query response 8x31e1 A bn1304.storage.live.com CNAME odc-bn
885 5.838272	192.168.43.1	192.168.43.60	DNS	323	Standard query response 8x31e1 A bn1304.storage.live.com CNAME odc-bn

flag.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

flag{f14376d0-793e-4e20-9eab-af23f3fdc158}

picture2

发现有zlib文件, 分离出来

binwalk -e 直接分离

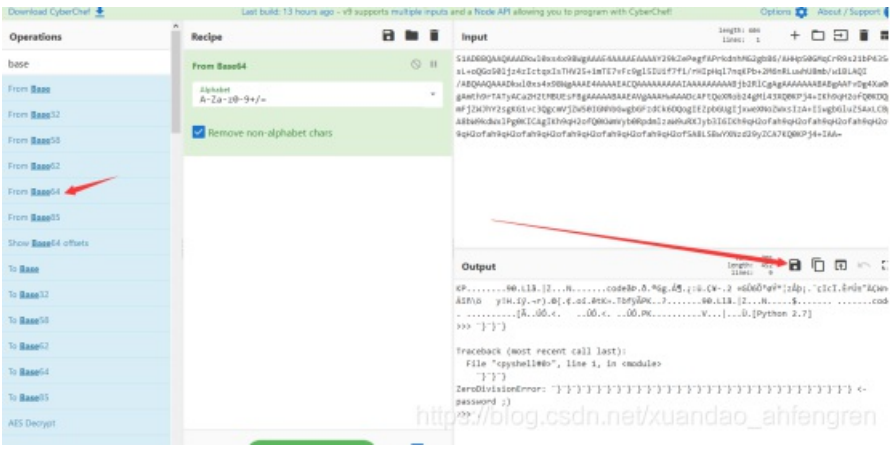
```
root@kali:~/test# binwalk e4103617b4a6476fb7aa8f862f2ee400.png
DECIMAL  HEXADECIMAL  DESCRIPTION
-----  -
0        0x0             png JPEG image data, JFIF standard 1.01
38884    0x97E4         Zlib compressed data, default compression
```

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

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E
00000000	50	4B	03	04	14	00	01	00	00	00	39	30	97	4C	60
00000010	1F	70	5A	00	00	00	4E	00	00	00	04	00	00	00	61
00000020	64	65	E3	DE	81	F0	0F	AE	47	67	84	C1	B6	81	B1
00000030	FC	01	C7	A5	2D	06	32	A0	AB	47	DB	36	D5	B3	F1
00000040	23	3C	73	02	FE	31	01	30	E7	40	C2	0E	00	00	71

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

Protocol	Length	Info
UDP	65	3401 → 4400 Len=23
UDP	65	3400 → 4400 Len=23
UDP	65	3401 → 4400 Len=23
UDP	65	3401 → 4400 Len=23
UDP	65	3400 → 4400 Len=23
UDP	65	3401 → 4400 Len=23
UDP	65	3401 → 4400 Len=23
UDP	65	3401 → 4400 Len=23
UDP	65	3401 → 4400 Len=23
UDP	65	3400 → 4400 Len=23
UDP	65	3400 → 4400 Len=23
UDP	65	3401 → 4400 Len=23

https://blog.csdn.net/xuandao_ahfengren

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

```
root@kali:~/test# file c297795634cb4f6e8e1d88be044ec0c4
c297795634cb4f6e8e1d88be044ec0c4: gzip compressed data, was "disk-image", last
modified: Mon Sep 15 08:42:23 2014, from Unix, original size modulo 2^32 209715
2
root@kali:~/test#
```

extundelete disk-image --restore-all

生成文件，里面flag加权限运行即可

The image shows a file manager interface with three items: a file icon labeled 'c297795634cb4f6e8e1d88be044ec0c4', a file icon labeled 'disk-image', and a folder icon labeled 'RECOVERED_FILES'. A red arrow points from the 'RECOVERED_FILES' folder to a terminal window. The terminal window shows the following command and output:

```
root@kali:~/test# extundelete disk-image --restore-all
NOTICE: Extended attributes are not restored.
Loading filesystem metadata ... 1 groups loaded.
Loading journal descriptors .. 17 descriptors loaded.
Searching for recoverable inodes in directory / ...
1 recoverable inodes found.
Looking through the directory structure for deleted files ...
0 recoverable inodes still lost.
root@kali:~/test#
```

```
root@kali:~/test/RECOVERED_FILES# chmod +x flag
root@kali:~/test/RECOVERED_FILES# ./flag
your flag is:
de6838252f95d3b9e803b28df33b4baa
root@kali:~/test/RECOVERED_FILES#
```

red_green

进行颜色识别转换为0和1

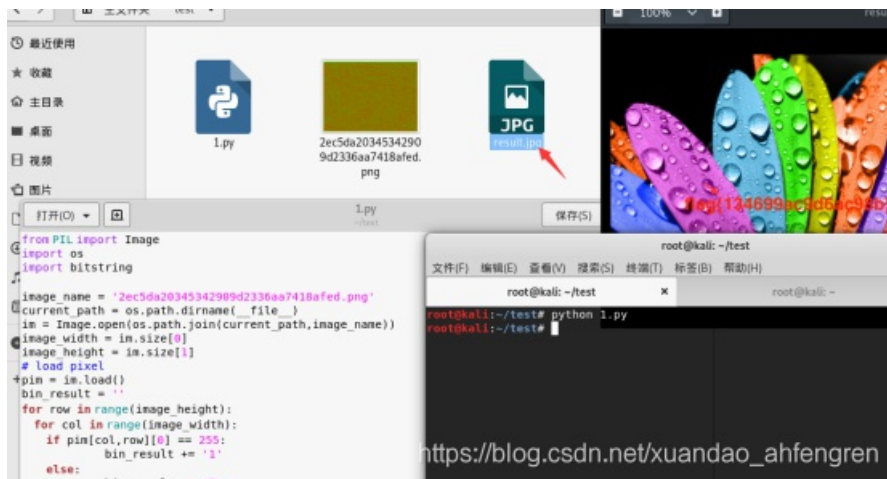
```

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())

```

76 1.py - C:\Users\19154\Desktop\1.py
File Edit Format Run Options Windows Help

```

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())

```

Python 2.7.6 Shell
File Edit Shell Debug Options Windows Help

```

Python 2.7.6 (default, Nov 10 2013, 19:24:18)
32
Type "copyright", "credits" or "license()" for
>>> ===== RESTART =====
>>>
misc.pcapng
flag(c2bbf9cecdaf656cf524d014c5bf046c)
>>>

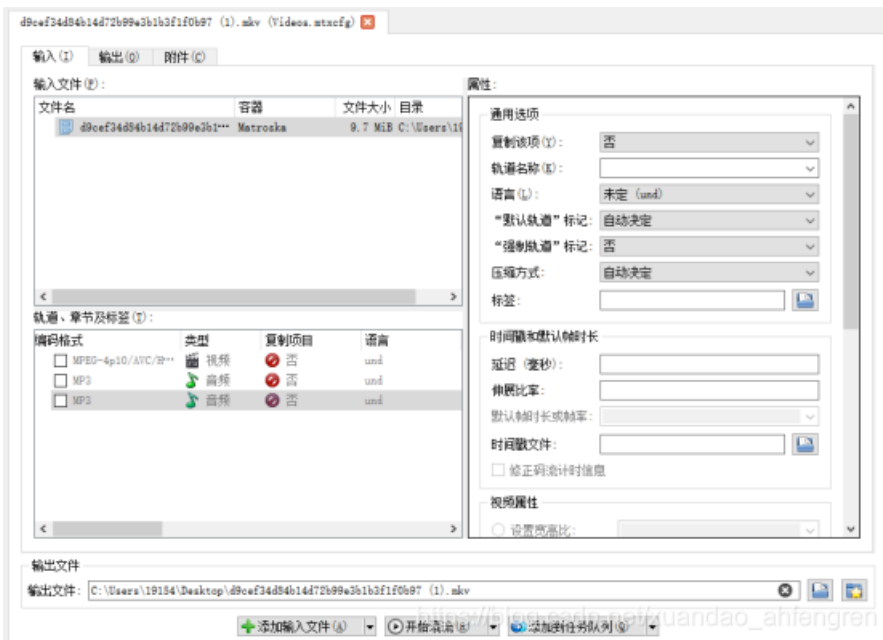
```

https://blog.csdn.net/xuandao_ahfengren

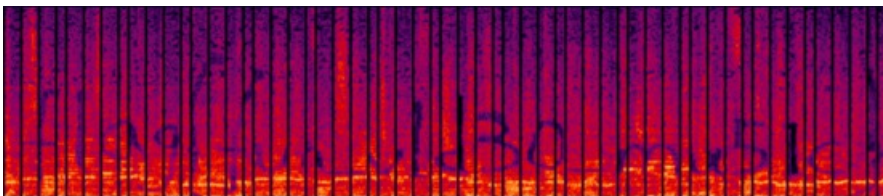
funny_video

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

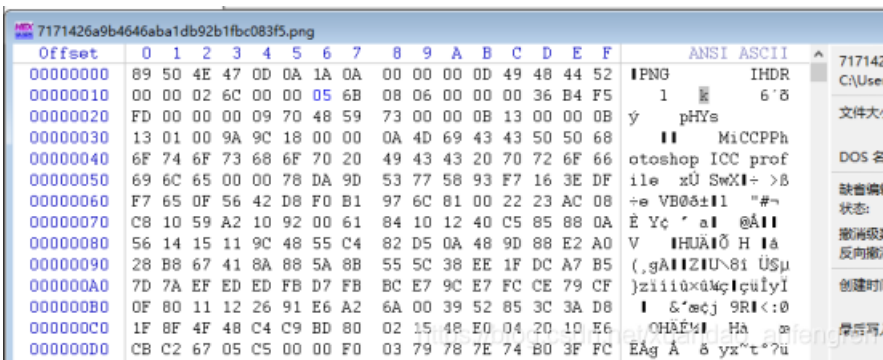


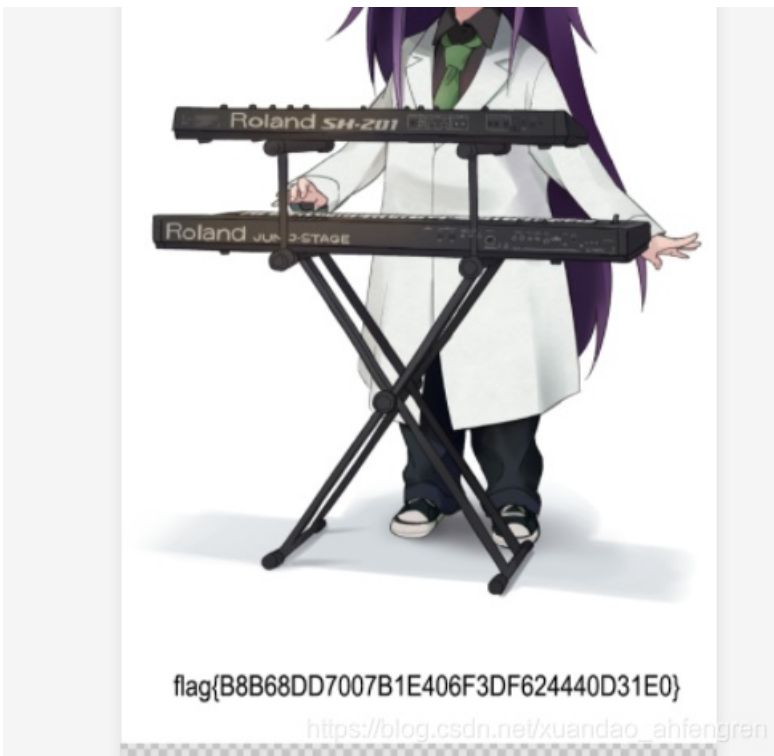
使用Audition打开提取的音频查看频谱，得到flag



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}

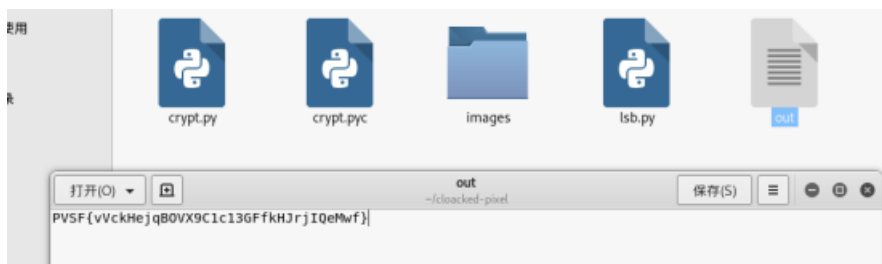
Keyword

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}



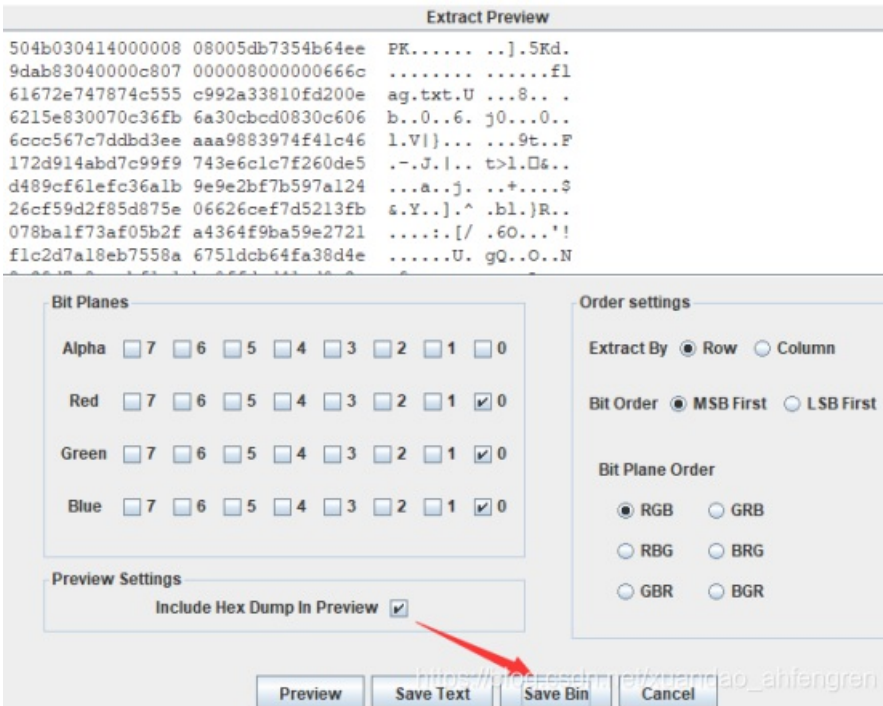
发现是 Nihilist 密码

解密后

flag QCTF{cCgeLdnrIBCX9G1g13KFfeLNsNMRdOwf}

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

