

CTF平台题库writeup（三）--BugKuCTF-杂项（1-20题详解）

原创

Hacking黑白红 于 2020-06-29 09:36:17 发布 4226 收藏 18

分类专栏: [CTF 信息安全 bugku](#) 文章标签: [安全 信息安全](#)

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/zsw15841822890/article/details/107013549>

版权



[CTF 同时被 3 个专栏收录](#)

15 篇文章 6 订阅

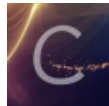
订阅专栏



[信息安全](#)

39 篇文章 8 订阅

订阅专栏



[bugku](#)

7 篇文章 2 订阅

订阅专栏

一、BugKuCTF-杂项(1-20)

| | | | |
|----------------------|---------------------|-------------------|--------------|
| 签到题 50 | 这是一张单纯的图片 50 | 隐写 50 | telnet 50 |
| 眼见非实(ISCCCTF) 50 | 啊哒 50 | 又一张图片, 还单纯吗 60 | 猜 60 |
| 宽带信息泄露 60 | 隐写2 60 | 多种方法解决 60 | 闪的好快 60 |
| come_game 60 | 白哥的鸽子 60 | linux 80 | 隐写3 80 |
| 做个游戏(08067CTF) 80 | 想蹭网先解开密码 100 | Linux2 100 | 账号被盗了 100 |
| 细心的大象 100 | 爆照(08067CTF) 100 | 猫片(安恒) 100 | 多彩 100 |
| 旋转跳跃 100 | 普通的二维码 100 | 乌云邀请码 100 | 神秘的文件 100 |

<https://blog.csdn.net/zsw15841822890>

1、签到题

扫描二维码直接获取flag

2、这是一张单纯的图片

<http://120.24.86.145:8002/misc/1.jpg>



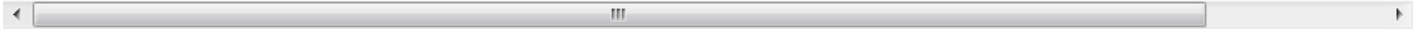
(1) .启蒙级隐写术

linux下保存到本地当作文本文件cat一下（或记事本、winhex等打开）

```
cat 1.jpg
```

(2) .在末尾发现

```
&#107;&#101;&#121;&#123;&#121;&#111;&#117;&#32;&#97;&#114;&#101;&#32;&#114;&#105;&#103;&#104;&#103;&#104;&#116;&#125;
```



unicode解码得到flag

利用网页上搜索的unicode转ascii进行转换即得FLAG！（工具可以用小葵花）

```
key{you are right}
```



3、隐写(文件头)

Linux下提示

Bu

下载2.rar，解压得到一张图片，首先放在winhex里看看

```

2.png | CTF.png |
Offset  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00000000  89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52  tENG  IHER
00000010  00 00 01 F4 00 00 01 84 08 06 00 00 00 CB D6 DF  é  ÈCB
00000020  8A 00 00 00 09 70 48 59 73 00 00 12 74 00 00 12  ð  pHyS t
00000030  74 01 DE 66 1F 78 00 00 0A 4D 69 43 43 50 50 68  t Ff x MiCCFFh
00000040  6F 74 6F 73 68 6F 70 20 49 43 43 20 70 72 6F 66  ctcshcp ICC prof
00000050  69 6C 65 00 00 78 DA 9D 53 77 58 93 F7 16 3E DF  ile xÜ SwX"= >B
00000060  F7 65 0F 56 42 D8 F0 B1 97 6C 81 00 22 23 AC 08  te VEGGz-1 "#-
00000070  C8 10 59 A2 10 92 00 61 84 10 12 40 C5 85 88 0A  È Y+ ' a, @Ä_-
00000080  56 14 15 11 9C 48 55 C4 82 D5 0A 48 9D 88 E2 A0  V αHUA,È H 'd
00000090  28 B8 67 41 8A 88 5A 8B 55 5C 38 EE 1F DC A7 B5  (.gAŠ"2<U\šİ ÜSp
000000A0  7D 7A EF ED ED FB D7 FB BC E7 9C E7 FC CE 79 CF  )zifiİ×Ü4ççüİyİ
000000B0  0F 80 11 12 26 91 E6 A2 6A 00 39 52 85 3C 3A D8  € &'a+j sR...:ø
000000C0  1F 8F 4F 48 C4 C9 BD 80 02 15 48 E0 04 20 10 E6  CHÄÈ+È Hå æ
000000D0  CB C2 67 05 C5 00 00 F0 03 79 78 7E 74 B0 3F FC  ÈÄg Ä é yx-t°ü
000000E0  01 AF 6F 00 02 00 70 D5 2E 24 12 C7 E1 FF 83 BA  "c pĈ.š Çdyf°
000000F0  50 26 57 00 20 91 00 E0 22 12 E7 0B 01 90 52 00  F6# 'ä" ç F
00000100  C8 2E 54 C8 14 00 C8 18 00 B0 53 B3 64 0A 00 94  È.IÈ È "š'd"
00000110  00 00 6C 79 7C 42 22 00 AA 0D 00 EC F4 49 3E 05  İy|E" " İdİ>

```

89 50 4E 47PE头是png照片的，就是说没有可能照片中嵌入了Exif信息
 在查看PNG文件格式时，IHDR后面的八个字节就是宽高的值

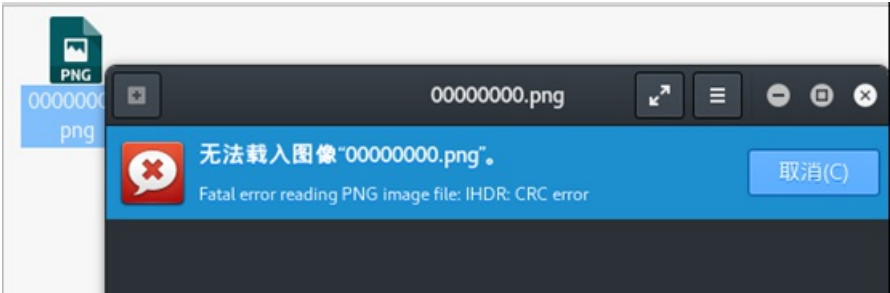
IHDR

文件头数据块IHDR(header chunk): 它包含有PNG文件中存储的图像数据的基本信息，并要作为第一个数据块出现在PNG数据流中，而且一个PNG数据流中只能有一个文件头数据块。

文件头数据块由13字节组成，它的格式如下表所示。

| 域的名称 | 字节数 | 说明 |
|--------------------|---------|--|
| Width | 4 bytes | 图像宽度，以像素为单位 |
| Height | 4 bytes | 图像高度，以像素为单位 |
| Bit depth | 1 byte | 图像深度： 索引彩色图像：1, 2, 4或8 灰度图像：1, 2, 4, 8或16 真彩色图像：8或16 |
| ColorType | 1 byte | 颜色类型： 0: 灰度图像, 1, 2, 4, 8或16 2: 真彩色图像, 8或16 3: 索引彩色图像, 1, 2, 4或8 4: 带α通道数据的灰度图像, 8或16 6: 带α通道数据的真彩色图像, 8或16 |
| Compression method | 1 byte | 压缩方法(LZ77派生算法) |
| Filter method | 1 byte | 滤波器方法 |
| Interlace method | 1 byte | 隔行扫描方法： 0: 非隔行扫描 1: Adam7(由Adam M. Costello开发的7遍隔行扫描方法) |

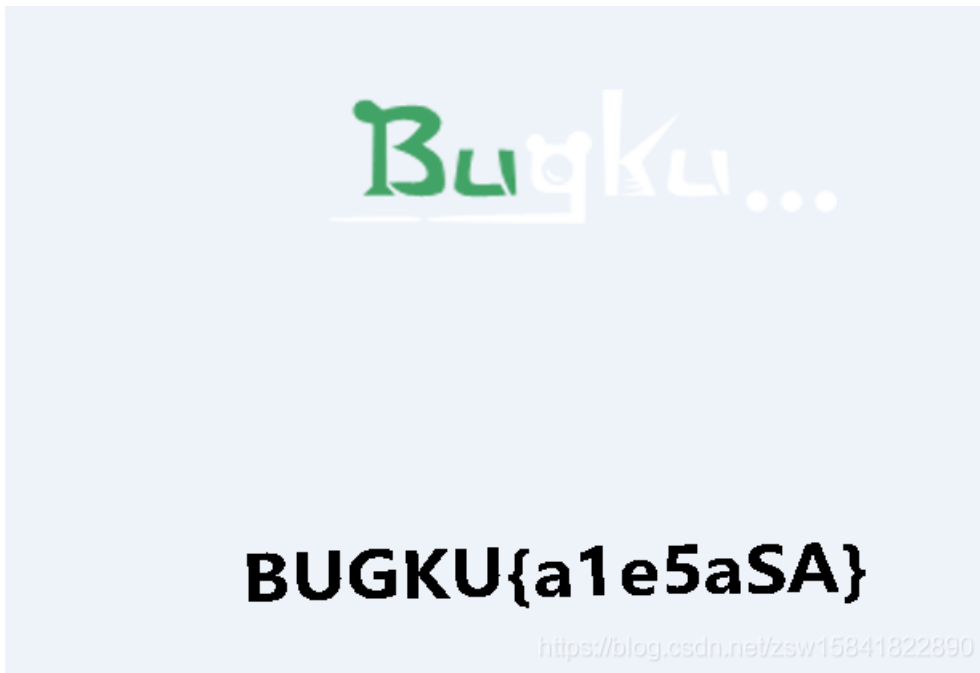
将图片放在Linux下，发现是打不开的，说明图片被截了



将图片的高改成和宽一样,即将A4改成F4，然后另存为

| Offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------------------|
| 00000000 | 89 | 50 | 4E | 47 | 0D | 0A | 1A | 0A | 00 | 00 | 00 | 0D | 49 | 48 | 44 | 52 | %ENG IHDR |
| 00000010 | 00 | 00 | 01 | F4 | 00 | 00 | 01 | F4 | 08 | 06 | 00 | 00 | 00 | CB | D6 | DF | é é ěB |
| 00000020 | 8A | 00 | 00 | 00 | 09 | 70 | 48 | 59 | 73 | 00 | 00 | 12 | 74 | 00 | 00 | 12 | š pHYs t |
| 00000030 | 74 | 01 | DE | 66 | 1F | 78 | 00 | 00 | 0A | 4D | 69 | 43 | 43 | 50 | 50 | 68 | t ěf x MiCCFFh |
| 00000040 | 6F | 74 | 6F | 73 | 68 | 6F | 70 | 20 | 49 | 43 | 43 | 20 | 70 | 72 | 6F | 66 | ctoshop ICC prof |
| 00000050 | 69 | 6C | 65 | 00 | 00 | 78 | DA | 9D | 53 | 77 | 58 | 93 | F7 | 16 | 3E | DF | ile xŮ SwX"÷ >B |
| 00000060 | F7 | 65 | 0F | 56 | 42 | D8 | F0 | B1 | 97 | 6C | 81 | 00 | 22 | 23 | AC | 08 | ÷e VE00z-1 "#- |
| 00000070 | C8 | 10 | 59 | A2 | 10 | 92 | 00 | 61 | 84 | 10 | 12 | 40 | C5 | 85 | 88 | 0A | È Yé ' a„ @A...^ |
| 00000080 | 56 | 14 | 15 | 11 | 9C | 48 | 55 | C4 | 82 | D5 | 0A | 48 | 9D | 88 | E2 | AO | v.n.eHUA,Č H 11á22890 |
| 00000090 | 28 | 88 | 67 | 41 | 82 | 88 | 52 | 88 | 55 | 5C | 38 | EE | 1F | DC | 27 | 8E | / adě^2,PAŠf fičn |

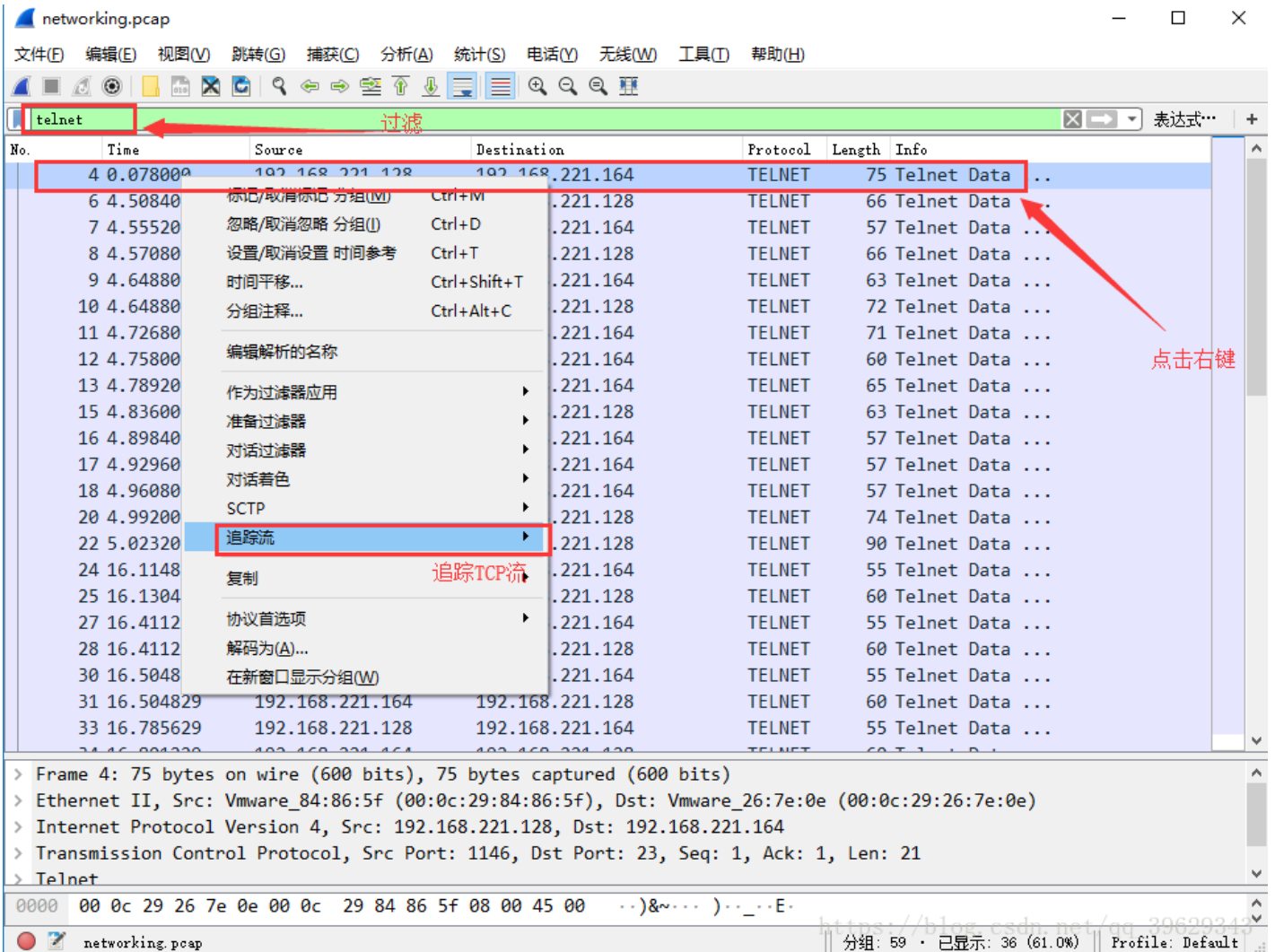
打开刚存的图片就可以得到FLAG了



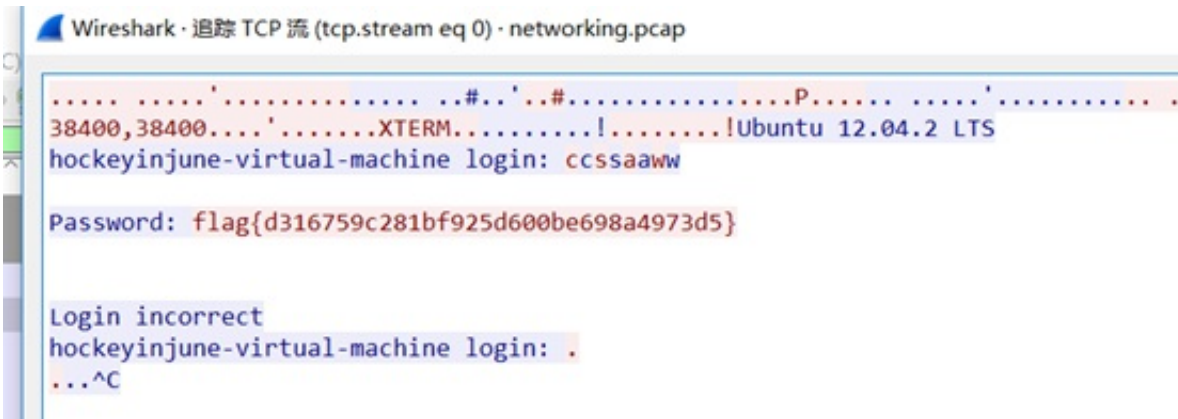
BUGKU{a1e5aSA}

4.telnet

将1.zip下载解压得到一个流量包文件，放到Wireshark中走一遍，因为提示的是telnet,所以使用规则显示telnet的包，然后追踪tcp流

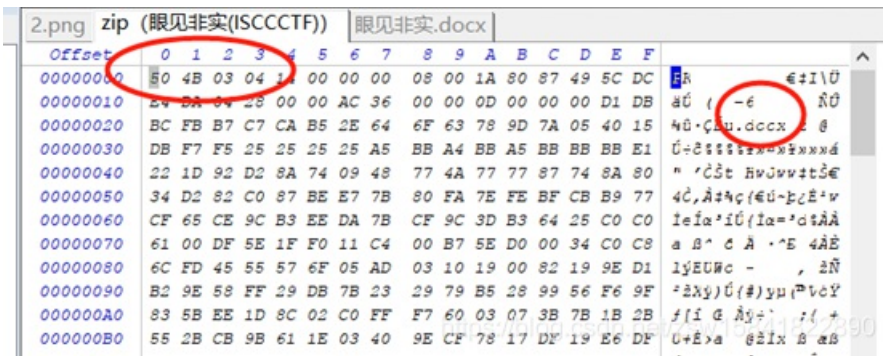


在tcp流中就能直接看到flag,在telnet协议下任意数据流都能看到flag

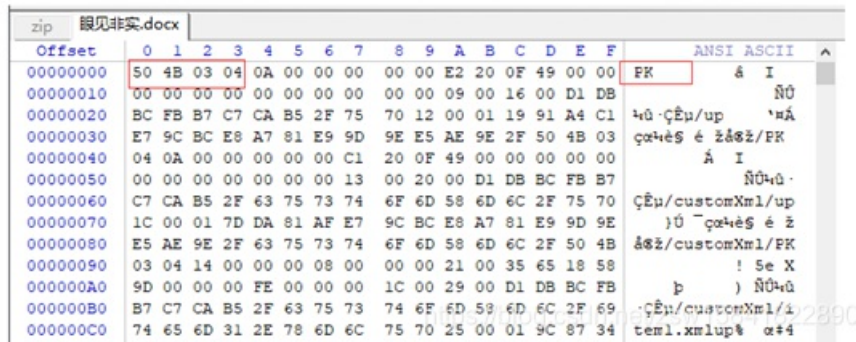


5、眼见非实(ISCCCTF)

下载下来是一个文件的格式,放到winhex中,发现有50 4B 03 04这是压缩文件的头,还有.docx格式文件,应该压缩包有一个文档,改文件后缀为.zip,解压得到文档



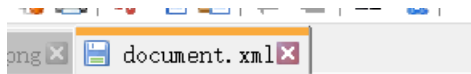
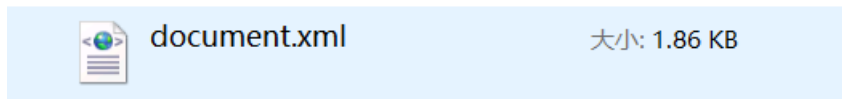
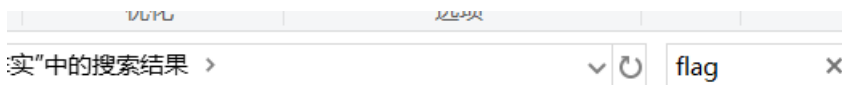
得到眼见非实.docx是打不开的，放到winhex中发现还是zip格式的文件



继续改后缀为.zip，然后解压得到一个文件夹



检索flag, 然后在word->document.xml中找到了flag



<w:rPr><w:t>flag{F1@g}</w:t></w:r>

【注】：

方法二

Kailli 下执行binwalk, 和foremost命令 可以不用修改文件扩展名


```
root@kali:~/桌面# binwalk ada.jpg
DECIMAL      HEXADECIMAL  DESCRIPTION
-----
0            0x0         JPEG image data, JFIF standard 1.01
30          0x1E       TIFF image data, big-endian, offset of first image direct
ory: 8
5236        0x1474     Copyright string: "Copyright Apple Inc., 2018"
7782        0x1E66     Unix path: /www.w3.org/1999/02/22-rdf-syntax-ns#"/></x:xm
pmeta>
218773      0x35695    Zip archive data, encrypted at least v2.0 to extract, com
pressed size: 34, uncompressed size: 22, name: flag.txt
218935      0x35737    End of Zip archive

root@kali:~/桌面# foremost ada.jpg
Processing: ada.jpg
|foundat=flag.txt?0n0000D;5jV00u0000-0Z0LI0000
*|
root@kali:~/桌面#
```

<https://blog.csdn.net/zsw15841822890>

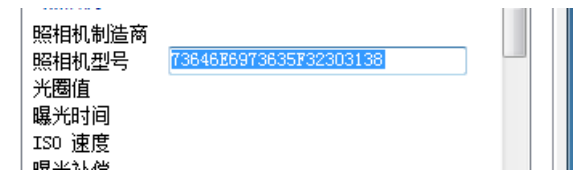
内含一个zip压缩包，分离文件：

```
root@kali:~/桌面# foremost ada.jpg
Processing: ada.jpg
|foundat=flag.txt?0n0000D;5jV00u0000-0Z0LI0000
*|
root@kali:~/桌面# dd if=ada.jpg of=ada.zip skip=218773 bs=1
记录了184+0 的读入
记录了184+0 的写出
184 bytes copied, 0.0297407 s, 6.2 kB/s
root@kali:~/桌面#
```

压缩包需要密码：

```
root@kali:~/桌面# unzip ada.zip
Archive:  ada.zip
[ada.zip] flag.txt password: 
```

回去看看源图片文件ada.jpg的详细信息



16进制，转化成字符串（Notepad++中的插件，直接16进制转ASCII）：

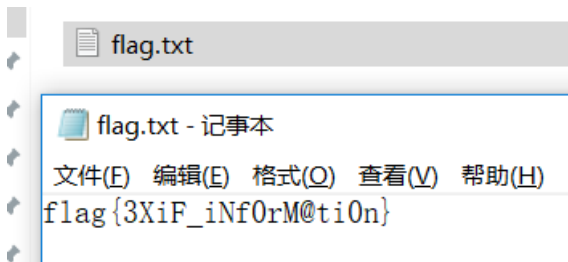
73646E6973635F32303138



<https://blog.csdn.net/zsw15841822890>

有疑问，为什么在小葵花里解密密码是 dnisc_2018，网上wp上是sdnisc_2018

得到解压密码： sdnisc_2018



分离文件

(1) 使用dd命令分离(linux/unix下)

我们可以使用dd命令分离出隐藏文件：

```
# dd if=carter.jpg of=carter-1.jpg skip=140147 bs=1
```

if是指定输入文件，of是指定输出文件，skip是指定从输入文件开头跳过140147个块后再开始复制，bs设置每次读写块的大小为1字节。

dd命令：<http://www.cnblogs.com/qq78292959/archive/2012/02/23/2364760.html>

(2) 使用foremost工具分离

foremost是一个基于文件文件头和尾部信息以及文件的内建数据结构恢复文件的命令行工具，直接将文件拆解

7. 又一张图片，还单纯吗



图片隐写，套路化，notepad++、winhex、右键属性、binwalk、foremost打一遍

Binwalk和foremost问题解决

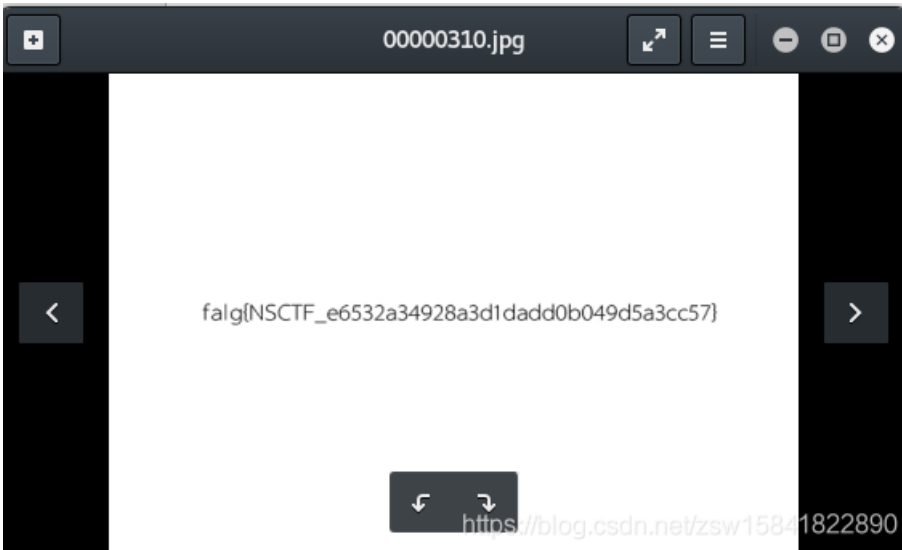
```

root@kali:~/桌面# binwalk 2.jpg
DECIMAL      HEXADECIMAL      DESCRIPTION
-----
0            0x0              JPEG image data, EXIF standard
12          0xC              TIFF image data, big-endian, offset of first image direct
ory: 8
13017       0x32D9           Unix path: /www.w3.org/1999/02/22-rdf-syntax-ns#"> <rdf:D
escription rdf:about="" xmlns:photoshop="http://ns.adobe.com/photoshop/1.0/" xmlns
158792     0x26C48           JPEG image data, JFIF standard 1.02
158822     0x26C66           TIFF image data, big-endian, offset of first image direct
ory: 8
159124     0x26D94           JPEG image data, JFIF standard 1.02
162196     0x27994           JPEG image data, JFIF standard 1.02
164186     0x2815A           Unix path: /www.w3.org/1999/02/22-rdf-syntax-ns#"> <rdf:D
escription rdf:about="" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:xap="htt
168370     0x291B2           Copyright string: "Copyright (c) 1998 Hewlett-Packard Com
pany"

root@kali:~/桌面# foremost 2.jpg
Processing: 2.jpg
|*|
root@kali:~/桌面# date
2020年 06月 28日 星期日 08:25:52 CST
https://blog.csdn.net/zsw15841822890

```

Foremost后out文件夹下看到



falg{NSCTF_e6532a34928a3d1dadd0b049d5a3cc57}

8、猜

flag格式key{某人名字全拼}



百度识图:



Key{liuyifei}

9、宽带信息泄露

flag格式: flag{宽带用户名}

使用RouterPassView工具查看,

大多数现代路由器都可以让您备份一个文件路由器的配置文件,然后在需要的时候从文件中恢复配置。路由器的备份文件通常包含了像您的ISP的用户名重要数据/密码,路由器的登录密码,无线网络的KEY。

如果你忘记了这些密码,但你仍然有你的路由器配置的备份文件,那么RouterPassView可以帮助你从路由器配置文件中恢复您的密码。该软件可以读取这个路由配置文件。

运行软件,打开"文件"菜单->"打开路由器配置文件",打开保存的路由器配置文件。
或者Grab Password From IE Window。

```
RouterPassView - C:\Users\bbjz\Desktop\BUGKU\9-(宽带信息泄露)\conf.bin
文件(F) 编辑(E) 查看(V) 选项(O) 帮助(H)
<MACAddress val=D0:C7:C0:43:53:69 />
<X_TP_IFName val=eth1 />
</WANIPConnection>
<WANIPConnection nextInstance=3 />
<WANPPPConnection instance=1 >
  <Enable val=1 />
  <DefaultGateway val=10.177.144.1 />
  <Name val=pppoe_eth1_d />
  <Uptime val=671521 />
  <Username val=053700357621 />
  <Password val=210265 />
  <X_TP_IFName val=ppp0 />
```

Flag{ 053700357621}

注: 秘钥破解工具

Aircrack-ng kaili下自带

命令: Aircrack-ng miyao.ivs (2017年9月部科信局在线测试题)

10、隐写2

Welcome_.jpg



想拿到flag? 心の中ないいくつかB数かの?

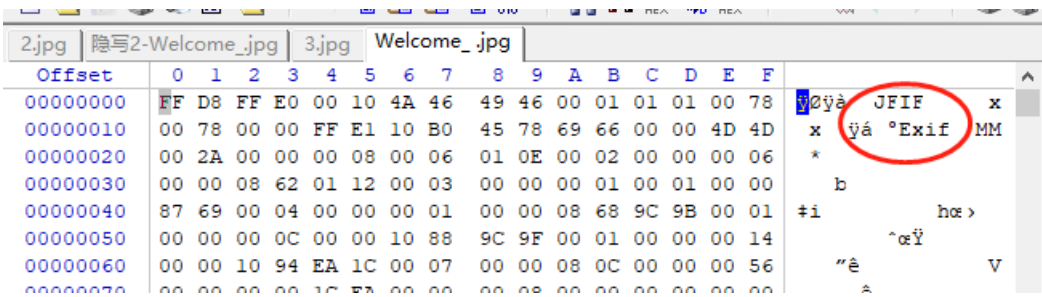
图片隐写, 套路化, notepad++, winhex, 右键属性, binwalk, foremost打一遍

Notepad++

```
DBETX\xFE {瘥\xE 媵銕\xAF? 恣疹[SYN  
掬F<珪NUL镜DC4P! \xF1SYN] 殒名n\xC4-  
NULflag.rarPKEx NULNULflag.rar
```

搜索flag发现文件 BTX鶇tEOM/ 藜\xFF ʔ?NULDC4NULNUL

Winhex打开



可以看到图片嵌入了Exif信息, 但是看属性没看到什么有用的提示, 老方法放到kali里找

使用binwalk提取

告诉你们一个秘密，密码是3个数哦。

查理曼：

查理曼，法兰克王国国王，征服了西欧与中欧大部分土地，具有了至高无上的权威，下令全国人民信仰基督教，查理重振了西罗马帝国。

雅典娜：

女神帕拉斯·雅典娜，是希腊神话中的女战神也是智慧女神，雅典是以她命名的。

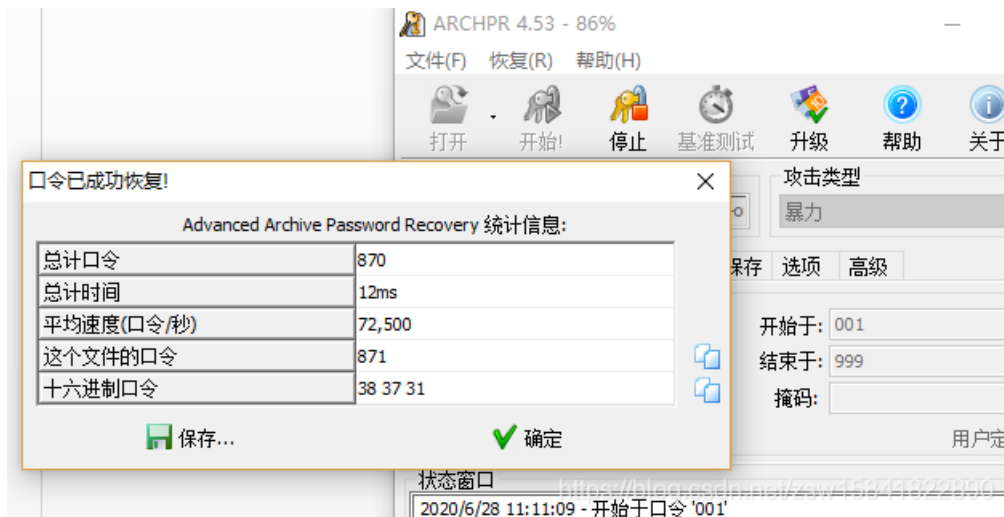
兰斯洛特，

英格兰传说中的人物，是亚瑟王圆桌骑士团中的一员。看上去就是一个清秀年轻的帅小伙儿，由于传说中他是一名出色的箭手，因此梅花J手持箭支。兰斯洛特与王后的恋爱导致了他与亚瑟王之间的战争。

Hint:

其实斗地主挺好玩的。
<https://blog.csdn.net/zsw15841822890>

爆破flag.zip,工具 Advanced Archive Password Recovery



压缩包密码：871

解压后图片



Winhex打开

| | | |
|----------|---|------------------|
| 00001A00 | 76 83 BE 19 02 12 19 85 DD F5 2F 71 D9 F8 ED F8 | vf% ...Y8/qÙøiø |
| 00001A10 | D6 32 7B 25 E4 F1 53 17 8C 80 50 37 D7 1D BF 9C | Ö2{ññS GEP7× çæ |
| 00001A20 | A0 2E B0 29 AC A6 B1 AD 38 00 A3 62 CF 8C 69 6D | .°)~;±-8 ðbİGim |
| 00001A30 | CB 15 9F 6F 6C A0 86 25 6E 12 70 EB BC 69 6B 41 | È Yø1 t%n pè+ikA |
| 00001A40 | 23 E4 67 D4 FF D9 20 20 20 20 66 31 40 67 7B 65 | #ägÖÛ fl@g{e |
| 00001A50 | 54 42 31 49 45 46 79 5A 53 42 68 49 47 68 41 59 | TB1IEFyZSBhIGhAY |
| 00001A60 | 32 74 6C 63 69 45 3D 7D 20 20 20 20 20 0D 0A 20 | 2t1ciE=} |
| 00001A70 | 1A | |

密文eTB1IEFyZSBhIGhAY2t1ciE=

带“=”

base64解密，工具小葵花



fl@g{ y0u Are a h@cker!}

11、多种方法解决

提示：在做题过程中你会得到一个二维码图片

使用winhex打开，发现是一个base64转图片，所以先将后缀改为.txt,然后将base64编码为图片（base64转换为图片工具）

| Offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | ANSI ASCII |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------------------------------|
| 00000000 | 64 | 61 | 74 | 61 | 3A | 69 | 6D | 61 | 67 | 65 | 2F | 6A | 70 | 67 | 3B | 62 | data:image/jpg;base64,iVBCRw0KGg |
| 00000010 | 61 | 73 | 65 | 36 | 34 | 2C | 69 | 56 | 42 | 4F | 52 | 77 | 30 | 4B | 47 | 67 | oAAAAANSUnEUGAAAl |
| 00000020 | 6F | 41 | 41 | 41 | 41 | 4E | 53 | 55 | 68 | 45 | 55 | 67 | 41 | 41 | 41 | 49 | UAAACFCAYAAAB12j |
| 00000030 | 55 | 41 | 41 | 41 | 43 | 46 | 43 | 41 | 59 | 41 | 41 | 41 | 42 | 31 | 32 | 6A | s8AAAAAXNSR0IArs |
| 00000040 | 73 | 38 | 41 | 41 | 41 | 41 | 41 | 58 | 4E | 53 | 52 | 30 | 49 | 41 | 72 | 73 | 4c6QAAAArnQU1BAA |
| 00000050 | 34 | 63 | 36 | 51 | 41 | 41 | 41 | 41 | 52 | 6E | 51 | 55 | 31 | 42 | 41 | 41 | Cxjwv8YQUAAAjce |
| 00000060 | 43 | 78 | 6A | 77 | 76 | 38 | 59 | 51 | 55 | 41 | 41 | 41 | 41 | 4A | 63 | 45 | hZcwAADsMAAA7DAc |
| 00000070 | 68 | 5A | 63 | 77 | 41 | 41 | 44 | 73 | 4D | 41 | 41 | 41 | 37 | 44 | 41 | 63 | dvqGQAAArZSURBVH |
| 00000080 | 64 | 76 | 71 | 47 | 51 | 41 | 41 | 41 | 72 | 5A | 53 | 55 | 52 | 42 | 56 | 48 | he7ZKBitxIFgTv/3 |
| 00000090 | 68 | 65 | 37 | 5A | 4B | 42 | 69 | 74 | 78 | 49 | 46 | 67 | 54 | 76 | 2F | 33 | 96Tx564G1UouicKg |
| 000000A0 | 39 | 36 | 54 | 78 | 35 | 36 | 34 | 47 | 31 | 55 | 6F | 75 | 69 | 63 | 4B | 67 | 19hwPCDcrMJ9m7/7 |
| 000000B0 | 31 | 39 | 68 | 77 | 50 | 43 | 44 | 63 | 72 | 4D | 4A | 39 | 6D | 37 | 2F | 37 | n45zfdxe5Z3sJ7pr |
| 000000C0 | 6E | 34 | 35 | 7A | 66 | 64 | 78 | 65 | 35 | 5A | 33 | 73 | 4A | 37 | 70 | 72 | Hbf9rXO3P41LvYPc |
| 000000D0 | 48 | 62 | 66 | 39 | 72 | 58 | 4F | 33 | 50 | 34 | 6C | 4C | 76 | 59 | 50 | 63 | tbeM80dvtP+3pnDp |
| 000000E0 | 74 | 62 | 65 | 4D | 38 | 30 | 64 | 76 | 74 | 50 | 2B | 33 | 70 | 6E | 44 | 70 | 9yF7tneQvvcZu/2 |
| 000000F0 | 39 | 79 | 46 | 37 | 74 | 6E | 65 | 51 | 76 | 76 | 6D | 63 | 5A | 75 | 2F | 32 | 1f78zhU+5i9yvx4T |
| 00000100 | 6C | 66 | 37 | 38 | 7A | 68 | 55 | 2B | 35 | 69 | 39 | 79 | 78 | 76 | 34 | 54 | 3T200/7eud68OT2H |
| 00000110 | 33 | 54 | 32 | 4F | 30 | 2F | 37 | 65 | 75 | 64 | 36 | 38 | 4F | 54 | 32 | 48 | 3LCft01/ae9Z1To+ |
| 00000120 | 33 | 4C | 43 | 66 | 74 | 30 | 6C | 2F | 61 | 65 | 39 | 5A | 6C | 54 | 6F | 2B | 23pPvX7/rwJHbfcs |
| 00000130 | 32 | 33 | 70 | 50 | 76 | 58 | 37 | 2F | 72 | 77 | 4A | 48 | 62 | 66 | 63 | 73 | I+3aW9Z33m1Gj7Le |
| 00000140 | 49 | 2B | 33 | 61 | 57 | 39 | 5A | 33 | 33 | 6D | 31 | 47 | 6A | 37 | 4C | 65 | n+9bs+PIndt5ywT3 |
| 00000150 | 6E | 2B | 39 | 62 | 73 | 2B | 50 | 49 | 6E | 64 | 74 | 35 | 79 | 77 | 54 | 33 | dp7lmfOTXafku6f/ |
| 00000160 | 64 | 70 | 37 | 31 | 6D | 66 | 4F | 54 | 58 | 61 | 66 | 6B | 75 | 36 | 66 | 2F | 2uD09i9y0n7NNd2n |
| 00000170 | 32 | 75 | 44 | 30 | 39 | 69 | 39 | 79 | 30 | 6E | 37 | 4E | 4E | 64 | 32 | 6E | vWZ06Ntt+S71+/68 |
| 00000180 | 76 | 57 | 5A | 30 | 36 | 4E | 74 | 74 | 2B | 53 | 37 | 6C | 2B | 2F | 36 | 38 | MJc500CSWpcyexnF |
| 00000190 | 4D | 4A | 63 | 35 | 4F | 30 | 4F | 53 | 57 | 70 | 63 | 79 | 65 | 78 | 6E | 46 | jfcsI+JWlukpRfv+ |
| 000001A0 | 6A | 66 | 63 | 73 | 49 | 2B | 4A | 57 | 31 | 75 | 6B | 70 | 52 | 66 | 76 | 2B | vDCXOTtDklqXMnsZ |
| 000001B0 | 76 | 44 | 43 | 58 | 4F | 54 | 74 | 44 | 6B | 6C | 71 | 58 | 4D | 6E | 73 | 5A | xY33LCPiVtbpKUX7 |
| 000001C0 | 78 | 59 | 33 | 33 | 4C | 43 | 50 | 69 | 56 | 74 | 62 | 70 | 4B | 55 | 58 | 37 | /rwwlzk7Q5JalzJ7 |
| 000001D0 | 2F | 72 | 77 | 77 | 6C | 7A | 6B | 37 | 51 | 35 | 4A | 61 | 6C | 7A | 4A | 37 | GcWN9ywj41bw6S1F |
| 000001E0 | 47 | 63 | 57 | 4E | 39 | 79 | 77 | 6A | 34 | 6C | 62 | 57 | 36 | 53 | 6C | 46 | |

改扩展名为txt


data:image/jpeg;base64,iVBORw0KGGoAAAANSUUhEUgAAAIUAAACFCAYAAAB12js8AAAAAXNfdxe5Z3sJ7prHbf9rXO3P4llvYPctbeM80dvtP+3pnDp9yF7tneQvwmcZu/2lf78zhU+5i9yvx4T3Tz+/68MJc5O0OSWpcyexnFjfcsl+JW1ukpRfv+vDCXOTtDklqXMnsZxY33LCPiVtbpKUX7/rwwlzk+nDC3CSWk7a/i73PctL2DbvH3CQpv37XhxPmJrGctP1d7H2Wk7Zv2D3mJkn59bs+nDA3ieWE1+fW7PjzJ7v12b33LSDtvsfuW75LuX7/rw5Ps3m/31rectP0Wu2/5Lun+9bs+PMnu/XZvfctJ22+x+yfzTmppE2U2I5YZ9+lh/3VfaPxtw00mZKLCfs00/k477K/tGYm0baTlnlhH36iSxflT78TPl605bdlUmdKUmdk5LUaXzdWB/eYX3LCfuUpM6UtDklqTmIqXNSkjqNrxvrwzusbzlhn5LUmZl2pyR1pil+pYT9k0ibaa7pHt6NY3uYJ8Svw3uaWF9vwn7ppE2013SPb2aRnewT4nlBvfUsL7lhH3TSJvoLunc

base64编码 X 转图片

```

osj+xt1vkuauc9ygd1qpm1zdc15umkNessj+5K1vovut9zgnmpcy155b60wKLECSI+JZbV1vCvN7imkrCSXoSLyZWKLGCSE+J5bV1CSIN7qmkknnesH+RLK50mb
W4Sywn7IOzmhH3a0u7ZN99hadmRNjeJ5YR9SnZzwj5taffsm++wtOxlm5vEcl+Jbs5YZ+2tHv2zXdYwNakzU1iOWGfkt2csE9b2j375jtcvTz+tuX0vrXF9sxNk
jrTT+T6rvyx37ac3re22J65SVJn+olc35U/9tuW0/vWFtszN0nqTD+R67vyx37bcnrf2mJ75iZJneknUn+V/aWYUyNtpqTNqZE2UyNtGlvSjTsT9VvtKHNqpm2Ut
Dk10mZqpE1jS7pxZ6j+qx1ITo20mZl2p0baTi20aWxJN+5M1G+1o8ypkTZT0ubUSJupkTaNLengnYnl6TujO2zP3DTSZkp2c8L+0xppM32HpfWTixPbMzeNt
JmS3Zyw/7RG2kzfYwN95MjE9sxNI22mZDcn7D+tkTbTd1haPzkysT1z00ibKdnNCftPa6TN9B2uXh5/S9rcbEk37jR2+5SkzpSkzo4kdaavTg6/JW1utqQbdxq7
fUpSZ0pSZ0eSOtNXJ4ffkjY3W9KNO43dPiWpMyWpsYnJnemrk8NvSzubLenGncZun5LUmZLU2ZGkzvTVWR/e0faJ7Xdzw/bMKbGc7PbNE1x3uqNtn9h+N
zdsz5wSy8lu3zzBdac72vaJ7Xdzw/bMKbGc7PbNE1x3uqNtn9h+Nzdsz5wSy8lu3zzBcsVewpyS1LmTWG7Y3nLCPm1JN05KLP/D8tRGzCIJnTJ5YbtLSfs05Z0
46TE8j8sT23EnJLUuZNYbtjecs1+bUk3TKos/8Py1EbMKUmdO4nlhu0t+zTlnTjpMTyP/R/i8Pwl//fjZYb3Jvv8Pd/il+WWG5wb77D3/8pfllicG9+Q5//6f4ZYnl
BvfmO1y9PH7KfTtbhq+zySpMyVtbr7D1cvjp2yxveWn4ftMkjpT0ubmO1y9PH7KfTtbhq+zySpMyVtbr7D1cvjp2yxveWn4ftMkjpT0ubmO1y9ftRg9y0n7FP
D+paTtk9O71sT13Mv7WD3LSfsU8P6lpO2T07vWxPXcy/tYpctJ+xTw/qWk7ZPTu9bE9dzL+1g9y0n7FPD+paTtk9O71sT1/P7EnOTWG5wb5LUmRptn3D/6b
6+eX04YW4Syw3uTZI6U6PtE+4/3dc3rw8nzE1iucG9SVJnarR9wv2n+/rm9eGEuUksN7g3SepMjBZPuP90X9+8PpwwN0mb72pYfzcn1rf8NHwffXXWhPmJ
mnzXQ3r7+bE+pafhu+jr876cMLcJG2+q2H93ZxY3/LT8H301VkfTpipm13Nay/mxPrW34avo++OuvDCXOT7OZGu7e+5YT9XylhH36DlFvFsTcJLu50e6tbz
lhf1diOWGfvsPVux8xN8lubrR761tO2N+VWE7Yp+9w9e5HzE2ymxvt3vqWE/Z3JZYT9uk7XL1+1GD3LX8avt8klhu2t5yc6F+/68OT2H3Ln4bvN4nlhu0tYjf61
+/68CR23/Kn4ftNYrlhe8vJif71uz48id23/Gn4fpNYbtjecnKif/3+++HTnub0fd4zieUtvLfrO1y9PH7K05y+z3smsbyF93Z9h6uXx095mtP3ec8klrfw3q7vcPXY+C
IPc/o+75nE8hbe2/Udzv9X+sv/OP/881/SqtvcdpBh+wAAAABJRu5ErkIggg==

```



<https://blog.csdn.net/zsw15841822890>

KEY{dca57f966e4e4e31fd5b15417da63269}

12、闪的好快

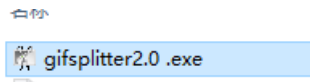
这是二维码吗？嗯。。。是二维码了，我靠，闪的好快。。。

题目来源：第七季极客大挑战



动态的gif图片

- 1. 发现下载下来是一个gif图片，并且是会动的二维码，我们可以猜测flag很有可能是藏在这些动的二维码里面，这个时候我们可以使用工具尝试分解gif里面的数据



- 1. 分离出来发现是18张不完整的gif图片，如果尝试每张去修复，工作量太大了，这个时候我们尝试去使用



1. 保存图片祭出神器StegSolve。
然后Analysis->Frame Browser。这里发现是18张图。也就是18张图片。



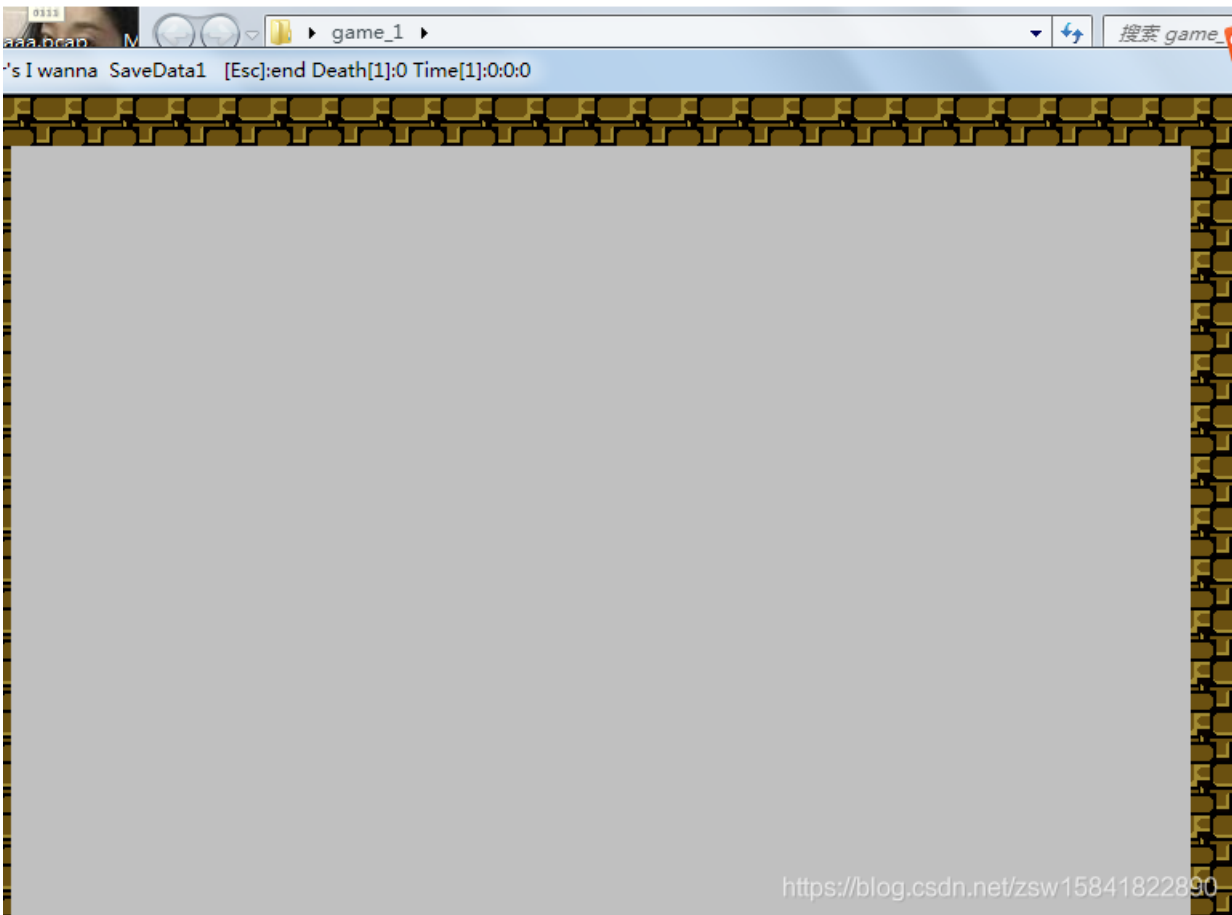
我拿手机一个挨着一个扫的。
扫出来的结果是SYC{F1aSh-so-f4sT}
但是提交不正确。
最后更改为SYC{F1aSh_so_f4sT}

13、come_game

听说游戏通关就有flag
题目来源：第七季极客大挑战

下载好题目，解压打开文件，打开可执行文件

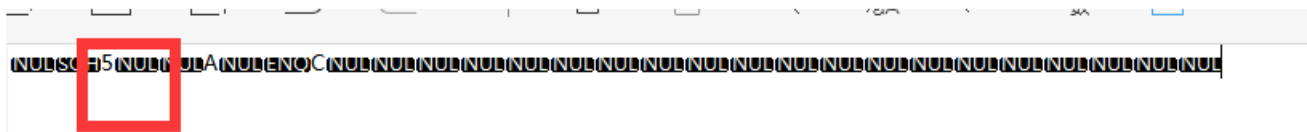
| 名称 | 修改日期 | 类型 | 大小 |
|---------------|-----------------|------|----------|
| 📁 _MACOSX | 2016/10/9 23:47 | 文件夹 | |
| 📄 DeathTime | 2020/6/28 16:35 | 文件 | 1 KB |
| 🔴 joker's.exe | 2016/10/9 23:45 | 应用程序 | 8,106 KB |



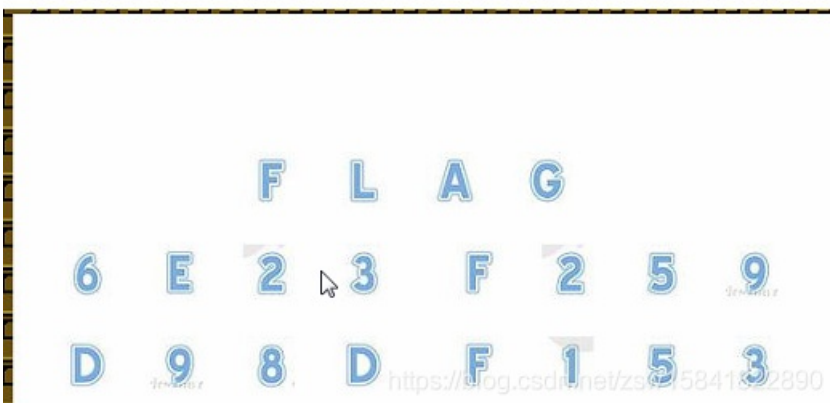
进行通关玩耍，会发现生成一个文件，用记事本打开，你会发现发现里面记录了通关数，考虑更为为最后一关

BUGKU > game_1

| 名称 | 修改日期 | 类型 | 大小 |
|------------------|-----------------|------|----------|
| 文件夹 _MACOSX | 2016/10/9 23:47 | 文件夹 | |
| 文件 DeathTime | 2020/6/28 16:35 | 文件 | 1 KB |
| 应用程序 joker's.exe | 2016/10/9 23:45 | 应用程序 | 8,106 KB |
| 文件 save1 | 2020/6/28 16:35 | 文件 | 1 KB |
| 文件 temp | 2020/6/28 16:35 | 文件 | 1 KB |



顺着玩是保存通关记录，而逆着玩，则是在读取通关记录，则会发现flag。



然后提交flag即可。

但是这里有一个坑需要注意就是这里并不是flag的格式而是SYC{6E23F259D98DF153}这种格式。

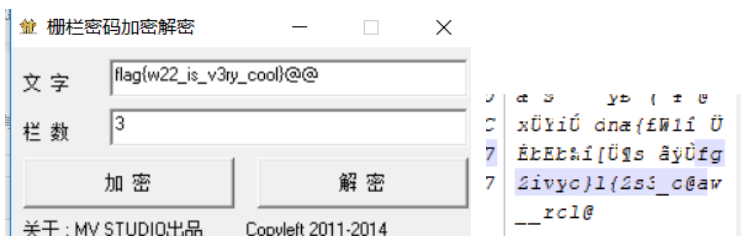
14、白哥的鸽子

图片隐写，套路化，notepad++、winhex、010edit、右键属性、binwalk、foremost打一遍

拿到文件binwalk,一切正常

```
root@kali:~/桌面# binwalk jpg\(\白哥的鸽子\)
DECIMAL      HEXADECIMAL  DESCRIPTION
-----
0            0x000000    JPEG image data, JFIF standard 1.01
30          0x00001E    TIFF image data, big-endian, offset of
directory: 8
```

丢进010editor和winhex（后者不清晰）



发现文件末位有fg2ivyo}{2s3_o@aw__rcl@字符串（含flag字样，推断出是栅栏密码，栅栏密码：就是把要加密的明文分成N个一组，形成一段无规律的话）



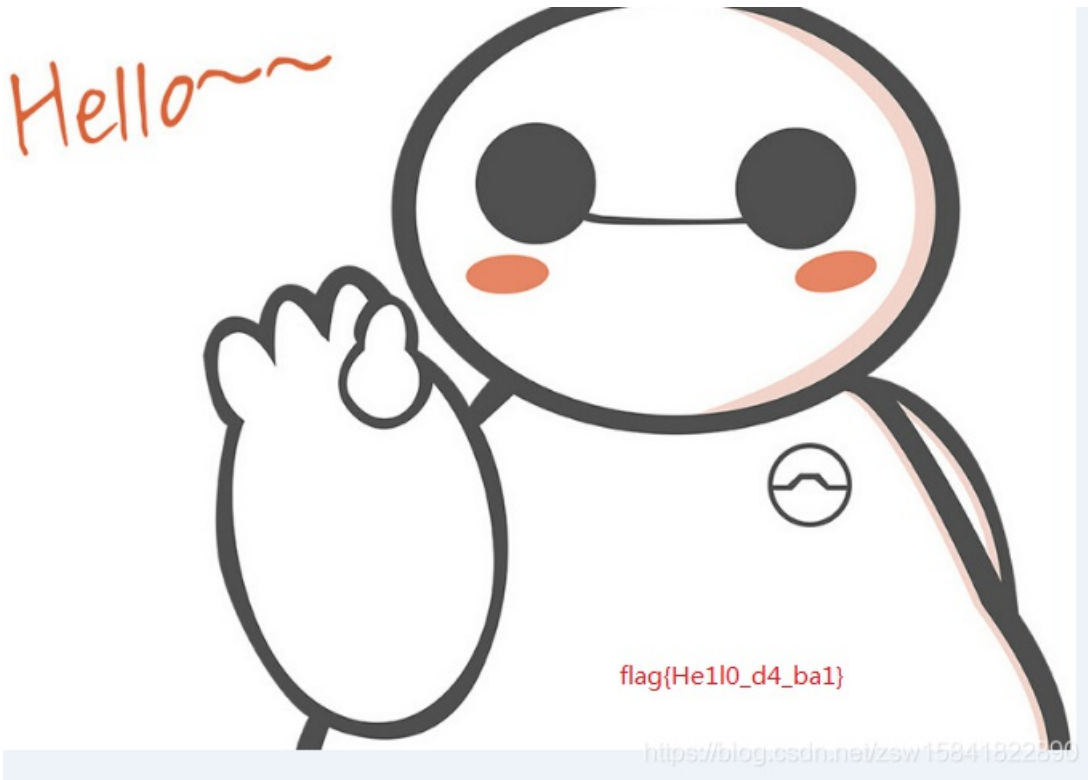
尝试栅栏密码得出flag{w22_is_v3ry_cool}@@，栏数为3

flag{w22_is_v3ry_cool}

15、linux

提示：linux基础问题

放在linux下解压（或者Windows下解压），然后得到一个flag二进制文件，使用linux命令查找关键字grep 'key' -a flag



flag{He1l0_d4_ba1}

17、做个游戏(08067CTF)

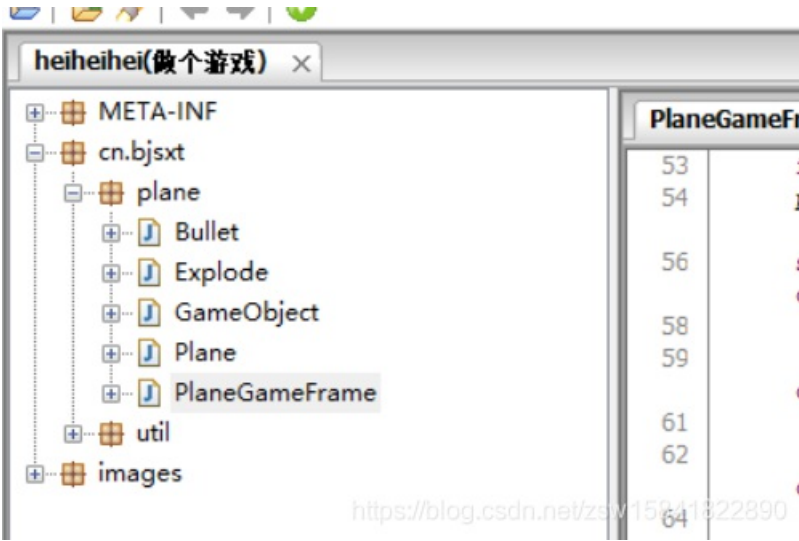
坚持60秒，Java程序heiheihei.jar



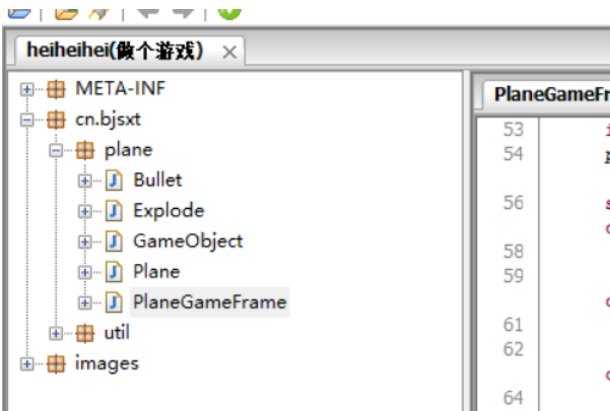
拿到题目，下载jar。直接解压，然后Java反编译.class文件

名称

- cn
- images
- META-INF



flag{RGFqaURhbGlfSmlud2FuQ2hpamk=} base64解密



18、想蹭网先解开密码

提示：flag格式：flag{你破解的WiFi密码}

tips：密码为手机号，为了不为难你，大佬特地让我悄悄地把前七位告诉你
1391040**

Goodluck!!

下载cap包，WiFi连接认证的重点在WPA的四次握手包，也就是eapol协议的包，过滤一下

| No. | Time | Source | Destination | Protoc | Len: |
|------|-----------|---------|-------------------|--------|------|
| 3066 | 45.138762 | D-Li... | LiteonTe_68:5f:7c | EAPOL | |
| 3068 | 45.154148 | Lite... | D-LinkIn_9e:4e:a3 | EAPOL | |
| 3070 | 45.168458 | D-Li... | LiteonTe_68:5f:7c | EAPOL | |
| 3072 | 45.195620 | Lite... | D-LinkIn_9e:4e:a3 | EAPOL | |

使用crunch生成密码字典

```
crunch 11 11 -t 1391040%%%% >>wifipass.txt
```

或者: `crunch 11 11 -t 1391040%%%% -o password.txt`

```
root@kali:~/桌面# crunch 11 11 -t 1391040%%%% >>wifipass.txt
Crunch will now generate the following amount of data: 120000 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 10000
```

利用aircrack 进行爆破

```
aircrack-ng w wifipass.txt wifi.cap
```

或者aircrack-ng -a2 wifi.cap -w wifipass.txt

```
root@kali:~/桌面# aircrack-ng -a2 wifi.cap -w wifipass.txt
Opening wifi.cap
Read 4257 packets.

# BSSID          ESSID          Encryption
1 3C:E5:A6:20:91:60 CATR           WPA (0 handshake)
2 3C:E5:A6:20:91:61 CATR-GUEST     WPA (0 handshake)
3 BC:F6:85:9E:4E:A3 D-Link_DIR-600A WPA (1 handshake)

Index number of target network ? 3
Opening wifi.cap
Reading packets, please wait...
https://blog.csdn.net/zsw15841822890
```

第三个存在握手包，就是他。

秒出答案

```
Quitting aircrack-ng...
root@kali:~/桌面# aircrack-ng -a2 wifi.cap -w wifipass.txt
Opening wifi.cap
Read 4257 packets.

# BSSID          ESSID          Encryption
1 3C:E5:A6:20:91:60 CATR           WPA (0 handshake)
2 3C:E5:A6:20:91:61 CATR-GUEST     WPA (0 handshake)
3 BC:F6:85:9E:4E:A3 D-Link_DIR-600A WPA (1 handshake)

Index number of target network ? 3

Opening wifi.cap
Reading packets, please wait...

Aircrack-ng 1.2 rc4

[00:00:11] 7688/9999 keys tested (664.48 k/s)
Time left: 3 seconds 76.89%

KEY FOUND! [ 13910407686 ]
```

flag{13910407686}

19、Linux2

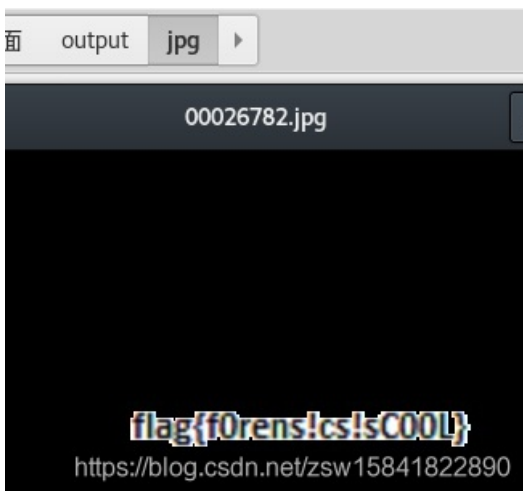
给你点提示吧：key的格式是KEY{}

压缩包brave.zip 解压，产生brave文件，固定的binwalk 和foremost执行

```
9373348 0x8F06A4 Minix filesystem,
, 0 zones brave
13712384 0xD13C00 JPEG image data, J
16832164 0x100D6A4 Minix filesystem,
, 0 zones

root@kali:~/桌面# foremost brave
Processing: brave
[*] /ice.sh
root@kali:~/桌面#
```

产生图片文件



flag{f0rens!cs!sC00L}

???出错???

flag格式是“KEY{”

尝试直接搜索brave文件！

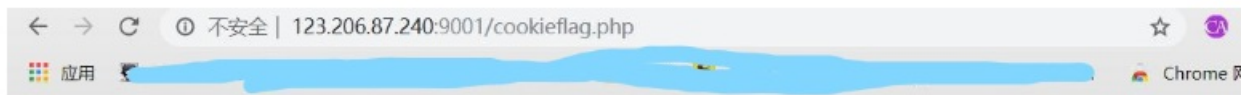
grep 'KEY' -a brave

```
root@kali:~/桌面# date
2020年 06月 29日 星期一 00:21:50 CST
root@kali:~/桌面# grep 'KEY' -a brave
0q00)' .7(000A000000'0p3000HKEY{24f3627a86fc740a7f36ee2c7a1c124a}
08L000-B00000 0?0)Y000000S00009H000mE0J0F0窥0,ENT000 e0ke0w 00Z'000
U3210#! 00Jtp00000ad0000KEY{}00 NO00F000 0002V00S0002V00
```

KEY{24f3627a86fc740a7f36ee2c7a1c124a}

20、BUGku 账号被盗了

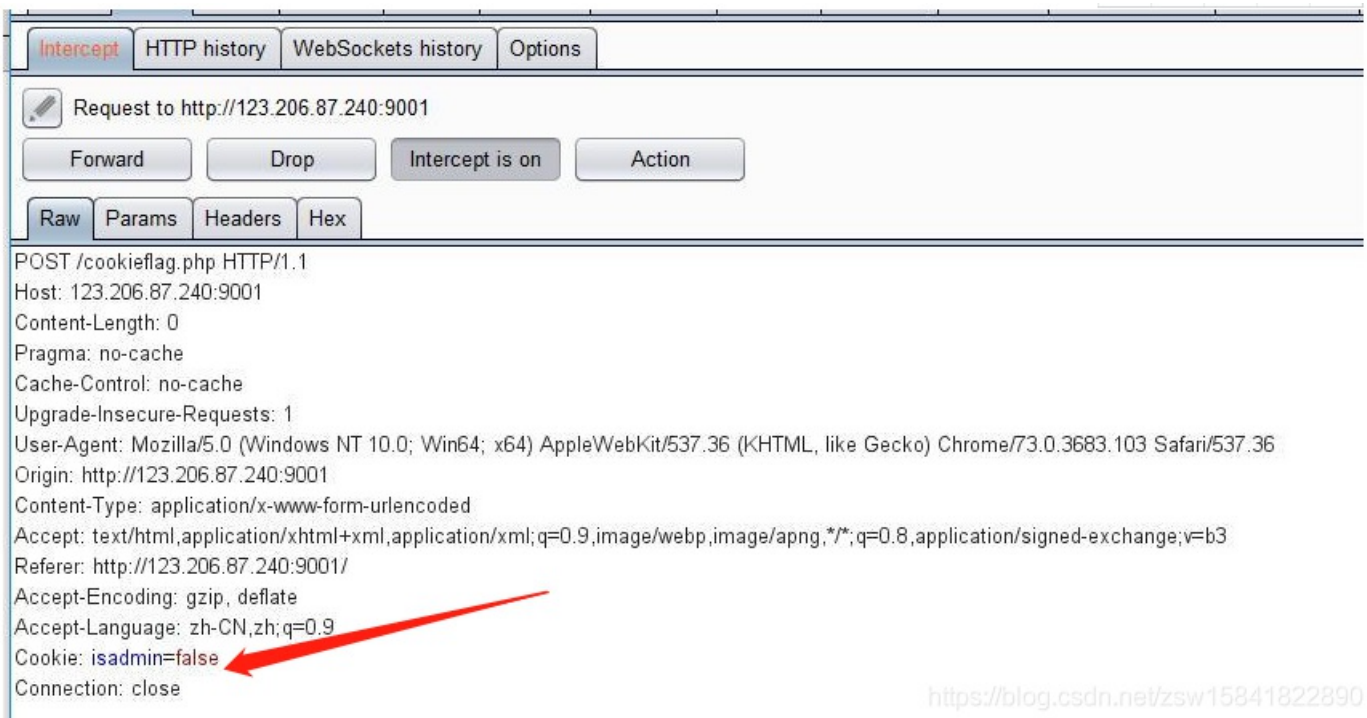
http://123.206.87.240:9001/cookieflag.php



You are not an admin!

文字说我们不是管理员

没什么提示，抓个包看看；



显示false，改成true，然后repeater下试试看

| | |
|--|--|
| 471 21.085779 14.17.57.241 192.168.43.224 SMTP | 72 S: 334 VXN1cm5hbWU6 |
| 472 21.085999 192.168.43.224 14.17.57.241 SMTP | 80 C: User: YmtjdGZ0ZXN0QDE2My5jb20= |
| 473 21.151023 14.17.57.241 192.168.43.224 SMTP | 72 S: 334 UGFzc3dvcmQ6 |
| 474 21.151170 192.168.43.224 14.17.57.241 SMTP | 68 C: Pass: YTEyMzQ1Ng== |
| 475 21.151323 14.17.57.241 192.168.43.224 TCP | 64 S: 40736 [ACK] Seq=1114269669 Len=0 |

<https://blog.csdn.net/zsw15841822890>

发现出来个网页，打开后自动下载一个叫123.exe的文件



跟游戏有关的，利用wireshark分析一下；

找到User和pass

| | | | | | |
|-----|-----------|----------------|----------------|------|---|
| 471 | 21.085779 | 14.17.57.241 | 192.168.43.224 | SMTP | 72 S: 334 VXNlcm5hbWU6 |
| 472 | 21.085999 | 192.168.43.224 | 14.17.57.241 | SMTP | 80 C: User: YmtjdGZ0ZXN0QDE2My5jb20= |
| 473 | 21.151023 | 14.17.57.241 | 192.168.43.224 | SMTP | 72 S: 334 UGFzc3dvcmQ6 |
| 474 | 21.151170 | 192.168.43.224 | 14.17.57.241 | SMTP | 68 C: Pass: YTEyMzQ1Ng== |
| 475 | 21.170563 | 14.17.57.241 | 192.168.43.224 | TCP | 54 25 -> 40726 [ACK] Seq=211 Ack=75 Win=14464 Len=0 |

选一个右键点击追踪流就行了，这两行用base64解密就是163邮箱的账号和密码。登陆之后找flag。

YmtjdGZ0ZXN0QDE2My5jb20=
334 UGFzc3dvcmQ6
YTEyMzQ1Ng==

收件箱 (1)
红旗邮件
待办邮件
智能标签
星标联系人邮件
草稿箱
已发送
已删除
广告邮件
收件箱2 (1)
KEY{sg1H78Si9C0s..
> 其他2个文件夹
> 邮件标签

Re:Re:KEY{sg1H78Si9C0s99Q}   

hackby 于 17:52 发给 hackby

真flag! flag{182100518+725593795416}

在 2019-04-11 17:50:38, "hackby" <bkcftfest@163.com> 写道:
- 隐藏引用文字 -
3楼: wdnmd这个flag是不是被改过 我提交WA啊

在 2018-11-19 11:53:39, "bkcftfest" <bkcftfest@163.com> 写道:
KEY{sg1H78Si9C0s99Q} 也不知道那个狗比把flag改了, 我还以为我找错地方了
1楼: 哈哈, 我有点想删flag, 但是我的良心制止了我
2楼: 真的flag去发件箱里找, 删改flag的一辈子单身!

<https://blog.csdn.net/zsw15841822890>

里面有人把flag修改了, 不过我发了一份真flag上去了, 去收件箱找就行了!

真flag! flag{182100518+725593795416}

flag{182100518+725593795416}

【注】未完待续