

攻防世界misc高手进阶篇教程（4）

原创

锋刃科技 于 2020-05-27 20:51:01 发布 2736 收藏 3

文章标签: [攻防世界 ctf](#)

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/106390188

版权

misc1

转成十进制后-128(偏移量为128)

再转成ascii码得到flag

```
import re

s = 'd4e8e1f4a0f7e1f3a0e6e1f3f4a1a0d4e8e5a0e6ece1e7a0e9f3baa0c4c4c3d4c6fbb9e1e6b3e3b9e4b3b7b7e2b6b1e4b2b6b9'

num = re.findall('\w{2}', s)


flag = ''

for i in num:

    ch = chr(int(i,16)-128)

    flag += ch

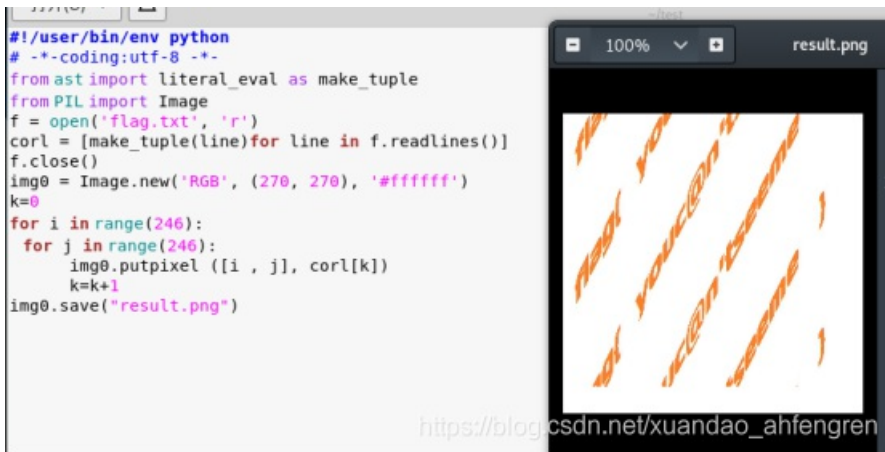
print(flag)
```



```
Python 2.7.6 Shell
Python 2.7.6 (default, Nov 10 2013, 19:24:18) [MSC v.1500 32 bit (Int
32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
>>> That was fast! The flag is: DDCTF{9af3c9d377b61d269b11337f330c935f}
>>>
```

Miscellaneous-200

```
from ast import literal_eval as make_tuple
from PIL import Image
f = open('flag.txt', 'r')
corl = [make_tuple(line) for line in f.readlines()]
f.close()
img0 = Image.new('RGB', (270, 270), '#ffffff')
k=0
for i in range(246):
    for j in range(246):
        img0.putpixel ([i , j], corl[k])
        k=k+1
img0.save("result.png")
```



flag{ youc@n'tseeme }

Miscellaneous-300

运行代码，等一定时间，然后会有12475.zip

```
import zipfile
import re
zipname = "C:\\Users\\19154\\Desktop\\"+"1.zip"
while True:
    if zipname != "C:\\Users\\19154\\Desktop\\73168.zip":
        ts1 = zipfile.ZipFile(zipname)
        #print ts1.namelist()[0]
        res = re.search('[0-9]*',ts1.namelist()[0])
        print res.group()
        passwd = res.group()
        ts1.extractall("C:\\Users\\19154\\Desktop\\",pwd=passwd)
        zipname = "C:\\Users\\19154\\Desktop\\"+ts1.namelist()[0]
    else:
        print "find"
```

我们爆破密码b0yzz

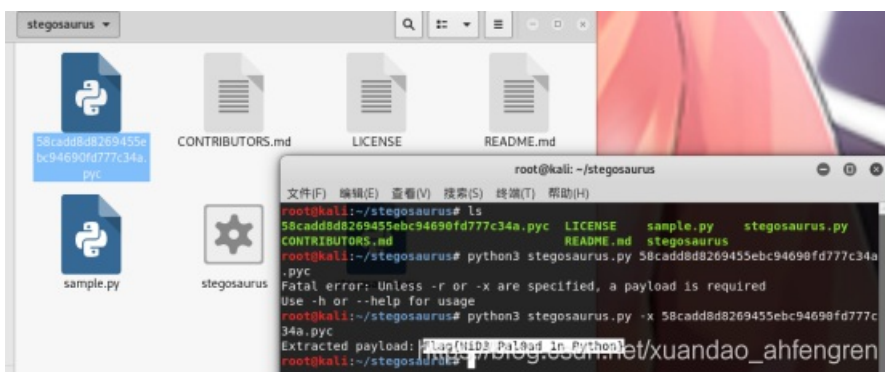
给的是音频文件，猜测是音频隐写，于是将文件拖入Audacity中查看频谱图得到flag

BallsRealBolls

Py-Py-Py

用stegosaurus直接获取flag

python3 stegosaurus.py -x 58cadd8d8269455ebc94690fd777c34a.pyc



传感器1

```
#!/usr/bin/env python
#coding:utf-8
import re
#hex1 = 'AAAAA56A69AA55A95995A569AA95565556' ## 0x8893CA58
hex1 = 'AAAAA56A69AA556A965A5999596AA95656'
def bintohex(s1):
    s2 = ''
    s1 = re.findall('.{4}',s1)
    print ('每一个hex分隔:',s1)
    for i in s1:
        s2 += str(hex(int(i,2))).replace('0x','')

    print ('ID:',s2)
def diffmqst(s):
    s1 = ''
    s = re.findall('.{2}',s)
    cc = '01'
    for i in s:
        if i == cc:
            s1 += '0'
        else:
            s1 += '1'
        cc = i # 差分加上cc = i
    print ('差分曼切斯特解码:',s1)
    bintohex(s1)
def mqst(s): #只能算曼切斯特编码,无法算差分
    mdict = {'5': '00', '6': '01', '9': '10', 'A': '11'}
    a1 = ''.join(mdict[i] for i in s)
    a2 = ''.join(mdict[i][::-1] for i in s)
    print ('曼切斯特解码: ',a1 )
    print ('曼切斯特解码2: ',a2)
    bintohex(a1)
    bintohex(a2)
if __name__ == '__main__':
    bin1 = bin(int(hex1,16))[2:]
    diffmqst(bin1)
mqst(hex1)
```

得到差分曼切斯特编码为8024d8845abf34119，左边去掉5个字符，右边去掉4个字符，换成大写就是flag。

```

File Edit Format Run Options Windows Help
#!/usr/bin/env python
#coding:utf-8
import re
hex1 = 'AAAAA56A69AA55A95995A569AA95565556' ## 0x8893CA58
hex1 = 'AAAAA56A69AA55A965A965A599596AA95656'
def bintoHex(s1):
    s2 = ''
    sl = re.findall('.{4}',s1)
    print ('每一个hex分隔:',sl)
    for i in sl:
        s2 += str(hex(int(i,2))).replace('0x','')
    print ('ID:',s2)
def diffmqt(s):
    sl = ''
    s = re.findall('.{2}',s)
    cc = '01'
    for i in s:
        if i == cc:
            sl += '0'
        else:
            sl += '1'
        cc = i # 差分加上cc = i
    print ('差分曼切斯特解码:',sl)
    bintoHex(sl)
def mqt(s): #只能算曼切斯特编码,无法算差分
    mdict = {'5': '00', '6': '01', '9': '10', 'A': '11'}
    a1 = ''.join(mdict[i] for i in s)
    a2 = ''.join(mdict[i][::-1] for i in s)
    print ('曼切斯特解码: ',a1)
    print ('曼切斯特解码2: ',a2)
    bintoHex(a1)
    bintoHex(a2)
if __name__ == '__main__':
    bin1 = bin(int(hex1,16))[2:]
    diffmqt(bin1)
    mqt(hex1)

```

```

Python 2.7.6 Shell
File Edit Shell Debug Options Windows Hel
Python 2.7.6 (default, Nov 10 2013, 19:
32
Type "copyright", "credits" or "license
>>>
>>>
('x5\x7\xae\x5\x86\x86\x6\x9b\xbc\
a7\xa3\x7\xa0\x81:', '100000000100100
00100011001')
('x6\xaf\x8f\x64\xb8\x80\xe4\xb8\xaaah
000', '0010', '0100', '1101', '1000', '
111', '0011', '0100', '0001', '0001', '
(ID:', '5024d8845abf34119')
('x29b\xbc\x5\x88\x87\x6\x96\xaf\
111111111000111011011110000011110010
('x6\x9b\xbc\x5\x88\x87\x6\x96\xaf\
'11111111100101110011110000101101100
('x6\xaf\x8f\x64\xb8\x80\xe4\xb8\xaaah
111', '1100', '0111', '0110', '1111', '
010', '0010', '0111', '1110', '0001', '
(ID:', 'ffc76f07932a27e11')
('x6\xaf\x8f\x64\xb8\x80\xe4\xb8\xaaah
111', '1100', '1011', '1001', '1111', '
101', '0001', '1011', '1101', '0010', '
(ID:', 'ffc9f0b63151bd22')
>>> |

```

https://blog.csdn.net/xuandao_ahfengren

签到题

Base64解码有得到

ggQ@gQ1fqh0htjpt_sw{gfhgs#}

凯撒密码解密得到,14位


ssc@sc1rct0atfvbf_ei{srtse#}

栅栏密码解密得到flag,7位

ssctf{ssCtf_seC10ver#@rabit}

Excaliflag

放进StegSovle左边点击即可

 StegSolve 1.3 by Caesum



3DS{Gr4b_Only_th1s_B1ts}

Disk

直接提取vmdk文件

提取出来里面有四个flag文件，但是打开并不是flag

第一块出来flag字样，后面拼接放入convert，解出后面字段

flag{4DS_1n_D1sk}

misc_pic_again

发现PK头，保存位zip文件



用winhex发现hctf

```
3 8D 2D A8 08 20 00 TL %" UH -"
3 48 83 EC 08 E8 5D SL)á1ÜHÁy Hi i è]
7 84 00 00 00 00 00 pÿÿHlit |
l FF 14 DC 48 83 C3 L!êL!ôD!ÿAy ÜH!Ã
3 5B 5D 41 5C 41 5D H9ëuëH!Ã [J\^A]
7 84 00 00 00 00 00 A^A_Ãff. |
3 C4 08 C3 00 00 00 óÃ H! H!Ã Ã
3 74 66 7B 73 63 78 hctf{scx
) 61 72 64 34 67 34 dc3tok3yb0ard4g4
) 01 1B 03 3B 30 00 1n~~~} ;0
7 7C 00 00 00 44 FE pÿÿ| Dp
7 A4 00 00 00 54 FF ÿÿL 1ÿÿÿª Tý
7 0C 01 00 00 14 00 ÿÿÃ Aÿÿÿ
```

hctf{scxdc3tok3yb0ard4g41n~~~}

3-1

发现python代码和一串好像加密的字符串，还有flag.zip也导出来

```
Wireshark · 追踪 TCP 流 (tcp.stream eq 6) · ++_++
19aaFYsQQKr+hVX6h12smAUQ5a767TsULEuebWsjEo=[root@localhost wireshark]# ppingg
bbaaiidduu..ccoomm

PING baidu.com (111.13.101.208) 56(84) bytes of data:
64 bytes from 111.13.101.208 (111.13.101.208): icmp_seq=1 ttl=48 time=33.4 ms
64 bytes from 111.13.101.208 (111.13.101.208): icmp_seq=2 ttl=48 time=32.1 ms
64 bytes from 111.13.101.208 (111.13.101.208): icmp_seq=3 ttl=48 time=34.7 ms
64 bytes from 111.13.101.208 (111.13.101.208): icmp_seq=4 ttl=48 time=31.9 ms
...^C
--- baidu.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3037ms
rtt min/avg/max/mdev = 31.921/33.067/34.784/1.155 ms
[root@localhost wireshark]# ccaatt 33

# coding:utf-8
.
.
.
__author__ = 'YFP'
.
.
from Crypto import Random
.
from Crypto.Cipher import AES
.
.
import sys
.
import base64
```

https://blog.csdn.net/xuandao_ahfengren

Wireshark · 导出 · HTTP 对象列表

分组	主机名	内容类型	大小	文件名
274	10.1.10.61:8000	application/octet-stream	169 bytes	flag.rar
566	pcr.da.netease.com	application/x-www-form-urlencoded	528 bytes	receiver
569	pcr.da.netease.com	application/json	12 bytes	receiver

文本过滤器:

https://blog.csdn.net/xuandao_ahfengren

Save Save All Close

```
# coding:utf-8
__author__ = 'YFP'
from Crypto import Random
from Crypto.Cipher import AES
import sys
import base64
IV = 'QWERTYUIOPASDFGH'
def decrypt(encrypted):
    aes = AES.new(IV, AES.MODE_CBC, IV)
    return aes.decrypt(encrypted)

def encrypt(message):
    length = 16
    count = len(message)
    padding = length - (count % length)
    message = message + '\0' * padding
    aes = AES.new(IV, AES.MODE_CBC, IV)
    return aes.encrypt(message)
str = 'this is a test'
example = encrypt(IV)
print(decrypt(example))
s='19aaFYsQQKr+hVX6hl2smAUQ5a767TsULEUebWSajEo='
flag=base64.b64decode(s)
print(decrypt(flag))
```

https://blog.csdn.net/xuandao_ahfengren

加上这些代码即可跑出密码

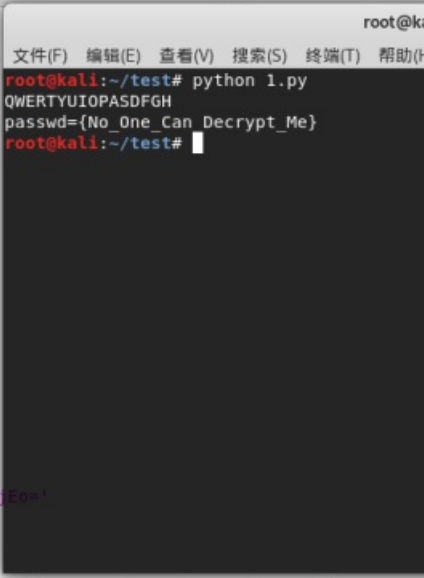
s='19aaFYsQQKr+hVX6hl2smAUQ5a767TsULEUebWSajEo='

flag=base64.b64decode(s)

print(decrypt(flag))

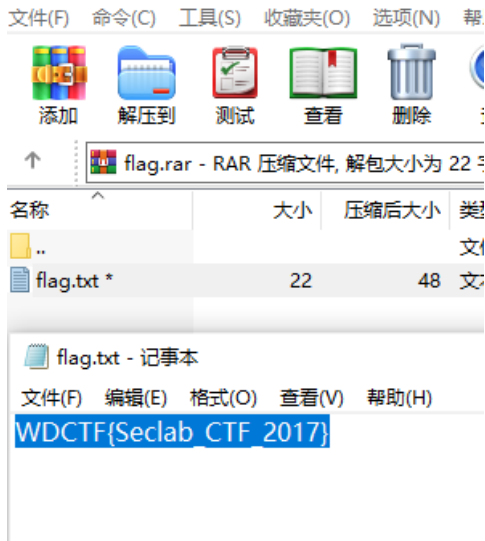
```
# coding:utf-8
__author__ = 'YFP'
from Crypto import Random
from Crypto.Cipher import AES
import sys
import base64
IV = 'QWERTYUIOPASDFGH'
def decrypt(encrypted):
    aes = AES.new(IV, AES.MODE_CBC, IV)
    return aes.decrypt(encrypted)

def encrypt(message):
    length = 16
    count = len(message)
    padding = length - (count % length)
    message = message + '\0' * padding
    aes = AES.new(IV, AES.MODE_CBC, IV)
    return aes.encrypt(message)
str = 'this is a test'
example = encrypt(IV)
print(decrypt(example))
s='19aaFYsQQKr+hVX6hl2smAUQ5a767TsULEUebWSajEo='
flag=base64.b64decode(s)
print(decrypt(flag))
```



```
root@kali:~/test# python 1.py
QWERTYUIOPASDFGH
passwd={No_One_Can_Decrypt_Me}
root@kali:~/test#
```

https://blog.csdn.net/xuandao_ahfengren



reverseMe

反转即可

```
flag{4f7548f93c7bef1dc6a0542cf04e796e}
```

```
flag{4f7548f93c7bef1dc6a0542cf04e796e}
```

test.py

首先进行反编译python文件

发现这是倒序的base64

得到fjU1MmYyNWcyNmcyOTgyYjY4MTc5NWMzZjc0ZzllNzMyfGhibWc=

```
str = 'jYygTOy' + 'cmNycWNyYmMlUj'
import base64

def flag1():
    code = str[::-3]
    result = ''
    for i in code:
        ss = ord(i) - 1
        result += chr(ss)

    print result[::-1]

def flag2():
    code = str[::-2]
    result = ''
    for i in code:
        ss = ord(i) - 1
        result += chr(ss)

    print result[::-2]

def flag3():
    pass
# WARNING: Decompile incomplete

flag1()
```


解码得到552f25g26g2982b681795c3f74g9e732|hbmj, 然后颠倒顺序得到
gmbh|237e9g47f3c597186b2892g62g52f255

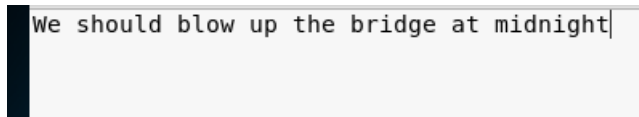
由于hbmj 与 flag的联系是ascii差1, 于是全部rot(-1), 得flag

flag{126d8f36e2b486075a1781f51f41e144}

Avatar

我们直接用outguess分解出来, 1.txt文件就是flag

outguess -r 035bfaa85410429495786d8ea6ecd296.jpg 1.txt



Wireshark

导出这个图片

```
00004d8 01011001 01000001 00110100 01100010 00001101 00001010 01000011 01101111 YA4b-...
00004e0 01101110 01101000 01100101 01101110 01101000 00101101 01000100 01101001 Content-Di
00004e8 01101001 01100000 01101111 01100011 01101001 01101000 01101001 01101111 50001110
00004f0 01101110 00110100 00100000 01100110 01101111 01100010 01101110 00101100 44444444
00004f8 01100100 01100001 01101000 01100001 00110111 00100000 01101110 01100001
```

修改高度

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI
00000000	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	%PNG
00000010	00	00	06	40	00	00	03	20	08	06	00	00	00	7B	C0	AE	@
00000020	5A	00	00	0C	14	69	43	43	50	49	43	43	20	50	72	6F	Z iCCP1
00000030	66	69	6C	65	00	00	48	89	95	57	07	58	53	C9	16	9E	file Hw*W
00000040	5B	52	08	09	2D	10	01	29	A1	37	41	8A	74	E9	BD	08	[R -);i
00000050	48	07	1B	21	09	49	28	11	12	82	8A	1D	59	54	70	2D	H ! I(,
00000060	A8	58	B0	A2	AB	20	0A	AE	05	90	B5	62	57	16	C1	DE	"X"« @
00000070	1F	88	A8	AC	AC	8B	05	2C	A8	BC	49	01	5D	5F	FB	DE	" "« , "»
00000080	F9	BE	B9	F3	E7	CC	39	67	FE	33	F7	DC	C9	0C	00	AA	ù4'óç¡9gp3
00000090	F6	AC	DC	DC	6C	54	0D	80	1C	61	BE	28	36	C4	9F	99	ö-UÜ!T e a



key:57pmYyWt

https://blog.csdn.net/xuandao_ahfengren

把里面的图片都在网站上去解密，然后就得到了一串16进制字符串格式的flag

1. 从电脑中选择一张带有隐藏信息的图片:

2. 输入需要解开信息的密码 (如果没有密码可以不填):

解密出隐藏的信息

图片中隐藏的信息为: flag + AHs-
44444354467B5145576F6B63704865556F32574F6642494E37706F6749577346303469526A747D+AH0-

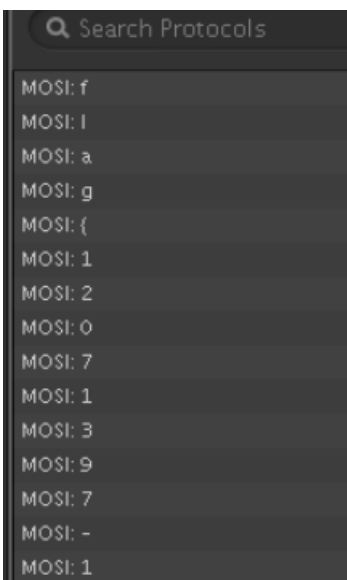
拿去解密就得到最后的flag了

```
DDCTF{QEWokcpHeUo2WOfBIN7pogIWsF04iRjt}
```

Saleae

使用Logic软件打开

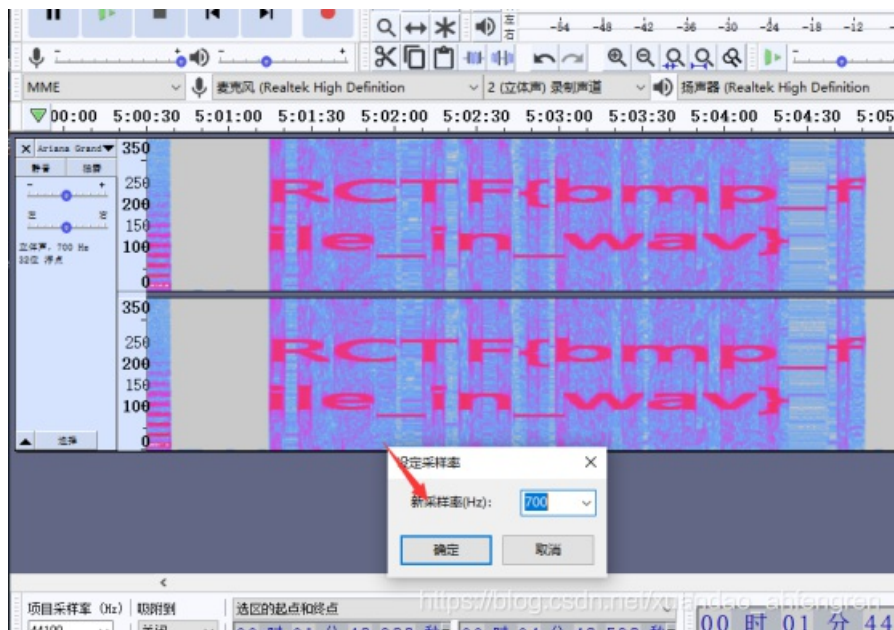
flag竖着读



flag{12071397-19d1-48e6-be8c-784b89a95e07}

intoU

修改采样率即可



Message

```
print bin(int(open("msg.txt","r").read(),16))[2:].replace("0",".").replace("1","#")
```

这个脚本能够生成.和#的序列。我们把这个序列放到notepad++里，一直把字体减小到最小，然后重新调整窗口的大小，直到我们能看出一些东西。最后图案显现出来：The flag is RCTF{ArEciBo_mEsSaGe}