

RAR文件码流分析

标记块: HEAD_TYPE=0x72

压缩文件头: HEAD_TYPE=0x73

文件头: HEAD_TYPE=0x74

旧风格的注释头: HEAD_TYPE=0x75

旧风格的用户身份信息: HEAD_TYPE=0x76

旧风格的子块: HEAD_TYPE=0x77

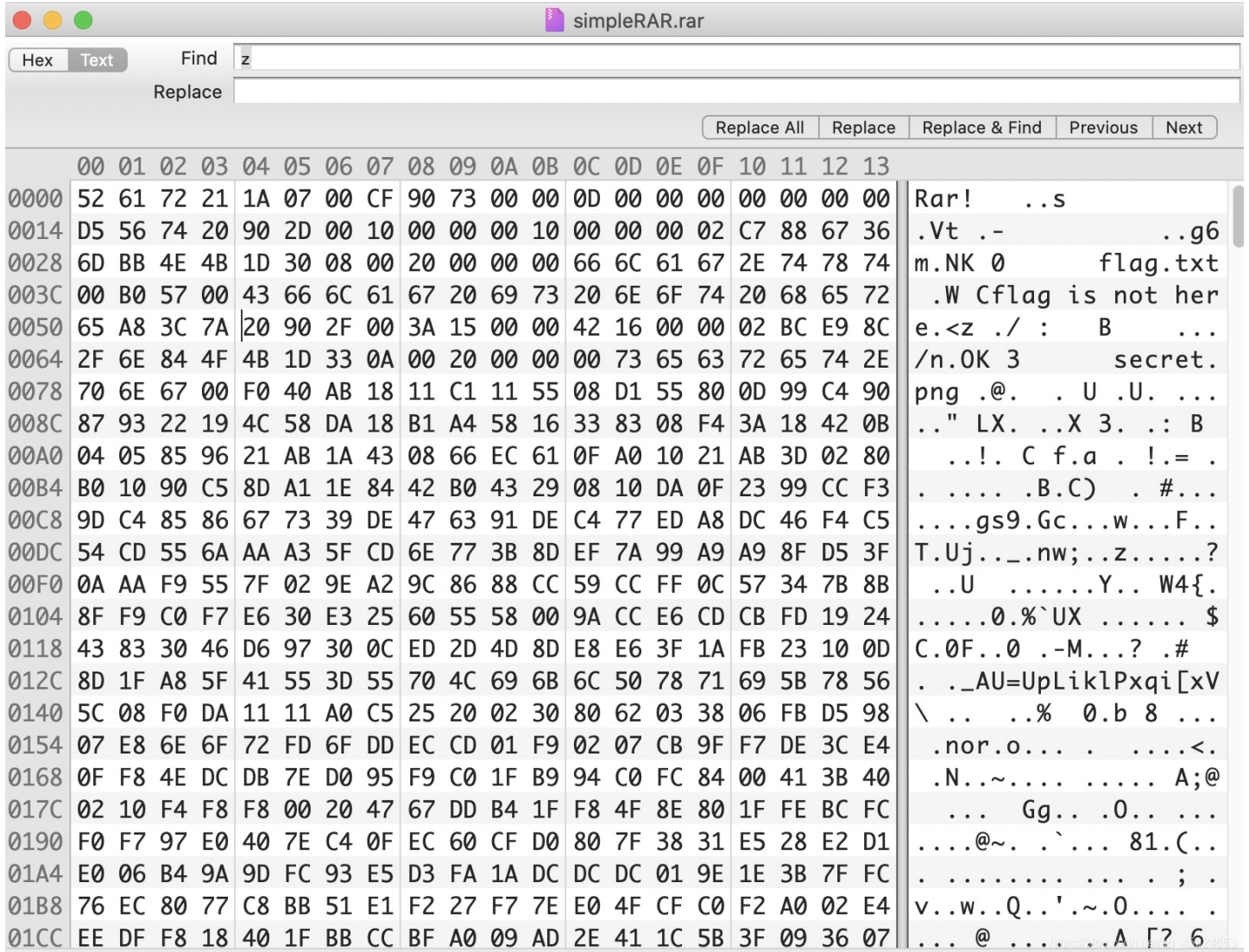
旧风格的恢复记录: HEAD_TYPE=0x78

旧风格的用户身份信息: HEAD_TYPE=0x79

子块: HEAD_TYPE=0x7A

最后的结束块: HEAD_TYPE=0x7B

将7A改为74文件头



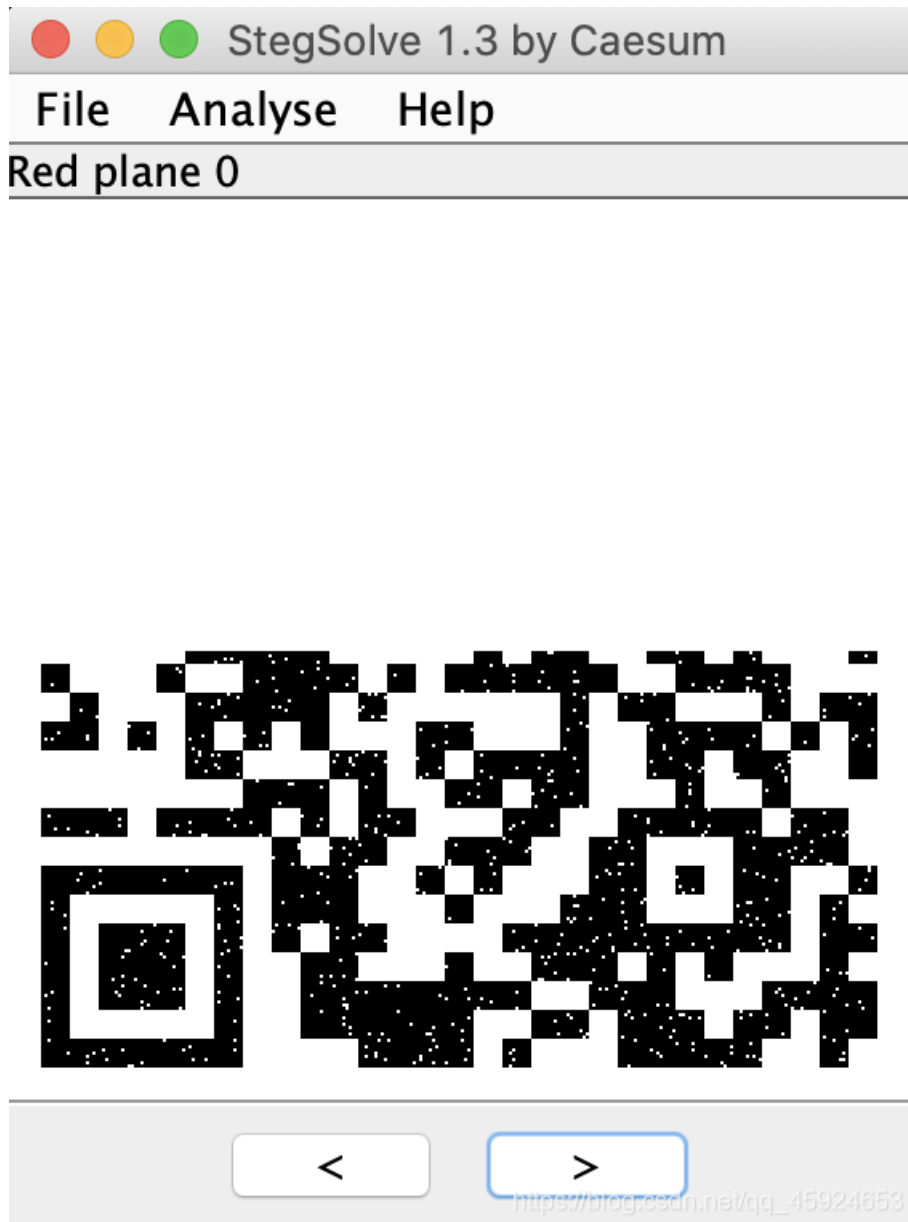
The screenshot shows a hex editor window titled 'simpleRAR.rar'. The search bar contains 'z'. The hex data is displayed in columns 00 to 13. The text view on the right shows the following content:

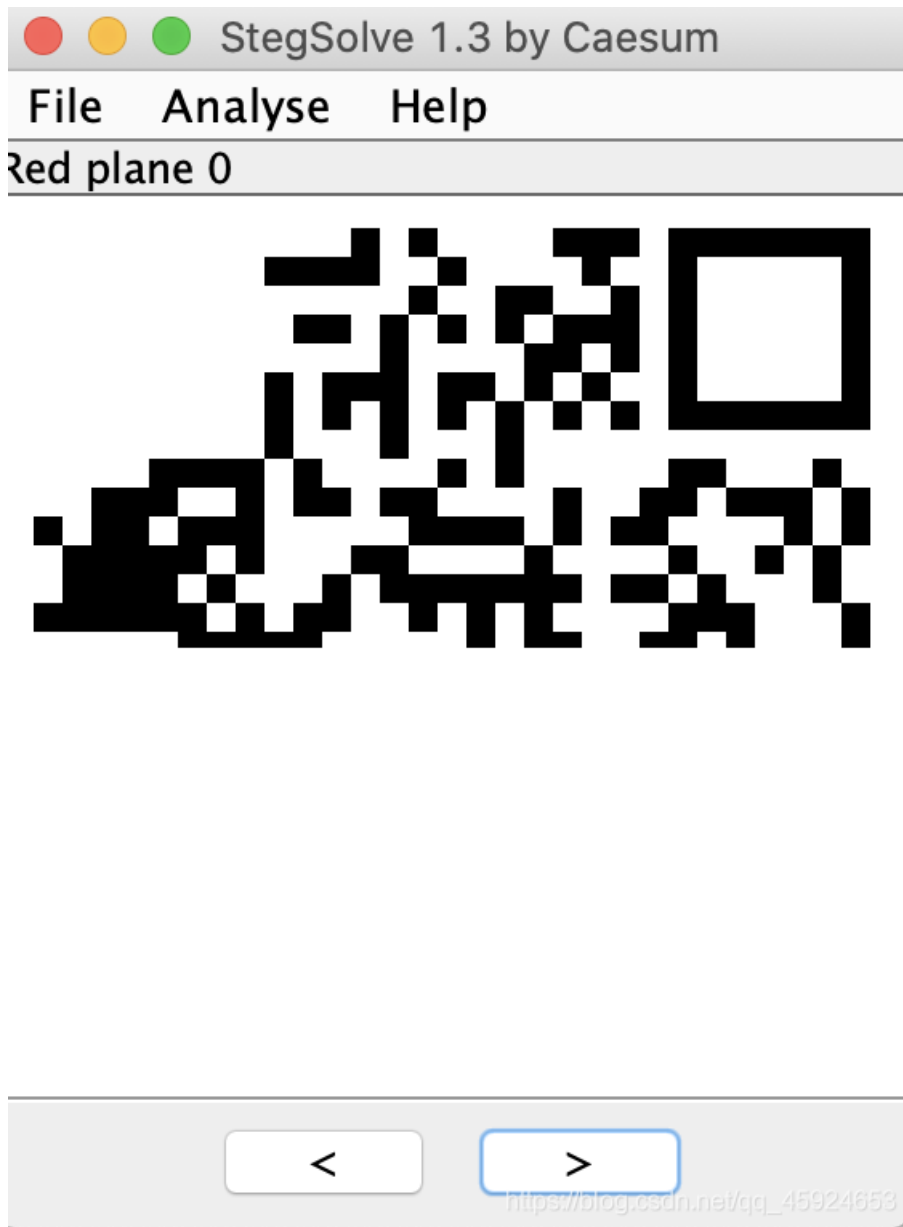
```
Rar! ..s
.Vt .- ..g6
m.NK 0 flag.txt
.W Cflag is not her
e.<z ./ : B ...
/n.OK 3 secret.
png .@. . U .U. ...
.." LX. ..X 3. .: B
..!. C f.a . != .
. .... .B.C) . #...
....gs9.Gc...w...F..
T.Uj...nw;..z....?
..U .....Y.. W4{.
.....0.%`UX ..... $
C.0F..0 .-M...? .#
. . _AU=UpLklPxqi[xV
\ .. ..% 0.b 8 ...
.nor.o... . ....<.
.N..~.... ..... A;@
... Gg.. .0.. ...
....@~. .`... 81.(..
. .... . . ; .
v..w..Q..'~.0.... .
... @ .... .A [? 6
```

解压后有一张png图片, 提示中说双图层用ps打开后报错



将两个图层保存打开stegsolve看一下，得到二维码，补全定位角得到flag





base64stego

base64隐写

隐写原理：base64编码再解码这个过程中，会出现"="个数*2bit的字节数的无效字节，也就是说这些字节无论填充什么，都不会影响结果

```
import base64
b = ''
for i in range(26):
    b = 'U' + chr(65 + i) + '=='
    print(b)
    print(base64.b64decode(b))
>>>
UA==
b'P'
UB==
b'P'
UC==
b'P'
UD==
b'P'
UE==
b'P'
UF==
b'P'
UG==
b'P'
UH==
b'P'
UI==
b'P'
UJ==
b'P'
UK==
b'P'
UL==
b'P'
UM==
b'P'
UN==
b'P'
UO==
b'P'
UP==
b'P'
UQ==
b'Q'
UR==
b'Q'
US==
b'Q'
UT==
b'Q'
UU==
b'Q'
UV==
b'Q'
UW==
b'Q'
UX==
b'Q'
UY==
b'Q'
UZ==
b'Q'
```

要是发P UA==可以UB==也可以
拿隐写脚本跑一下

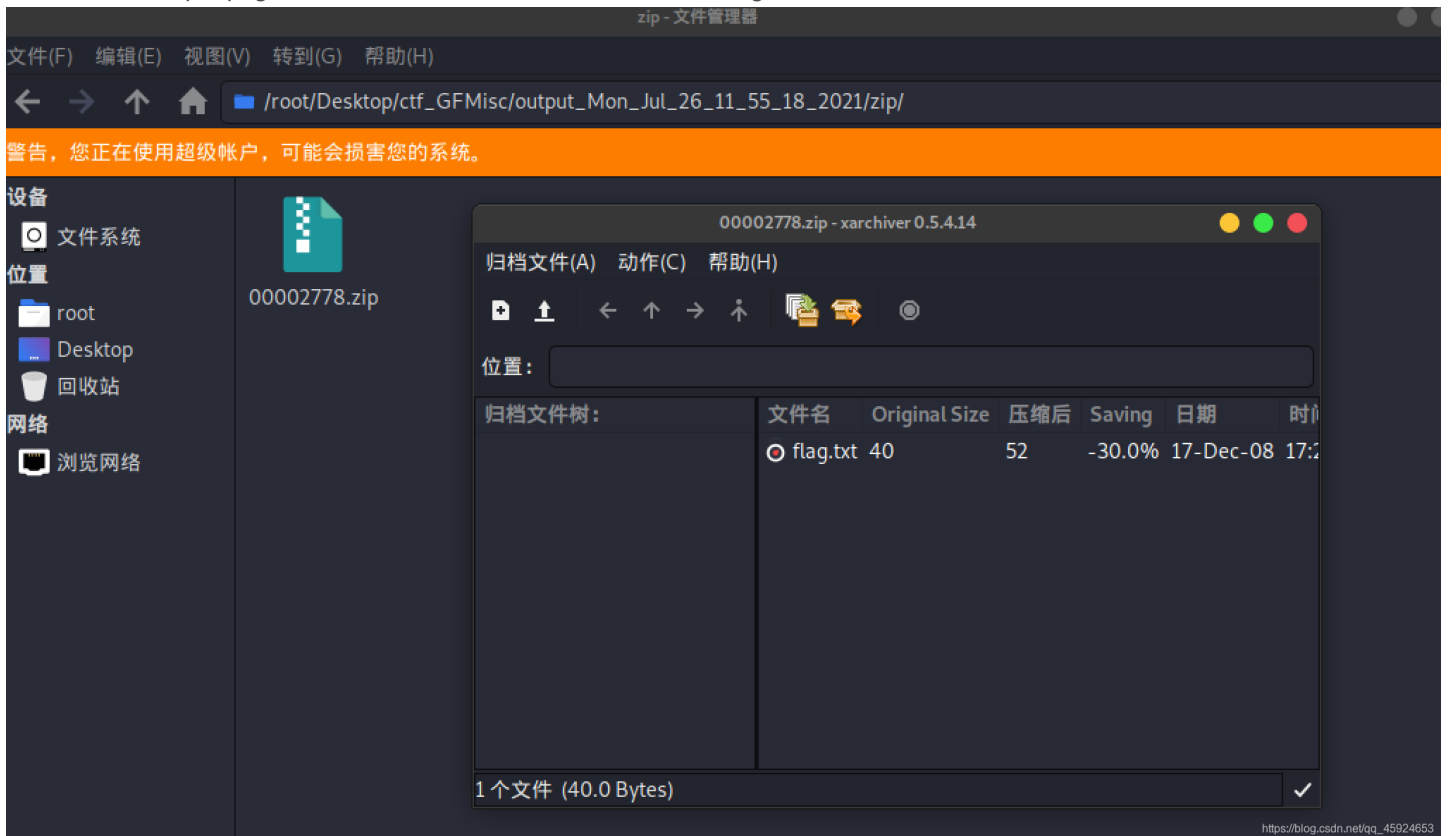
```
import base64
b64chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
with open('stego.txt', 'rb') as f:
    bin_str = ''
    for line in f.readlines():
        stegb64 = str(line, "utf-8").strip("\n")
        rowb64 = str(base64.b64encode(base64.b64decode(stegb64)), "utf-8").strip("\n")#删除首尾换行

    offset = abs(b64chars.index(stegb64.replace('=', ''))[-1]) - b64chars.index(rowb64.replace('=', ''))[-1])
    #把末尾的'='去掉,offset = |给定字符串在b64表中的位置-解码字符串在b64表中的位置|
    equalnum = stegb64.count('=') #计算等号个数
    if equalnum:
        bin_str += bin(offset)[2:].zfill(equalnum * 2)#返回offset的二进制形式
        print(''.join([chr(int(bin_str[i:i + 8], 2)) for i in range(0, len(bin_str), 8)])) #8 位一组
```

```
> python3 base64隐写.py
Base_sixty_four_point_five
```

功夫再高也怕菜刀

下载附件得到个pcapng文件，foremost一下有个压缩包，里面有flag.txt文件可是需要密码



先打开wireshark看一下，搜索下关键字

Wireshark interface showing a network capture. The packet list pane shows a series of HTTP requests and responses. The selected packet (No. 159) is an HTTP GET request for /flag.txt. The packet details pane shows the structure of the request, including the Hypertext Transfer Protocol section. The packet bytes pane shows the raw data of the request, including the header and body.

跟踪TCP流, 发现ffd8是jpg文件

Wireshark interface showing a TCP stream. The packet list pane shows a series of TCP segments. The selected packet (No. 884) is a TCP segment containing a POST request. The packet details pane shows the structure of the request, including the Hypertext Transfer Protocol section. The packet bytes pane shows the raw data of the request, including the header and body.

Wireshark · 追踪 TCP 流 (tcp.stream eq 7) · kongfu.pcapng

```
E6E895EFD1CE3BEE1A16E56EFF0093FB3725AAD9ED6B5ED4E7CD9D73211B864236CF98E78B583B281C0F3A5E3D4FDCDE0B1C614E4F18FDD92876FF00D3BC1FC523742659393073
93927F8940D09D876DFD0E71DCDC4F8E588C63CA4CE47B719C836316EA4186C991488F30F39B9883D2207826288921B18071C0C915E2D6776F5D175FEBF4F55FF2ED3F76845BE5
D2D6B74BEC058FCAFE92FF97866DC499DA23F93E53E583CF950FF0015C3E3F8DF395239C80470173912380015E073E486EA1792D7327B9C1DB9E98E06002D7AE0E37997E60597CD
23399A5FE1B78C81C2A1E085F438CFA2B1EE1B76F121C004198AF73CF36F1E381D006E78F70A09F1EBCADE9BFF5D2DE9B27E56BED1577F0F1DBFA5D34EEDBBF9369EBACE5CB4CC
D6F93FBA95B9FBC09C37B43E5E87AD1485AEBB480A0EC84E360FEE369FBBD3A9E9D68AF39D76FCA71FEFF9F005EF1E8AB69A2E9FF002F27DCECADD6D35D3747E89851B883
FBA3B079A40E2DAD7B429DCD20386EF927B965F4539CE7CB6F2F81DAD2D3D7D3CE973D00DC43702DC53F6189FFB0827F3357A4FF97DF00AF88ABF4C9E8ECA7E3BBE96FE5FBD
B7D22A3F90C3E28ABEFCB7F00B7BD9AEB7DB9BA774ACF795ED44C59028119084C59FF00976B7E775C38FF009EB27504F39395032B57A2193188C6D386F283D238FF5ADD0C9E
373632BF4D28CD613FA1FCA2AD15FF005D77F005E707FED3AE6A7A37E4DF96CF6F0005D76F76FD6A72A3ABF449F57AC9537DF5FE2D9F7BAFF182E5BC7B046539624FD98381
CB7FCB48C973C003076139E578E10E74214CED2ABDD77B1B70D9CDC4F9FDE5D49D3771F2541F978E7F8CD55EF37FD78A7FE8095A36FFEB17FEC187FF407ACA5A69D134BF14BCFB6
9FF6EAD938DC77F5FF0024FC9FDAD5FAC9E975CB6630BF2F0655F332B9EB7575FC4EDD7F731EE0F5F62ED888A010727CE53280DEB7976EEA2FAC116EC278F61B94082DADF87F
D7853CF93AAC5BFD0ED38FB95CFF00E852565255F64539CE7CB6F2F81DAD2D3D7D3CE973D00DC43702DC53F6189FFB0827F3357A4FF97DF00AF88ABF4C9E8ECA7E3BBE96FE5FBD
1B82718E7FB957E35CEFF349206D374E9FC6C3FD55943D781801F1C0C77DA3352C7EE99FF005D2E3F9AD695BFFABB3FFB083FFE4895495025DFF0058C3EFF8B5EFAF59366AD096
9D1AF2FD2D0B6B1BAE8E5FF228CE14E5D5D8B70BF6A65FF963171E5D945FED1E03E318E73F75B3A288CBB8C623611A89D97A5ADB10365BA0ED34A383FC47273D6435980FFC3B3FF00
B0937F4AD3FF009653FF00D8453F985734F6F7EE697E3FABB6EADA2E8BC2DFBDC55FF000D9D691BA767CD6A3505B24792DE593D322CECF8C0704CF2EE039DA496F56E34624276
95C447CB3E506FF974B5C1DD339E974D9E30339618077A014E4E9A97F7E45FFA1B05D6FBBA97D6B0BF00424AE5935D2FD5AD949EEBFB28F48AB2F7515177695ACEF15DD6AA974
7BFF0013ABE0E714A72156E3088043B210A43F6A18E887342CBE9E98FF005FD37F38EB5207F54881F7294FDEA8918C956B7B7D1EE1F231E4490B9F5725D54A46EB803CF352D3A
A1AD33FEB2E3FEC170FF00E8305613BDD0EBA296DEDA76F87E5EEF48A4F46D9E7FE5D0FBFF00135E5E5F3DE6ED3C4B9F2FCB18C173F6547E0CF3FF1DECDC93C2463B376471D70AFA
57E3190A5079C2D1FE8A575BEBBF390F3C9821270070B818FE2622A47F87F0829BF0041357ACDFD6A72A3ABF449F57AC9537DF5FE2D9F7BAFF182E5BC7B046539624FD98381
BF17A7B8CB6635C63E6994480363ADF51EC38CFD9E12C3230738FF006F8D28D000779F30193FD208FF0097BBA3CC76D177F261F9431CED2064754154ED7EF699FF005EB73FCA5A
D0B2EBA67FBD73FF00A11AC66F45A6FCB6F2E654DFCEDCFF003B3BFC4CABDB6E9F00927FFC83DAD66D35F046D6C293E198EF1B945DB27065973FBAB080F408884813D3A6157A30
8A3320975C2A70B19303CB9E90D8038310A43F6A18E887342CBE9E98FF005FD37F38EB5207F54881F7294FDEA8918C956B7B7D1EE1F231E4490B9F5725D54A46EB803CF352D3A
37BF2EAC9586F99BC6C60805C15CE206DBF59DA4E592CB9C32F392DB4F3E61AA4DBB7317C42DE50DD000B033D117D2E270C01FE23B079706B3DC497FEC2A9FCDDAA5CFCDBEF
FB0847C9A58A78ADFEB8375BEBBF390F3C9821270070B818FE2622A47F87F0829BF0041357ACDFD6A72A3ABF449F57AC9537DF5FE2D9F7BAFF182E5BC7B046539624FD98381
FF009697936060492632A47D464797562EFA6BFF5D6DBF9D55BBEBA97FD70B6FE71D7A30565726BEEBDD15F93B69A5BE1471CDBBBA371D375EF7B25F3B7D9FF00324EFAC9B2
B904794225ECCD66AC3EEA8E25BF988F452632718C7A2806038010A296CB30B357FF0096B2F00CB5BEB807F8136B1C718C7F0B93627FF5977F60B7FF00D92A197FD637FD81C7
A2F9AB00273B8F924A0F3319C5ADA71B224EE67973C8EFB807059F15D0B737DDDB1B797C71C59DA724B9E989E6DDECC77FAB8C5888F8979FF005F9CF9CEA85C70DBCF00FAEF157641
5925ABBF11FE47E7756B4797A7F5D7F69D0AFCEBBD6752FAEEA293DE5CD5CE36C1E59087C807A5DDB7592E64E4FEF64CFB8EA4E1C6D15EE7718FCBCA0E6D51F811C6
3FD65ECE7B310095201E808E1549BB7DFF00312FF76D77F69D52BAFF005B77FF005E16FF00CA2AEEA6C7FED9A76F756D14724DBB5EFBF7139079983839FB40D374553FF
9D940B276794BB8163F6457E3C61FEB6FA619E8A01DB888E063214E689DA36ED0641BCF900E7FD2AE3FE5A5D499FF0096711C85046DF9718BFB565FACDF60D8FF00401554AEAB
FF0060C97F9BD76D35B79E9B2F25D7D3492D526A5C7525A6D72FF8297FEDD66EDAB7292B36B968484301B49954C876E724DED9FBCE47FCF08893C93CFD58E2AB92436FEDEA993
F78464FDB2EB90B129186305C9C1C7CA474CE515607DEB3FF00AF588FE52D578BF6E1FF005CE7AECAGAE5F82CFD99FF00F27AF751D7E295F9A52B5FADBBF973BFBFDC95FFBB1
E5AD211F3EFF0098061F68653CD3E014B58881FEAD38DFB41C633D90567C843799E6390830F1D7C95FF0055650FAFD01B1D31CF90CDE1F76C3FBAACDF00A10ACDF0097783F
EBFDFF00F65AEDA6B48F5D99EBE06E7F1EADFEF759B6896A371BABBDD2EDF69EAB7B7C1A1745EE696824EACB8E40F2E12487E72E3836303C525B9F11631B65888679619003931
4AF1B0C33815E56CE774DE5E71938B4B53CE31FC53CA08E4E09DD88E04D27FAA97FEC27FD64AA97BFF002DFF5F917FEC5DDA40AEDD9733577286AF953B27FF0080E9D9C9856B
47979E527CA936DA8D9455F47574AF65AA4F2D5755149B779B9549189DA41F2C843E483D2D28925E67FF00A6D367823E63D41F99683252C4A6C014E18D88603F7518CF9B792F6F
30F3B4B1C82011C0506FDE74BEFAC1FCAB3EE7ADFF00FD7083F9C75D0F8D3874D7D6DCDE565FC3EDB85FC91B73D2F7A4BCF937D6DCFC97CECAADBD5D295FE39332E24C6C11839
25BECAC4F5E7CDB9B48DADB41C01E32139C694801447F38DCDE486F0096F3747B997B08D3076EEF61D9C56A5DFD8F8FB745FF0004EB1E6EABF79000E7FE42BCAD75A76F2
6B874AE97CB4E8BDD4EE9352F630E95A3E76FC545FFEDDD2DBCDAB3924C99B0A31FB05FEAC7D279723EF48D993930C648EBC08CE935893BE771399079983839FB40D374553FF
003CA33D80E79EEC00D49BEF41FF005E32FF0027ACA3D6D3FEB85CF00296BC6ACF5F9A57F7B3FFE4FE6959FC53E6F770E96AFB5FCB6E67D366DC3A689B86D2D236C8BA9000B8970
5BF798CE679C1F96043FF3CE3EA5BA7191FC158B70E4F98643BB0409D971F3B7FCB2B58F03855C7CDB41039FEEA83A72FDFB3FF7DFF0066AC83D2D3FEBEA4FF00D0C578F8896CBB
C53FBD53767DFF0089D775CDFCEDEEE192BFA36B69DAFE5F0E96DDBD8690465CF8326FE08C7DA197A20E365847D831C00DE9D201CE3D8B67866DADB71291FF2C1E8802FABB
F1B8638D0D3CEAD297FD5A7FD737F4AC8B8E92FF00D7E8FE46BC5AD276BFCFEFF7B57F87CE4FAA87BD865771E9ADBD2D251FD36D0E156693E6C9BA90E580C21D9866FF009F780F
1B467A4926E00FF176E0B3118731C7CD8D876128A738B780F2646FFA6B2678C1E49E30480356FEEDCCFF00D7D27F37ACAD47A5FF00FB07F235E3577A7B7F85FF001F7775B3756
B452F6B0DAB827BC9C75F5651DB6D39FE5E6F8526F0EE1CB30D9F28018C00E3F751F5789931FF002D1B191E99E32319C999F681B7A90C6156C6069D73276E31F2E788183F2F37
EE7EFDCCFD7383F9C758F77D65FF00AF78FF00F408EBC6C4DD4ADE4F54EDBD97BBA2E9A7F2A3E830E45E97D7DA1EBD5FC7AF77BAC9B59E658327293B9CF2E1BEF7FB5FF02EBF
8D15763FF571FF00B8BFA08A2B86D0FE1FFC0579793FBFEFE9E65DA5FF81BFF002F5FE968FFD9HTTP/1.1 200 OK
```

分组 1088, 53 客户端 分组, 2 服务器 分组, 3 转码 点击选择.

整个对话 (206kB) Show data as ASCII 流 7

查找: ffd9 查找下一个

Help 滤掉此流 打印 另存为... 返回 Close

https://blog.csdn.net/qq_45924653

FFD8到最后一个FFD9复制到HexFiend, ASCII形式保存位.jpg文件


```

gongf.jpg
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13
00000 FF D8 FF E0 00 10 4A 46 49 46 00 01 01 01 00 78 00 78 00 00 ..... JFIF      x x
00014 FF DB 00 43 00 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 .. C
00028 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
0003C 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
00050 01 01 01 01 01 01 01 01 01 01 01 FF DB 00 43 01 01 01 01 01 01 01 01 .. C
00064 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
00078 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
0008C 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 FF C0 ..
000A0 00 11 08 01 39 01 E2 03 01 22 00 02 11 01 03 11 01 FF C4 00 9 . " ..
000B4 1F 00 00 01 05 01 01 01 01 01 01 00 00 00 00 00 00 00 01
000C8 02 03 04 05 06 07 08 09 0A 0B FF C4 00 B5 10 00 02 01 03 03 .....
000DC 02 04 03 05 05 04 04 00 00 01 7D 01 02 03 00 04 11 05 12 21 } !
000F0 31 41 06 13 51 61 07 22 71 14 32 81 91 A1 08 23 42 B1 C1 15 1A Qa "q 2... #B..
00104 52 D1 F0 24 33 62 72 82 09 0A 16 17 18 19 1A 25 26 27 28 29 R..$3br.      %&'()
00118 2A 34 35 36 37 38 39 3A 43 44 45 46 47 48 49 4A 53 54 55 56 *456789:CDEFGHIJSTUV
0012C 57 58 59 5A 63 64 65 66 67 68 69 6A 73 74 75 76 77 78 79 7A WXYZcdefghijstuvwxyz
00140 83 84 85 86 87 88 89 8A 92 93 94 95 96 97 98 99 9A A2 A3 A4 .....
00154 A5 A6 A7 A8 A9 AA B2 B3 B4 B5 B6 B7 B8 B9 BA C2 C3 C4 C5 C6 .....
00168 C7 C8 C9 CA D2 D3 D4 D5 D6 D7 D8 D9 DA E1 E2 E3 E4 E5 E6 E7 .....
0017C E8 E9 EA F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FF C4 00 1F 01 00 03 .....
00190 01 01 01 01 01 01 01 01 01 00 00 00 00 00 00 00 01 02 03 04 05
001A4 06 07 08 09 0A 0B FF C4 00 B5 11 00 02 01 02 04 04 03 04 07 .....
001B8 05 04 04 00 01 02 77 00 01 02 03 11 04 05 21 31 06 12 41 51 w !1 AQ
001CC 07 61 71 13 22 32 81 08 14 42 91 A1 B1 C1 09 23 33 52 F0 15 aq "2. B.... #3R.
001E0 62 72 D1 0A 16 24 34 E1 25 F1 17 18 19 1A 26 27 28 29 2A 35 br. $4.%      &'()*5
001F4 36 37 38 39 3A 43 44 45 46 47 48 49 4A 53 54 55 56 57 58 59 6789:CDEFGHIJSTUVWXY
00208 5A 63 64 65 66 67 68 69 6A 73 74 75 76 77 78 79 7A 82 83 84 Zcdefghijstuvwxyz...
102226 out of 102226 bytes
https://blog.csdn.net/qq_45924653

```

打开图片这应该就是解压密码



解压flag.txt中得到flag

人生漫漫其修远兮，网安无止境。
一同前行，加油！