

XCTF MISC 几道进阶题

原创

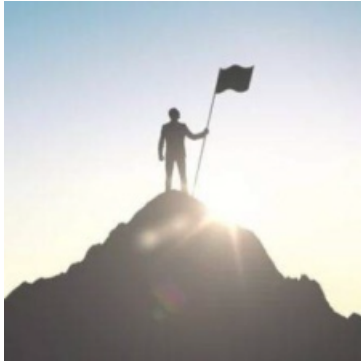
[A_dmins](#) 于 2019-12-11 13:50:13 发布 814 收藏 2

分类专栏: [CTF题](#) [XCTF](#)

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/qq_42967398/article/details/102784401

版权



[CTF题](#) 同时被 2 个专栏收录

115 篇文章 11 订阅

订阅专栏



[XCTF](#)

24 篇文章 0 订阅

订阅专栏

XCTF MISC 几道进阶题

好久没有打CTF了, 做几道misc玩玩吧

小小的PDF

直接binwalk看一下，发现存在三张图片，，，

```
root@kali:~# binwalk 7e5ab2e7587d4a4abf9c705dfb935a92.pdf
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	PDF document, version: "1.4"
452	0x1C4	JPEG image data, JFIF standard 1.01
73254	0x11E26	JPEG image data, JFIF standard 1.01
81606	0x13EC6	Zlib compressed data, default compression
82150	0x140E6	JPEG image data, JFIF standard 1.01
104469	0x19815	Zlib compressed data, default compression
105134	0x19AAE	Zlib compressed data, default compression

binwalk -e一下没发现有什么东西，，，，

直接使用dd命令吧：`dd if=7e5ab2e7587d4a4abf9c705dfb935a92.pdf of=1 skip=82150 bs=1`

```
root@kali:~# dd if=7e5ab2e7587d4a4abf9c705dfb935a92.pdf of=1 skip=82150 bs=1
26695+0 records in
26695+0 records out
26695 bytes (27 kB, 26 KiB) copied, 0.157018 s, 170 kB/s
root@kali:~#
```

得到flag:

SYC{so_so_so_easy}



https://blog.csdn.net/qq_42967398

不明白以前为什么不会写，，，

Cephalopod

下载下来是一个pcap文件，用strings大法先看看：

```
root@kali:~# strings 1.pcap | grep "flag"
flag.png
flag.png
flag.png
root@kali:~#
```

有图片??? binwalk一下：

```
root@kali:~# binwalk 1.pcap
DECIMAL      HEXADECIMAL  DESCRIPTION
-----
0            0x0          Libpcap capture file, little-endian, version 2.4,
Ethernet, snaplen: 262144
26441       0x6749      PNG image, 1754 x 2480, 8-bit/color RGBA, non-interlaced
26577       0x67D1      Zlib compressed data, best compression
```

binwalk -e无用，foremost也无用，难受

试试dd命令：`dd if=1.pcap of=1 skip=82150 bs=1`

emmmm，图片出来了但是看不了，无用，，，，，

在windows下wireshark也无用，没看见能提取的文件，，，

最后查阅资料得知tcpextract可以从网络流量中提取文件，然而kali上没有，需要安装，，，

安装直接百度其他人的吧，装好直接利用命令：`tcpextract -f 1.pcap`

能够得到flag图片，，，，，



```
HITB{95700d8aefdc1648b90a92f3a8460a2c}
```

misc 2 - 1

下载文件发现图片打不开，拖入winhex，发现文件头不对，修改文件头：

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ANSI	ASCII
00000000	80	59	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	€YNG	IHDR
00000016	00	00	00	00	00	00	02	F8	08	06	00	00	00	93	2F	8A	ø	"/Š
00000032	6B	00	00	00	04	67	41	4D	41	00	00	9C	40	20	0D	E4	k	gAMA α@ ä
00000048	CB	00	00	00	20	63	48	52	4D	00	00	87	0F	00	00	8C	Ë	cHRM † Œ
00000064	0F	00	00	FD	52	00	00	81	40	00	00	7D	79	00	00	E9	ýR	@ }y é
00000080	8B	00	00	3C	E5	00	00	19	CC	73	3C	85	77	00	00	0A	<	<ã Ìs<...w
00000096	39	69	43	43	50	50	68	6F	74	6F	73	68	6F	70	20	49	9iCCP	Photoshop I
00000112	43	43	20	70	72	6F	66	69	6C	65	00	00	48	C7	9D	96	CC	profile HÇ -
00000128	77	54	54	D7	16	87	CF	BD	77	7A	A1	CD	30	D2	19	7A	wTT*	#İ*szjİ0 z

修改之后还是打不开，突然发现表示宽度的地方显示为0???

怪不得打不开，，，随便修改一下宽度，然后发现：



没什么用，估计要crc爆破宽度了，，，

```
import struct
import binascii
import os

m = open("1.png", "rb").read()

for i in range(0, 65535):
    c = m[12:16] + struct.pack('>i', i) + m[20:29]
    crc = binascii.crc32(c) & 0xffffffff
    if crc == 0x932f8a6b:
        print(hex(i))
```

```
python 1.py  
0x2c5
```

得到flag图片：



János-the-Ripper

下载一个压缩包解压之后一个未命名的文件，查看一下发现是PK：

```
PKETXEOTDC4NULETXNULBSNULSO DD督w' NU
```

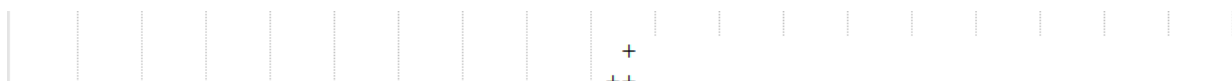
改名为.zip，解压需要密码，，，发现不是伪加密
直接使用工具爆破，得到密码：fish



彻底怀疑自己以前是不是没带脑子，，，，

can_has_stdio?

下载解压得到一个文件，发现：



MISCall

下载文件放到kali中使用file查看一下文件：

```
root@kali:~# file 123
123: bzip2 compressed data, block size = 900k
```

是一个bzip2的压缩文件，直接使用命令：`tar -xvjf 123` 的到文件：

```
root@kali:~# tar -xvjf 123
ctf/
ctf/flag.txt
ctf/.git/
ctf/.git/description
ctf/.git/refs/
ctf/.git/refs/heads/
ctf/.git/refs/heads/master
ctf/.git/refs/stash
ctf/.git/refs/tags/
ctf/.git/ORIG_HEAD
ctf/.git/logs/
ctf/.git/logs/refs/
ctf/.git/logs/refs/heads/
ctf/.git/logs/refs/heads/master
ctf/.git/logs/refs/stash
ctf/.git/logs/HEAD
ctf/.git/HEAD
ctf/.git/COMMIT_EDITMSG
ctf/.git/hooks/
ctf/.git/hooks/pre-commit.sample
ctf/.git/hooks/update.sample
ctf/.git/hooks/applypatch-msg.sample
ctf/.git/hooks/pre-applypatch.sample
```

https://blog.csdn.net/qq_42967398

好像是.git目录??? 进入目录看一下：



好像“.”开头的文件都被隐藏了??? flag.txt中没有flag，估摸着.git中隐藏了什么东西。。。。

查看了一下.git列表好像没有什么特别的地方，，，，

最后得知一个git stash命令，

git stash会把所有未提交的修改（包括暂存的和非暂存的）都保存起来，用于后续恢复当前工作目录

查看现有stash: `git stash list`

查看列表: `git stash show`

复原文件: `git stash apply`

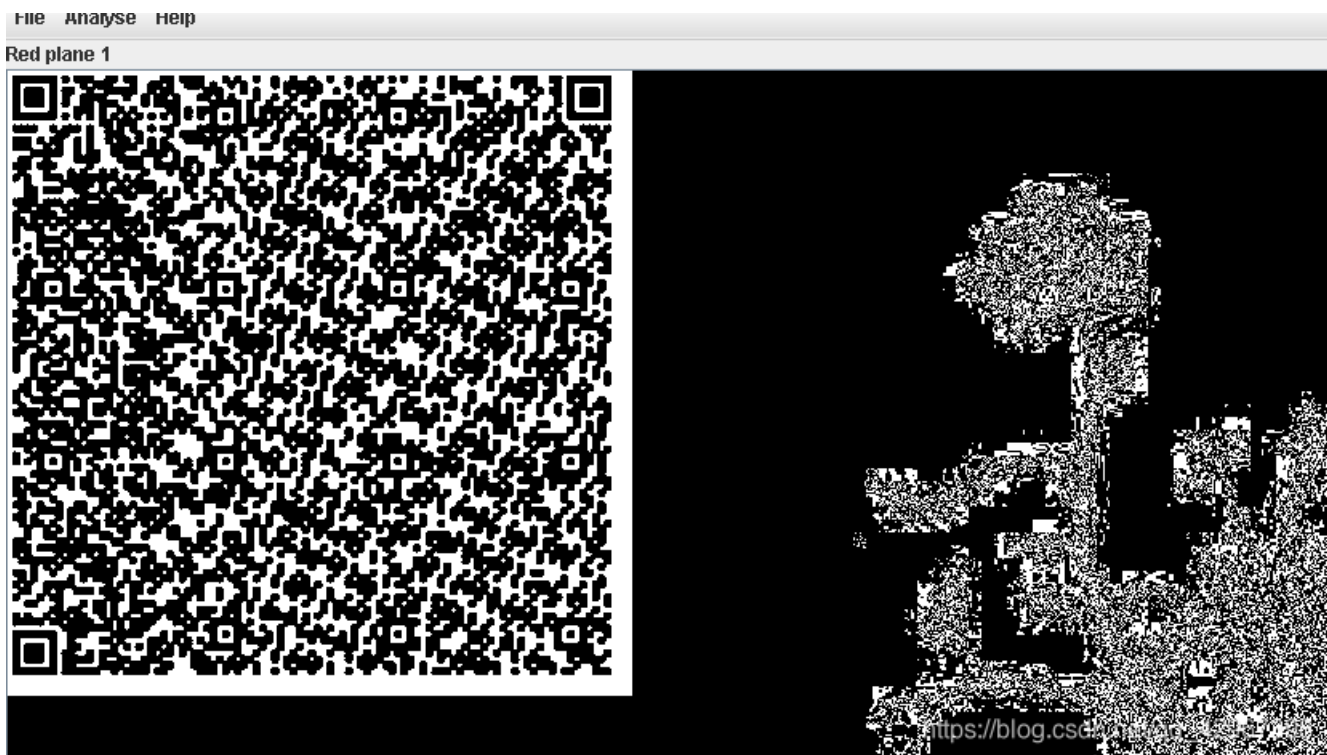
```
root@kali:~/ctf# git stash list
stash@{0}: WIP on master: bea99b9 Initial commit 38338
root@kali:~/ctf# git stash show
flag.txt | 25 ++++++
s.py | 4 +
2 files changed, 28 insertions(+), 1 deletion(-)
root@kali:~/ctf# git stash apply 539c30d56b278e7529dc4
On branch master
Changes to be committed:
  new file:   flag.txt
```

```
g:(use "git reset HEAD <file>..." to unstage)
git/objects/40/7bb5da52f79fcel1f5bdd998f16b9aeffa974ef
git/obj:new file:/ s.py
git/objects/26/
Changes not staged for commit:a780555299b3e8aef4eed7c4
g:(use "git add <file>..." to update what will be committed)
g:(use "git checkout <file>..." to discard changes in working directory)
git/branches/
git/con modified: flag.txt
git/info/
```

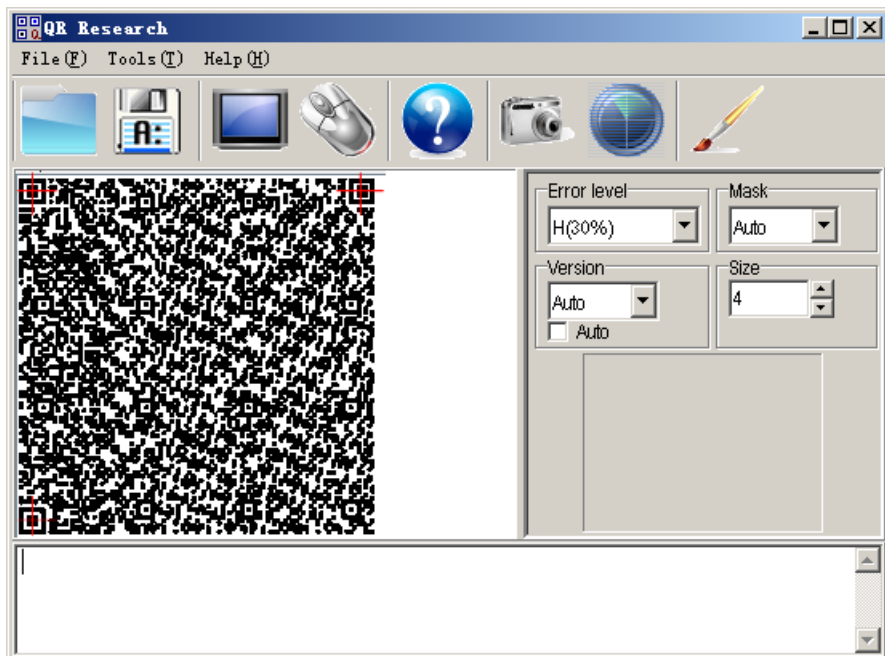
出现一个s.py文件，运行得到flag，，，，

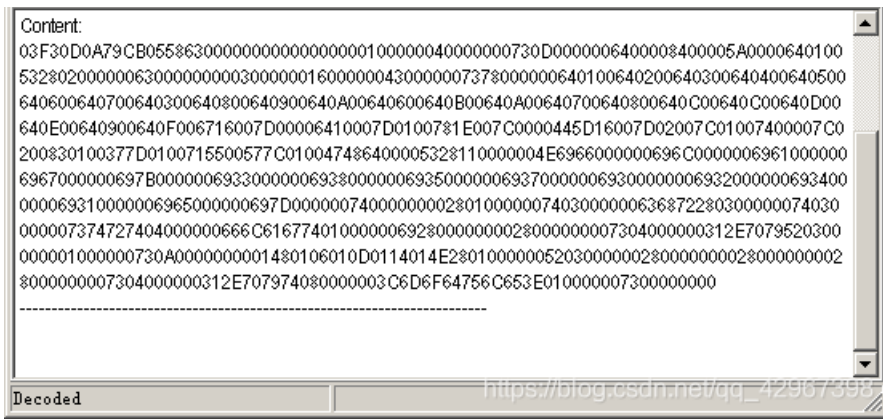
适合作为桌面

下载文件解压发现是一张图片！！放入stego中查看一下，发现存在二维码：



利用二维码扫描工具扫到内容：

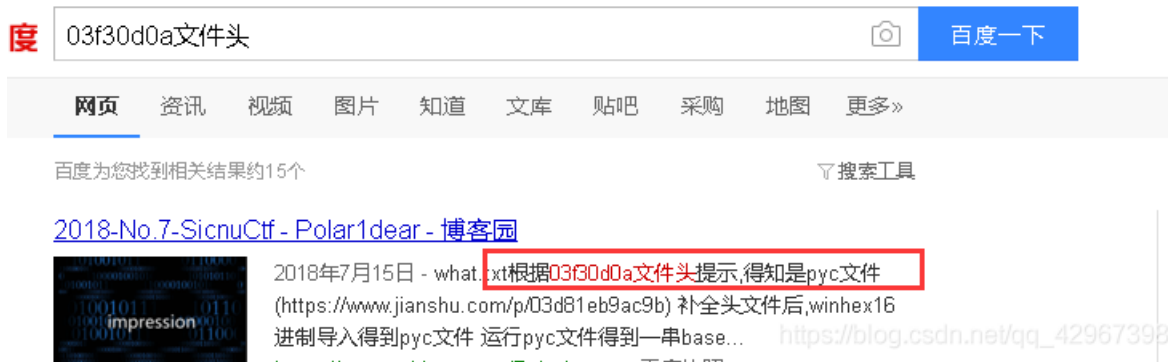




扣下来:

```
03F30D0A79CB055863000000000000001000004000000730D000006400008400005A000064010053280200000630000000030000
001600000043000000737800000640100640200640300640400640500640600640700640800640900640A00640B00640C00640D00
640E00640F006410006411000781E007C0000445D16007D02007C01007400007C0
200830100377D0100715500577C01004748640000532811000004E696600000696C0000069610000069670000069693000006963300000696340000069631000006965000006967D00000740000000280100000740300000063687228030000074030
00000737472740400000666C6167740100000692800000002800000007304000000312E7079520300
00001000000730A000000001480106010D0114014E280100000520300000028000000028000000073040000031
800000007304000000312E707974080000003C6D6F64756C653E010000007300000000
```

怀疑是一个文件之类的，查找一下是否为文件头:



原来是pyc的文件头，直接用winhex保存为pyc文件

利用在线反编译工具的到源码:

```
def flag():
    str = [
        102,
        108,
        97,
        103,
        123,
        51,
        56,
        97,
        53,
        55,
        48,
        51,
        50,
        48,
        56,
        53,
        52,
        52,
        49,
        101,
        55,
        125]
    flag = ''
    for i in str:
        flag += chr(i)

    print flag
```

直接运行一下得到flag!

```
<base> C:\Users\Administrator\Desktop>python 1.py
flag{38a57032085441e7}
```

misc 3 - 1

下载文件下来，发现是一个rar压缩包，解压得到一个文件，发现是一个pcap文件
wireshark打开文件来，没找到什么，直接搜索字符串flag找到一个flag.rar文件??

269	39.7103520	10.1.70.61	10.1.10.61	TCP	57746	>	iradm	[ACK]	Seq=1 Ack=1 win=29312 Len=0 TSV=522971881 TSER=145367124
271	39.7111530	10.1.70.61	10.1.10.61	TCP	57746	>	iradm	[PSH, ACK]	Seq=1 Ack=1 win=29312 Len=121 TSV=522971882 TSER=145367124
273	39.7118470	10.1.10.61	10.1.70.61	TCP	iradm	>	57746	[PSH, ACK]	Seq=1 Ack=122 win=532736 Len=20 TSV=145367126 TSER=522971882
274	39.7119800	10.1.10.61	10.1.70.61	TCP	iradm	>	57746	[FIN, PSH, ACK]	Seq=201 Ack=122 win=532736 Len=169 TSV=145367126 TSER=522971882
275	39.7125770	10.1.70.61	10.1.10.61	TCP	57746	>	iradm	[ACK]	Seq=122 Ack=201 win=30336 Len=0 TSV=522971884 TSER=145367126
276	39.7152570	10.1.70.61	10.1.10.61	TCP	57746	>	iradm	[FIN, ACK]	Seq=122 Ack=371 win=31360 Len=0 TSV=522971886 TSER=145367126
277	39.7152770	10.1.10.61	10.1.70.61	TCP	iradm	>	57746	[ACK]	Seq=371 Ack=123 win=532736 Len=0 TSV=145367130 TSER=522971886

Frame 271: 187 bytes on wire (1496 bits), 187 bytes captured (1496 bits)

Ethernet II, Src: 80:f6:2e:fb:f1:a3 (80:f6:2e:fb:f1:a3), Dst: ec:a8:6b:3a:c1:e5 (ec:a8:6b:3a:c1:e5)

Internet Protocol, Src: 10.1.70.61 (10.1.70.61), Dst: 10.1.10.61 (10.1.10.61)

Transmission Control Protocol, Src Port: 57746 (57746), Dst Port: iradm (8000), Seq: 1, Ack: 1, Len: 121

Data (121 bytes)

Data: 474554202F666c61672e72617220485454502F312e310d0a...
[Length: 121]

```
0000 ec a8 6b 3a c1 e5 80 f6 2e fb f1 a3 08 00 45 00 ..k:....E.
0010 00 ad 27 af 40 00 3f 06 af 20 0a 01 46 3d 0a 01 ..'.@.?..F..
0020 0a 3d e1 92 1f 40 8d d9 cd 4d f7 8e cc 03 80 18 .._..@...M.....
0030 00 e5 bc 66 00 00 01 01 08 0a 1f 2b ea ea 08 aa ..f.....
0040 20 54 47 43 94 20 2f 06 8c 61 67 2e 72 31 72 20 ..f.....
0050 48 54 54 50 2f 31 2e 31 0d 0a 55 73 65 72 2d 41 ..HTTP/1.1..User-A
0060 67 65 6e 74 3a 20 57 67 65 74 2f 31 2e 31 34 20 gent: wget/1.14
0070 28 6c 69 6e 75 78 2d 67 6e 75 29 0d 0a 41 63 63 (linux-gnu)..Acc
0080 65 70 74 3a 20 2f 2a 0d 0a 48 6f 73 74 3a 20 ept: /* ..Host:
0090 31 30 2e 31 2e 31 30 2e 36 31 3a 38 30 30 30 0d 10.1.10.61:8000.
00a0 0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 4b 65 65 .connect ion: Kee
00b0 20 2d 41 6c 69 76 65 0d 0a 0d 0a p-Alive. ...
```

https://blog.csdn.net/qq_42967398

save下来，发现需要密码??? 难受了，返回去继续看wireshark, emmmm
在tcp.stream eq 6中发现可疑内容:

```
[root@localhost wireshark]# llss
1 2 3 test
[root@localhost wireshark]# ccaatt 11
Rar!...3...
.....TU..<..... +.....f flag.txt0.....n.Kr..z...uEo.Bn&=i.S.>...4.B.~...xj.".
...u.....3.....jWj..%m..!+.h...+s..q#..]...3Ks.y.....r.2...wVQ...[root@localhost wireshark]# ccaatt 22
19aaFYsQQKr+hVX6h12smAUQ5a767TsULEUebWSajEo=[root@localhost wireshark]# ppiinnngg 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
.
.
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(str)
.
.
print(decrypt(example))
.

```

一串字符串：19aaFYsQQKr+hvX6hl2smAUQ5a767TsULEUebWSajEo=

还有一个python脚本，，，，emmmm，估摸着要我们解密！！

直接修改一下脚本：

```
# 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)

example = base64.b64decode("19aaFYsQQKr+hVX6h12smAUQ5a767TsULEUebWSajEo=")
print(decrypt(example))
```

运行得到:

```
<base> C:\Users\Administrator\Desktop>python 1.py
b'passwd=<No_One_Can_Decrypt_Me>\x00\x00'
```

得到解压密码: `No_One_Can_Decrypt_Me`

解压得到flag, , ,