

攻防世界 misc高手进阶区

原创

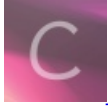
[~流氓没有兔](#) 于 2020-08-10 00:02:41 发布 1895 收藏 3

分类专栏: [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/Bok_choy/article/details/107900844

版权



[CTF 专栏收录该内容](#)

2 篇文章 0 订阅

订阅专栏

攻防世界 misc高手进阶区

每日打卡~~~

base64÷4
embarrass
神奇的Modbus
wireshark-1
pure_color
Aesop_secret
a_good_idea
Training-Stegano-1
can_has_stdio?
János-the-Ripper
Test-flag-please-ignore
Banmabanma
reverseMe
Hear-with-your-Eyes
What-is-this
MISCall
Reverse-it
something_in_image
打野
倒立屋
2017_Dating_in_Singapore
simple_transfer
Erik-Baleog-and-Olaf
hit-the-core
glance-50
Dif
4-1
适合作为桌面
心仪的公司

每日打卡~~~

小萌新正在探索的路上~

这里会记录自己做题的过程和方法，会持续更新的~

争取每日打卡哒哒哒~

base64÷4

base64÷4

👍 10 最佳Writeup由admin提供

难度系数: ★ 1.0

题目来源: 暂无

题目描述: 暂无

题目场景: 暂无

题目附件: 附件1

https://blog.csdn.net/Bok_choy

由题目可以得出关键信息—— $64 \div 4 = 16$ ，所以该加密方式应该是base16~

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
666C61677B45333342374644384133423834314341393639394544444241323442363041417D
```

将文本复制后进行base64解密得到原文:

```
666C61677B45333342374644384133423834314341393639394544444241323442363041417D
```

编码 解码 清空

```
flag{E33B7FD8A3B841CA9699EDDBA24B60AA}
```

https://blog.csdn.net/Bok_choy

```
flag{E33B7FD8A3B841CA9699EDDBA24B60AA}
```

embarrass

embarrass

👍 9 最佳Writeup由随便娶一个 · jerrita提供

难度系数: ★ 1.0


```

root@kali:~# cc
root@kali:~# strings misc_02.pcapng|grep flag
GET /flag.php HTTP/1.1
GET /flag.doc HTTP/1.1
flag{Good_b0y_W3ll_Done}
flag{Good_b0y_W3ll_Done}
flag{Good_b0y_W3ll_Done}
flag{Good_b0y_W3ll_Done}

```

神奇的Modbus

神奇的Modbus

👍 7
最佳Writeup由sins7 • giun提供

难度系数: ★ 1.0

题目来源: XCTF 4th-SCTF-2018

题目描述: 寻找flag,提交格式为sctf{xxx}

题目场景: 暂无

题目附件: 附件1

https://blog.csdn.net/Bok_choy

由题目描述得到信息——提交格式为sctf{}，大胆搜寻sctf{}:

The screenshot shows a Wireshark capture of network traffic. A search filter 'sctf' is applied to the packet list. The selected packet is a Modbus response (packet 66) containing a Read Holding Registers response. The register value is 102. Below the packet list, the hex and ASCII data is shown. The ASCII column contains the string 'sctf', which is highlighted with a red box.

然后就得到flag啦，但是这道题很坑，因为你找到的flag中是mdbus没有o，而真正的flag是有o的，可能这就是“神奇的modbus吧”hhh

sctf{Easy_Modbus}

法二：转载大佬的wp（侵权）



序号	解题思路	点赞数	上传者	操作
1	xctf-wp	6	admin	

很详细的解题思路：

SCTF 2018 : 神奇的Modbus

[目标]

了解modbus协议

[环境]

无

[工具]

Wireshark

[分析过程]

在数据包中寻找flag就行，flag是明文形式存储。

工业设备消息传输使用modbus协议。所以我就采集了modbus的通信数据包。在这些数据传输中存在着flag。接替思路是这样的，利用该modbus读取设备的相关数据。首先读取数据的话modbus有01，02，03，04功能码，01是读取线圈状态：取得一组逻辑线圈的当前状态（ON/OFF），02是读取输入状态：取得一组开关输入的当前状态（ON/OFF），03是读取保持寄存器：在一个或多个保持寄存器中取得当前的二进制值，04读取输入寄存器：在一个或多个输入寄存器中取得当前的二进制

wireshark输入modbus过滤追踪tcp流

Wireshark · 追踪 TCP 流 (tcp.stream eq 4) · modbus

```

.....E.....
7..4.....".....
.....}.....
.....+.(.....j.....
.....G.....#.....
.....K.....=.....:.....
.....6...../.....
.....].....
%..".....T.....#..c.t.f.
{.E.a.s.y._M.d.b.u.s.}.....
3....._.....
1.....].....N.!
.....y.....{.....
7..4.....E.....
.....E.a.s.y._M.d.b.u.s.}.....G.....
5.....~.....7..4.....
...../.....N.'.....
$.....-.....
.....|.....
.....7.....

```

分组 2103, 2, 032 客户端 分组, 2, 032 服务器 分组, 4, 061 run(s). 点击选择。

Entire conversation (58 kB) 显示和保存数据为 ASCII 流 4

查找: 查找下一个(N)

滤掉此流 打印 Save as... 返回 Close Help

sctf{Easy_Mdbus} (记得加一个o哦)

wireshark-1

wireshark-1

👍 7 最佳Writeup由admin提供

难度系数: ★ 1.0

题目来源: 广西首届网络安全选拔赛

题目描述: 黑客通过wireshark抓到管理员登陆网站的一段流量包 (管理员的密码即是答案)。 flag提交形式为flag{XXXX}

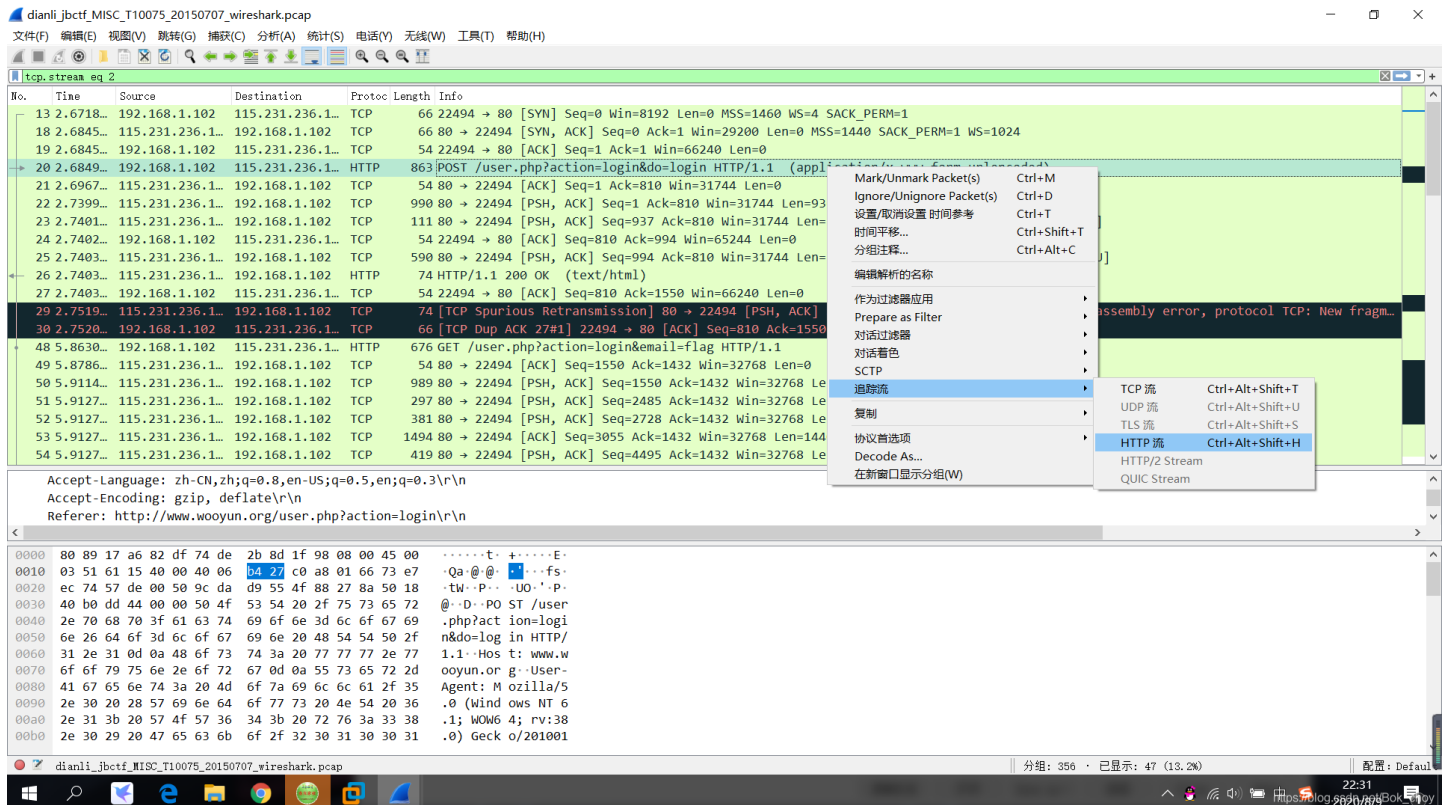
题目场景：暂无

题目附件：附件1

https://blog.csdn.net/Bok_choy

“黑客通过wireshark抓到管理员登陆网站的一段流量包（管理员的密码即是答案）。”

由于是管理员登录网站然后被抓捕了信息，所以可以通过网站入手，追踪http流



描述又说了flag就是管理员密码，于是得到的这串字符就是flag了

```
POST /user.php?action=login&do=login HTTP/1.1
Host: www.wooyun.org
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:38.0) Gecko/20100101 Firefox/38.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://www.wooyun.org/user.php?action=login
Cookie: __cfduid=d473db479254a41d53bd0aae31cb7dc3b1433775400;
Hm_lvt_c12f88b5c1cd041a732dea597a5ec94c=1434891316,1435283549,1435557576,1435590542; bdshare_firsttime=1433775454650;
wy_uid=-1; PHPSESSID=h8i10mi6rdc8l9coc708otq661; Hm_lpvt_c12f88b5c1cd041a732dea597a5ec94c=1435590574
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 65

email=flag&password=fffb7567a1d4f4abdffdb54e022f8facd&captcha=BYUGHTTP/1.1 200 OK
Date: Mon, 29 Jun 2015 15:09:10 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Server: yunjiasu-nginx
CF-RAY: 1fe28d0a63e91c3b-JXG
Content-Encoding: gzip

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
```



```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="refresh" content="3; url=/user.php?action=login&email=flag"/>
<title> WooYun.org | ..... </title>
<meta name="author" content="80sec"/>
<meta name="copyright" content="http://www.wooyun.org/" />
<meta name="keywords"
content=",wooyun,.....,web.....,....." />
<meta name="description"
content="WooYun....." />
<link href="/css/style.css" rel="stylesheet" type="text/css"/>
```

pure_color

pure_color 👍 1 最佳Writeup由老乐与涛 · Sla提供

难度系数: ★ 1.0

题目来源: school-ctf-winter-2015

题目描述: 格式为flag{xxxxxx}

题目场景: 暂无

题目附件: 附件1

https://blog.csdn.net/Bok_choy

下载附件后得到了一张png图片，打开后什么都没有，猜测是图片隐写

https://blog.csdn.net/Bok_choy

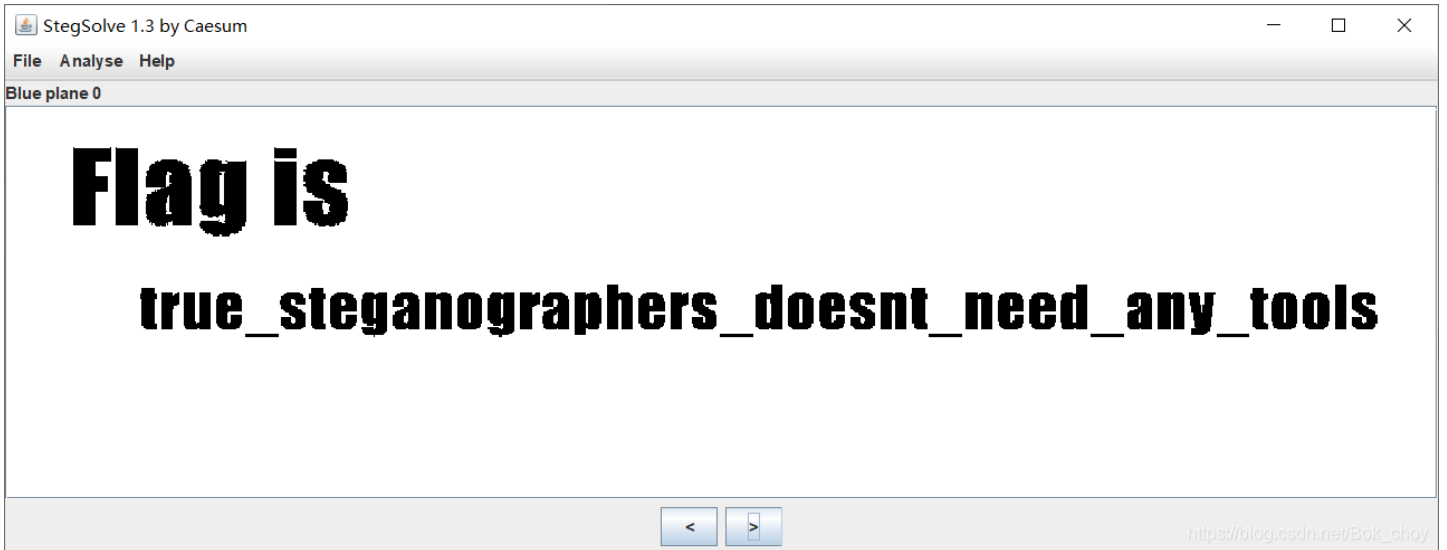
遇到图片隐写的题一般就会用到工具stegsolve
放进去跑一下



Flag is



https://blog.csdn.net/Bok_choy



得到flag

[Aesop_secret](#)

Aesop_secret 1 最佳Writeup由admin提供

难度系数: 1.0

题目来源: 2019_ISCC

题目描述: 暂无

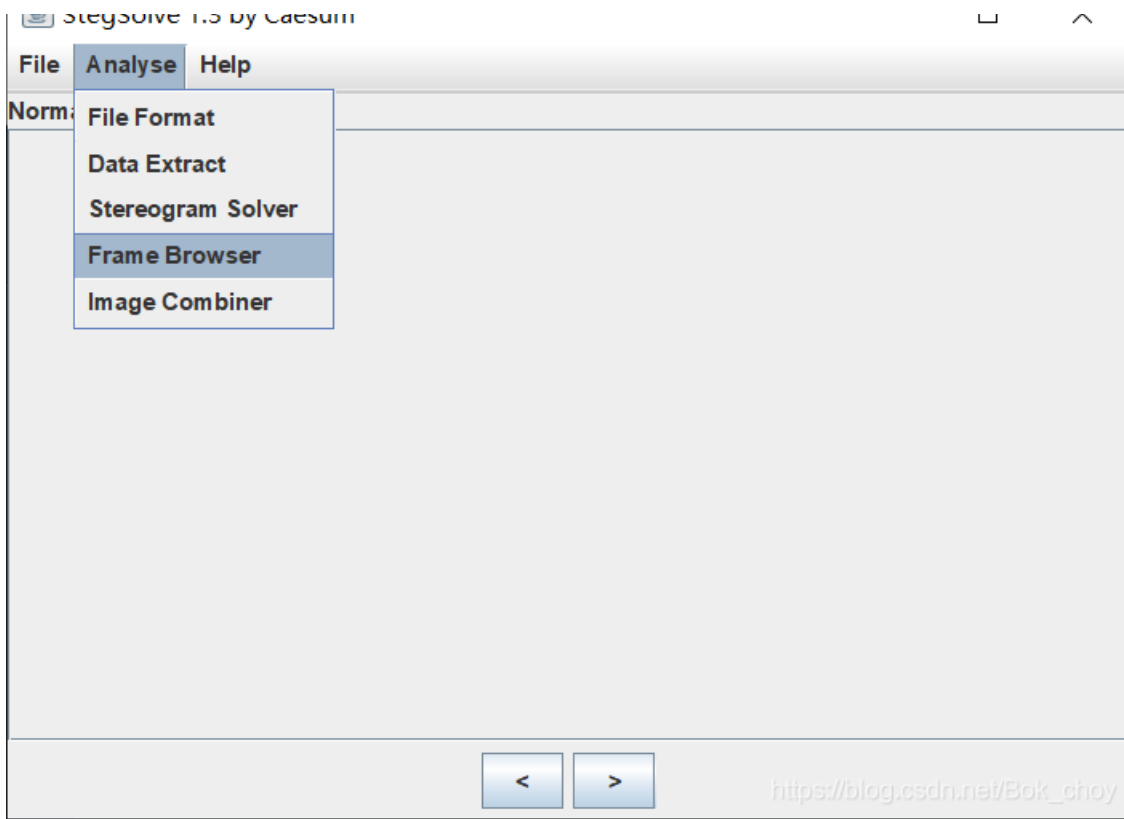
题目场景: 暂无

题目附件: [附件1](#)

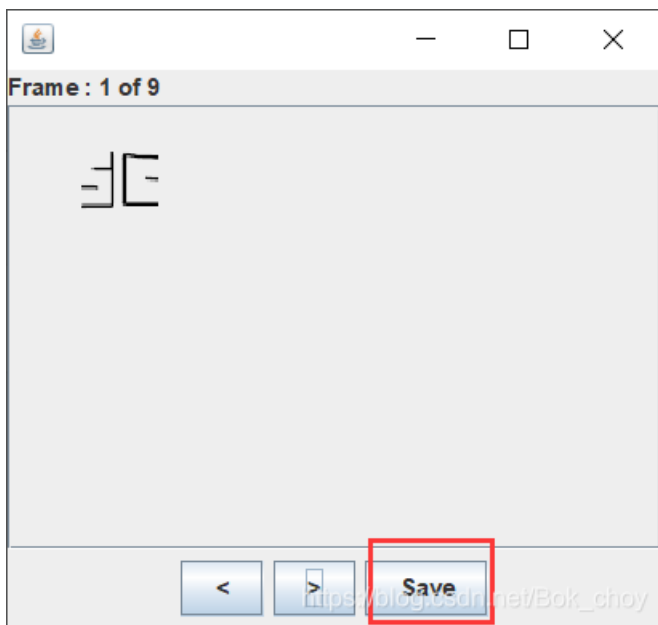
https://blog.csdn.net/Bok_choy

同样，下载下来后解压是一个gif的动图，同样需要放进stegsolve里





然后把图片的每一帧都保存下来



- 📄 frame1.bmp
- 📄 frame2.bmp
- 📄 frame3.bmp
- 📄 frame4.bmp
- 📄 frame5.bmp
- 📄 frame6.bmp
- 📄 frame7.bmp
- 📄 frame8.bmp
- 📄 frame9.bmp

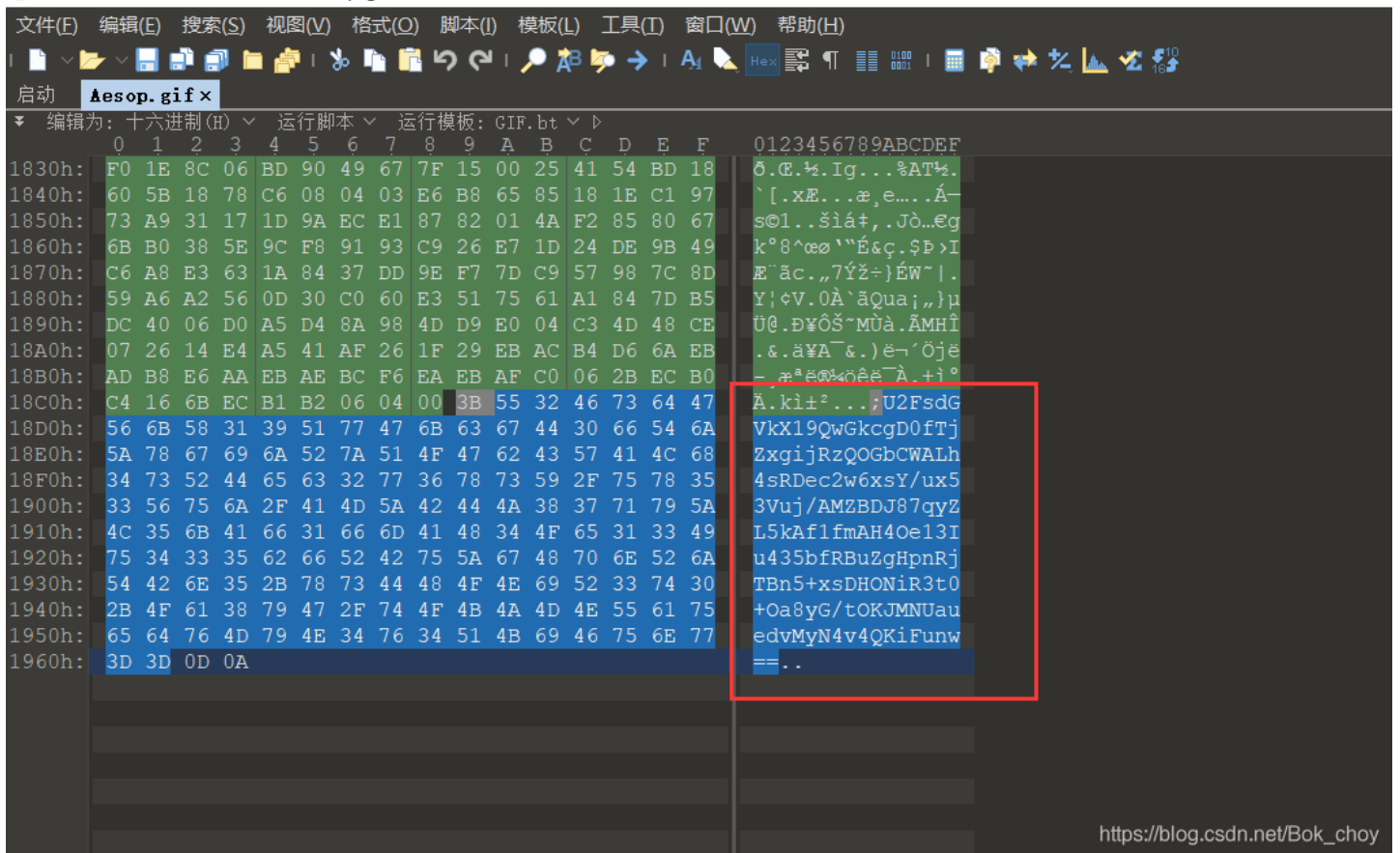
然后得到了9张图片

然后拼图得到 ISCC (这里是看到其他大佬拼图拼出来的, 菜鸡我拼不出来.....o(∩_∩)o)
 图片隐写我们还会查看是否夹带着文件或者其他什么信息, 于是扔进虚拟机里跑一下

```
root@kali: # binwalk -e Aesop.gif

DECIMAL      HEXADECIMAL  DESCRIPTION
-----
0            0x0          GIF image data, version "89a", 124 x 70
1004        0x3EC        Unix path: /www.w3.org/1999/02/22-rdf-syntax-ns#>
<rdf:Description rdf:about="" xmlns:xmpMM="http://ns.adobe.com/xap/1.0/mm/" xml
ns:stRef=
```

没有发现什么信息, 打开010editor, 捕获到了一串密文



看见尾端有两个==, 以为是base64加密, 但是试了下并没有得到结果, 后来得到大佬的提示, 看到题目, 可以试一试aes加密, 正好之前拼图得到了一串字符ISCC
 将ISCC作为密匙解密得到新的字符串
 于是再次尝试解密, 得到flag



flag{DugUpADiamondADeepDarkMine}

ISCC

密码是可选项，也就是可以不填。

< 解密 加密 >

解密成功

U2FsdGVkX18OvTUIZubDnmvk2ISAKb8Jt4Zv6UWpE7Xb43f8uzeFRUKGMo6QaaNF
HZriDDV0EQ/qt38Tw73tbQ==

0.2KB
0.2KB

在线项目管理 - 使用Wrike轻松搞定

广告
https://blog.csdn.net/Bok_choy

flag{DugUpADiamondADeepDarkMine}

解码链接https://www.sojson.com/encrypt_aes.html

[a_good_idea](#)

a_good_idea

最佳Writeup由admin提供

难度系数: ★ 1.0

题目来源: 2019_NJUPT_CTF

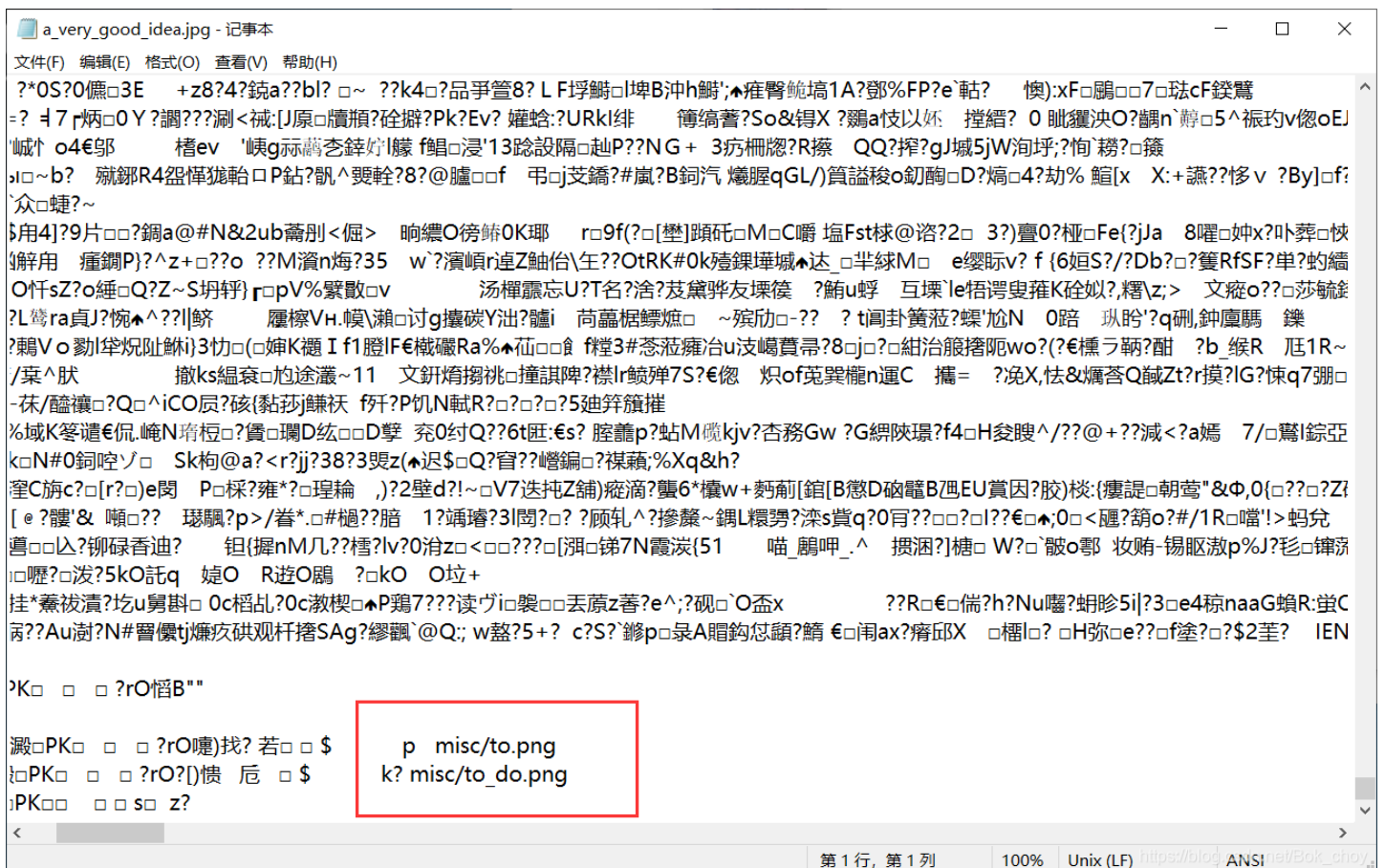
题目描述: 汤姆有个好主意

题目场景: 暂无

题目附件: 附件1

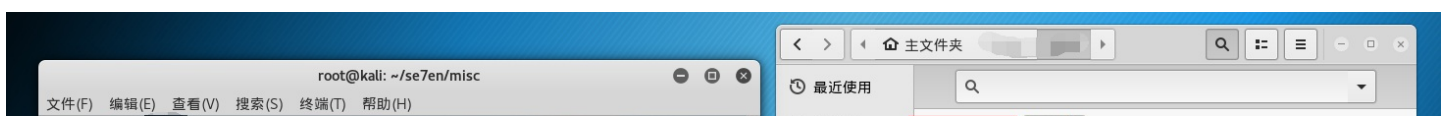
https://blog.csdn.net/Bok_choy

下载后解压得到misc文件夹，打开发现是一张Tom的表情包，用记事本打开发现最后藏了两张png图片



运行命令提取文件

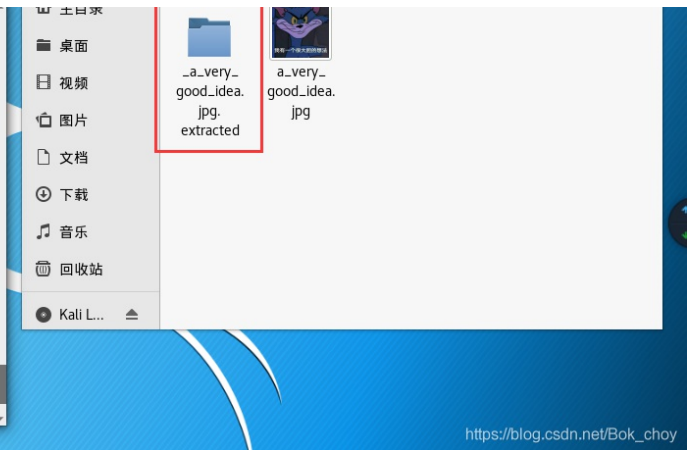
```
binwalk -e a_very_good_idea.jpg
```



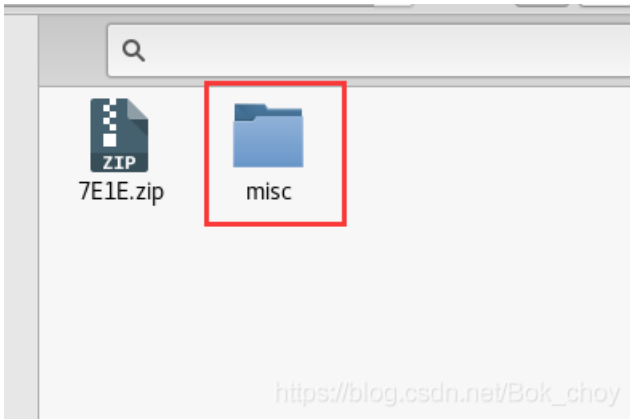
```
root@kali:~# cd misc
root@kali:~/misc# binwalk -e a_very_good_idea.jpg

DECIMAL      HEXADECIMAL     DESCRIPTION
-----
0            0x0             JPEG image data, JFIF standard 1.01
30          0x1E            TIFF image data, big-endian, offset of first image
directory: 8
32286       0x7E1E         Zip archive data, at least v1.0 to extract, name:
misc/
32321       0x7E41         Zip archive data, at least v2.0 to extract, compr
essed size: 34, uncompressed size: 32, name: misc/hint.txt
32398       0x7E8E         Zip archive data, at least v2.0 to extract, compr
essed size: 128210, uncompressed size: 128200, name: misc/to.png
160649      0x27389        Zip archive data, at least v2.0 to extract, compr
essed size: 177379, uncompressed size: 177368, name: misc/to_do.png
338443      0x52A0B        End of Zip archive

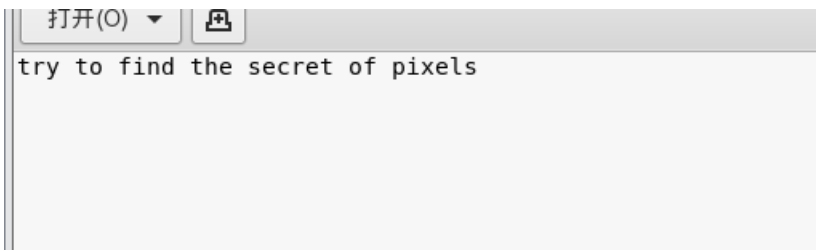
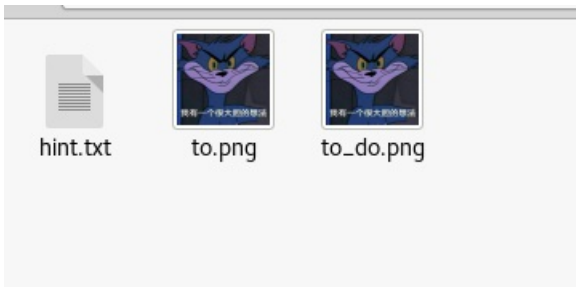
root@kali:~/misc#
root@kali:~/misc#
root@kali:~/misc#
root@kali:~/misc#
root@kali:~/misc#
```



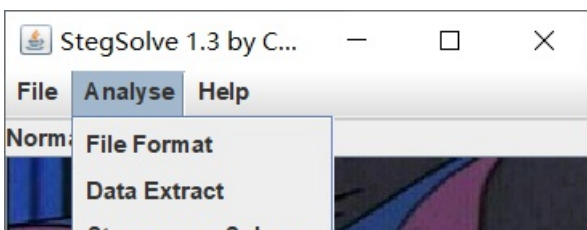
https://blog.csdn.net/Bok_choy

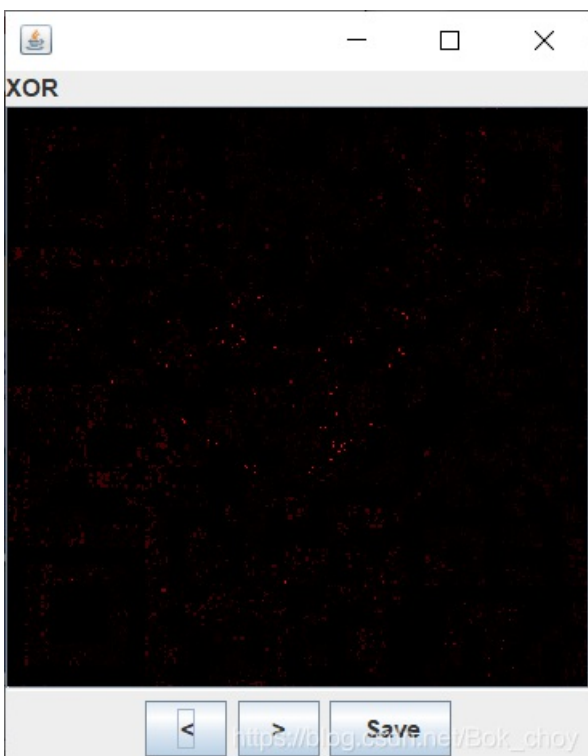
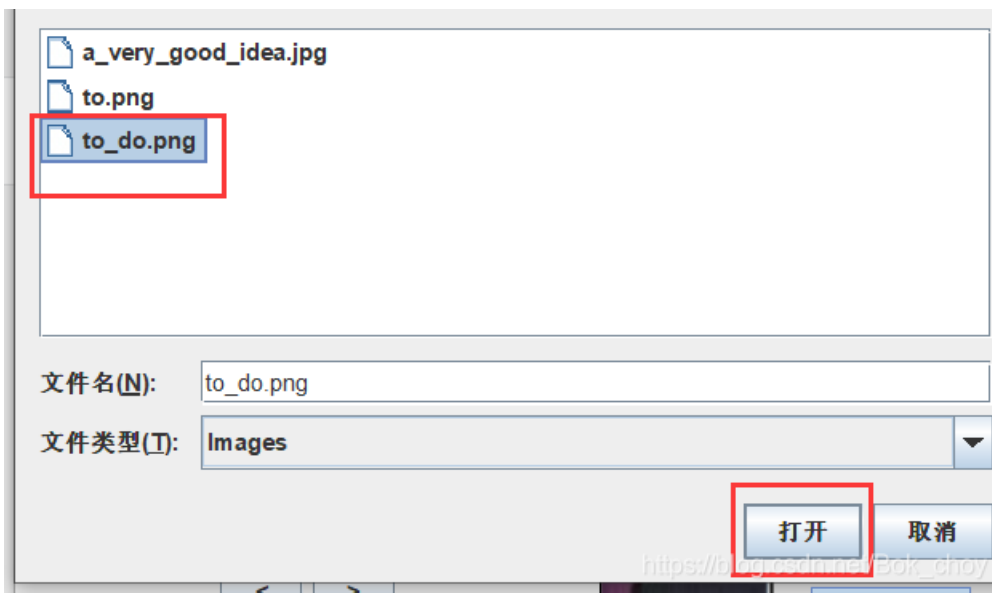


https://blog.csdn.net/Bok_choy



这个信息让我误以为是关于图片高度的隐写，实际上并没有。当我们看见两张相似的图片是，我们可以作比较，看看有什么不同。同样，扔进stegsolve





然后我们得到了这样一张图

当我们仔细观察这张图时会发现他可能是一张二维码

通过PS调节亮度就可以得到一张二维码

此外还可以用到Beyond Compare 软件，直接得到二维码



扫码得到flag

NCTF{m1sc_1s_very_funny!!!}

Training-Stegano-1


```
flag{esolangs_for_fun_and_profit}
```

Text to Ook!

Text to short Ook!

Ook! to Text

Text to Brainfuck

Brainfuck to Text

https://blog.csdn.net/Bok_choy

```
flag{esolangs_for_fun_and_profit}
```

János-the-Ripper

János-the-Ripper

最佳Writeup由sins7 • giun提供

难度系数: ★ 1.0

题目来源: tinyctf-2014

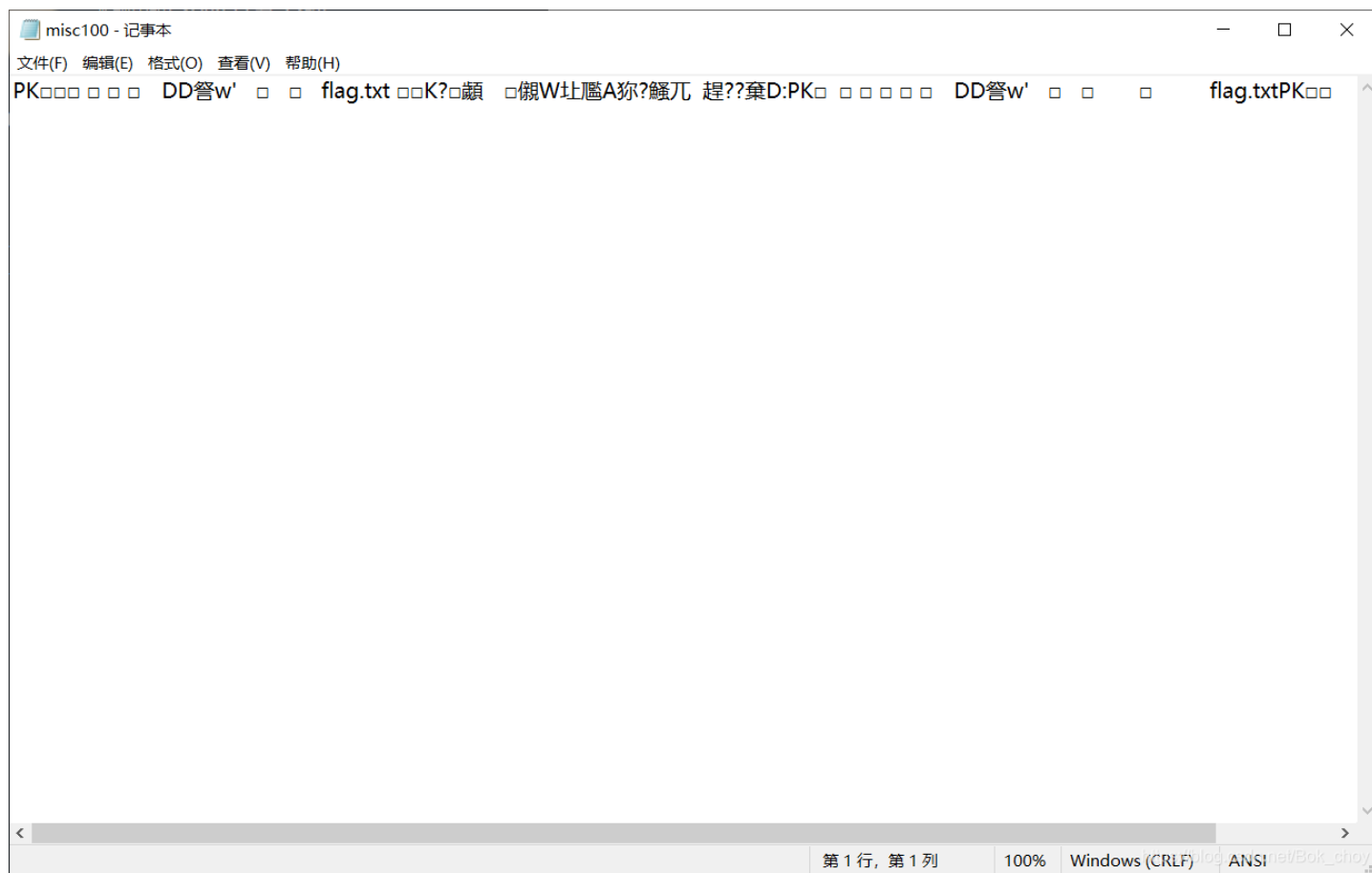
题目描述: 暂无

题目场景: 暂无

题目附件: 附件1

https://blog.csdn.net/Bok_choy

记事本打开发现有flag.txt藏在最后面



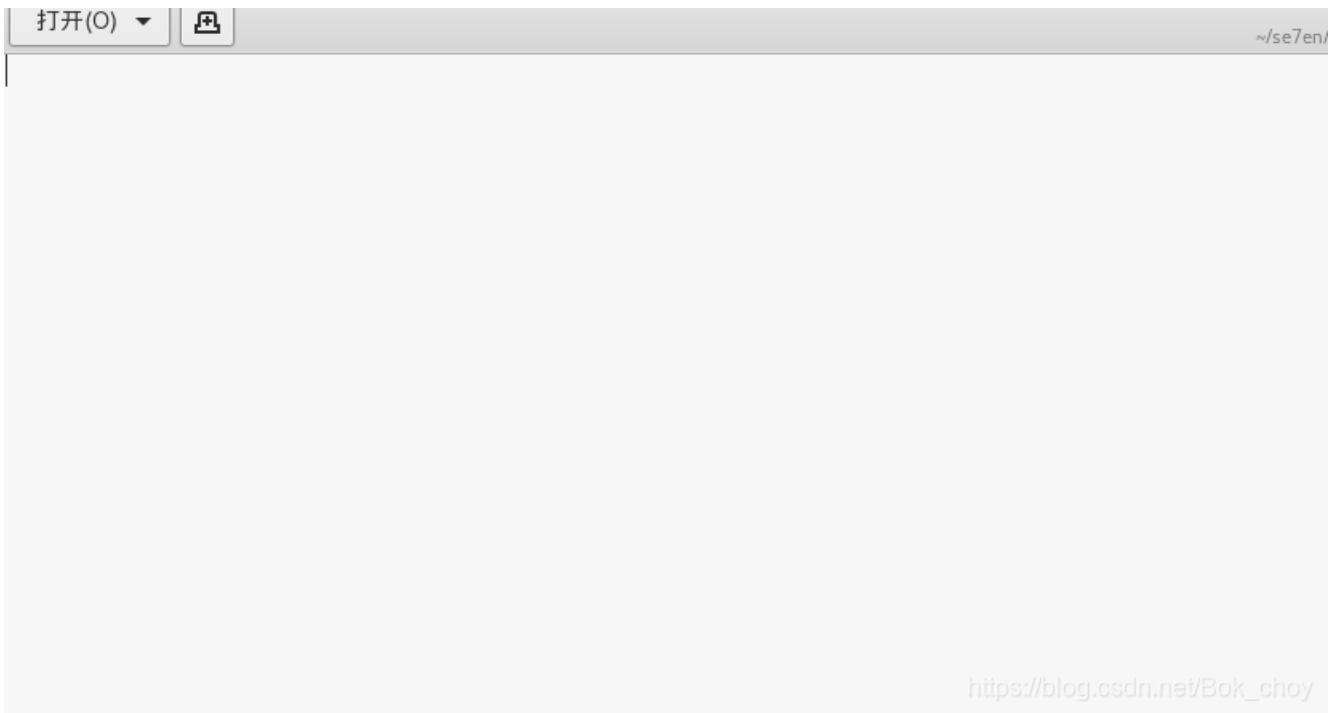
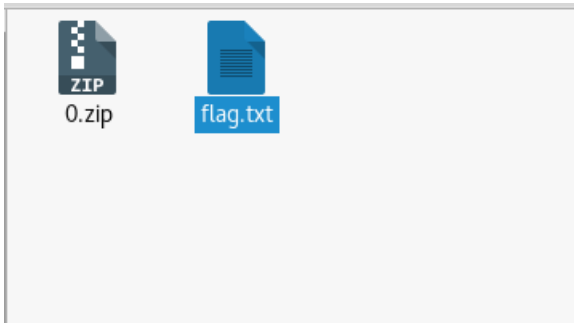
将文件提取出来

```
binwalk -e misc100
```





打开后有个文件夹，里面的flag.txt文件打开是空的，真正的flag在压缩包里，但是压缩包是经过加密的



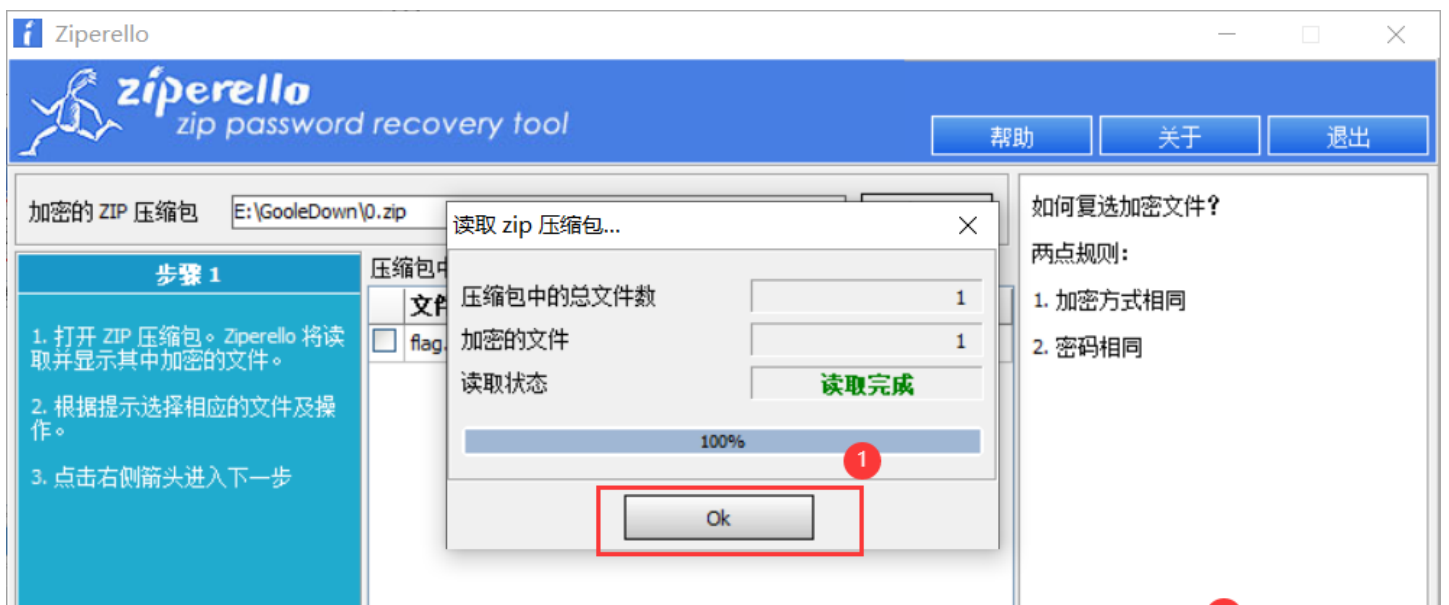
https://blog.csdn.net/Bok_choy

此时需要用到压缩包解密工具，题目是一种解密工具，但是我虚拟机没有安装，所以直接拷贝出来用其他工具，我用的是ziperello，然后按照他的步骤依次来就是了

这里附加其他博主的关于John the Ripper使用的博文传送门~



记得将后面的flag文件点上





这里有三种破解方式，可以根据自己的条件选择，因为没有什么关于密码的信息，所以我们直接暴力破解吧

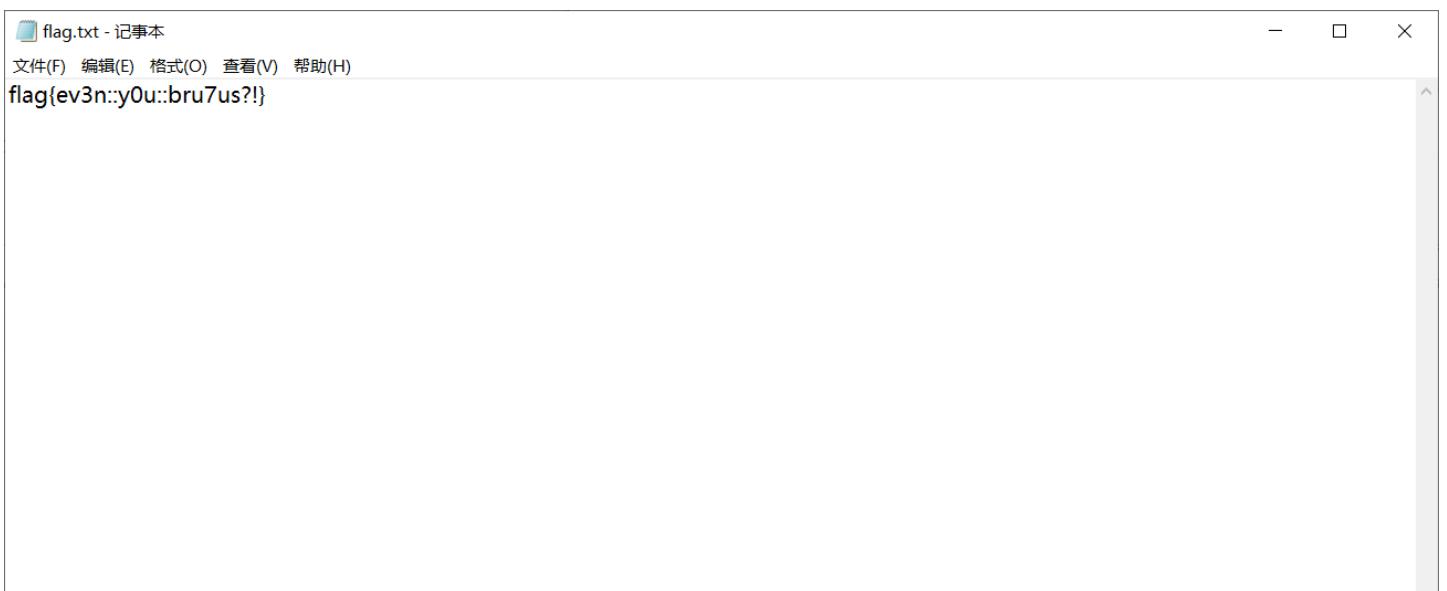


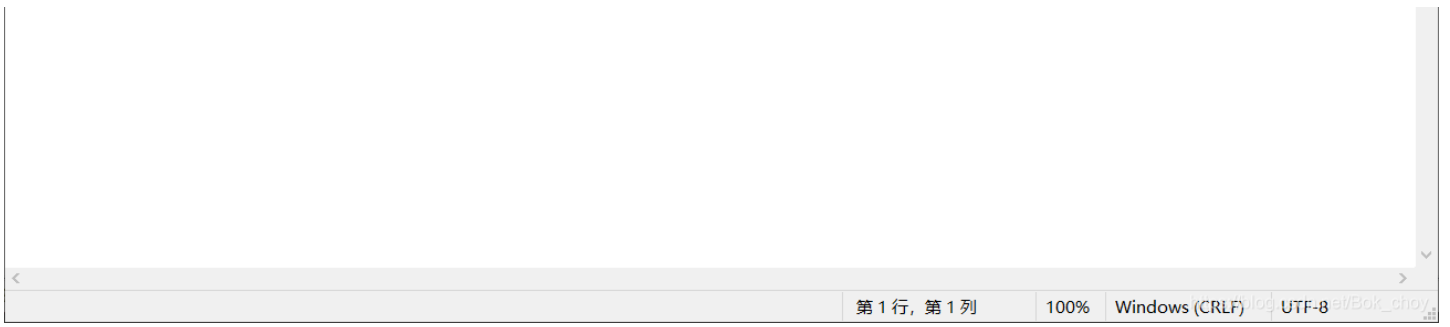
猜测密码中存在的字符集，然后勾选，设置好密码长度即可





密码出来了! 密码是fish, 用密码解压后得到flag





flag{ev3n::y0u::bru7us?!}

Test-flag-please-ignore

Test-flag-please-ignore

最佳Writeup由 **B301** • dals 提供

难度系数: ★ 1.0

题目来源: [tinyctf-2014](#)

题目描述: 暂无

题目场景: 暂无

题目附件: [附件1](#)

https://blog.csdn.net/Bok_choy

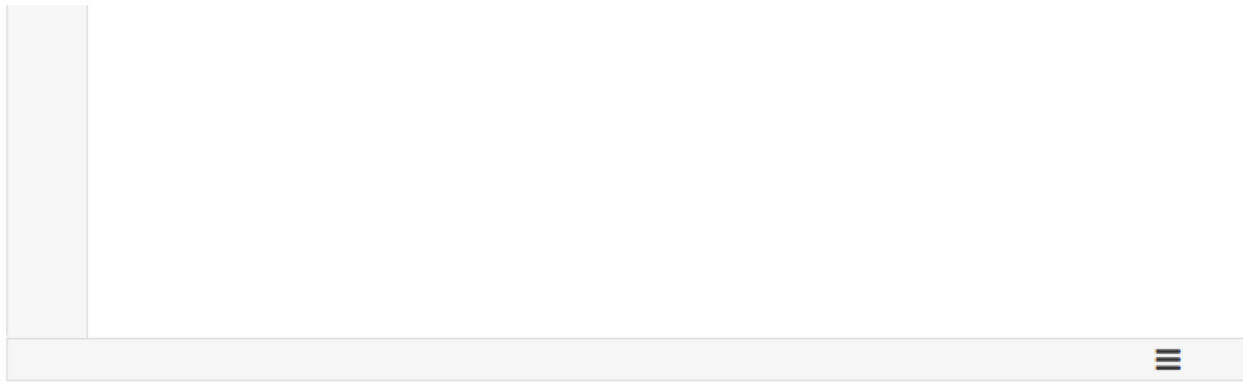
下载解压记事本打开



一来就base64解密，果然不可能这么简单o(∩_∩)o，仔细观察发现字符串没有出现f之后的字母了，所以应该是16进制加密16进制解码传送门

加密或解密字符串长度不可以超过10M

1	666c61677b68656c6c6f5f776f726c647d
---	------------------------------------



- 16进制转字符
- 字符转16进制
- 测试用例
- 清空结果
- 复制结果

```
1 flag{hello_world}
```

https://blog.csdn.net/Bok_choy

flag{hello_world}

Banmabanma

Banmabanma 2 最佳Writeup由admin提供

难度系数: 1.0

题目来源: [世安杯](#)

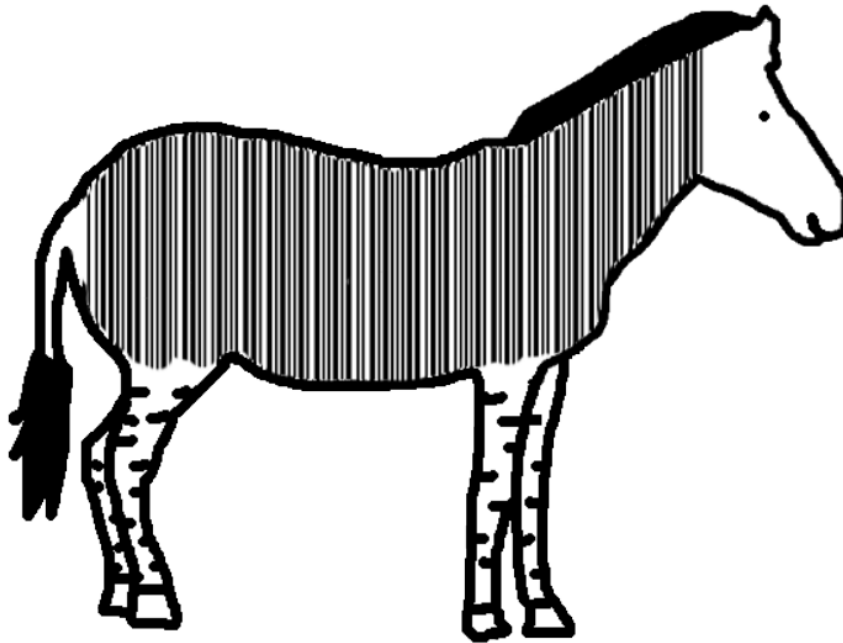
题目描述: flag格式为flag{xxxx}

题目场景: 暂无

题目附件: [附件1](#)

https://blog.csdn.net/Bok_choy

下载解压打开是一张斑马的图片，斑马身上是条形码，条形码经过处理不可以扫出来
尝试过抠图，太难了，对小白十分不友好不友好不友好
大佬推荐了一个在线扫码工具，巨好用！！3Q~
传送门



https://blog.csdn.net/Bok_choy



408.737.7092

sales@inliteresearch.com

Free Online Barcode Reader

1. Select barcode types

1D: Code 39, Code 128... PDF417 Postal: IMB, 4state ...

QR code DataMatrix Driver License, ID cards

2. Select Image File (PDF, TIFF, JPEG, BMP, GIF, PNG, WMF, WEBP)

斑马斑马.png

Maximum file size: 12 Mb.

3.

To see demonstration with our sample image:

Barcode Reader Software Development Kit (SDK). Decode barcodes in C#, VB, Java, C\C++, Delphi, PHP and other languages.

[Get ClearImage SDK](#)

Barcode Director. Barcode scanner application renames, sorts and splits documents using barcode values.

[Get Barcode Director](#)

Barcode Reader Web Server with RESTful API. Client SDKs for JavaScript, .NET (C# or VB), Java, Node.js, PHP, Python or Ruby.

[Web API Test Server](#)

This site offers free limited demonstration. See [terms of service](#).

ClearImage ver. 9.2.6273

https://blog.csdn.net/Bok_choy



408.737.7092

sales@inliteresearch.com

Free Online Barcode Reader

To get such results using [ClearImage SDK](#) use [TBR Code 103](#).

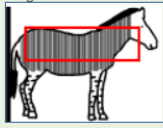
If your **business** application needs barcode recognition capabilities, email your technical questions to support@inlitesearch.com email your sales inquiries to sales@inlitesearch.com

File: 斑马斑马.png New File

Pages: 1 Barcodes: 1

Barcode: 1 of 1 Type: Code39
Length: 16 Rotation: none
Module: 1.6pix Rectangle: {X=71,Y=93,Width=410,Height=119}

Page 1 of 1



FLAG IS TENSHINE

Barcode Reader Software Development Kit (SDK)

Decode barcodes in C#, VB, Java, C/C++, Delphi, PHP and other languages.

[Get ClearImage SDK](#)

Barcode Director. Barcode scanner application renames, sorts and splits documents using barcode values.

[Get Barcode Director](#)

Barcode Reader Web Server with RESTful API. Client SDKs for JavaScript, .NET (C# or VB), Java, Node.js, PHP, Python or Ruby.

[Web API Test Server](#)

This site offers free limited [demo](#) [in.net/Bok_choy](#) demonstration. See [terms of](#)

FLAG IS TENSHINE

直接读取太简单了吧爽歪歪~~~

reverseMe

reverseMe 最佳Writeup由admin提供

难度系数: ★★ 2.0

题目来源: XCTF 3rd-GCTF-2017

题目描述: 暂无

题目场景: 暂无

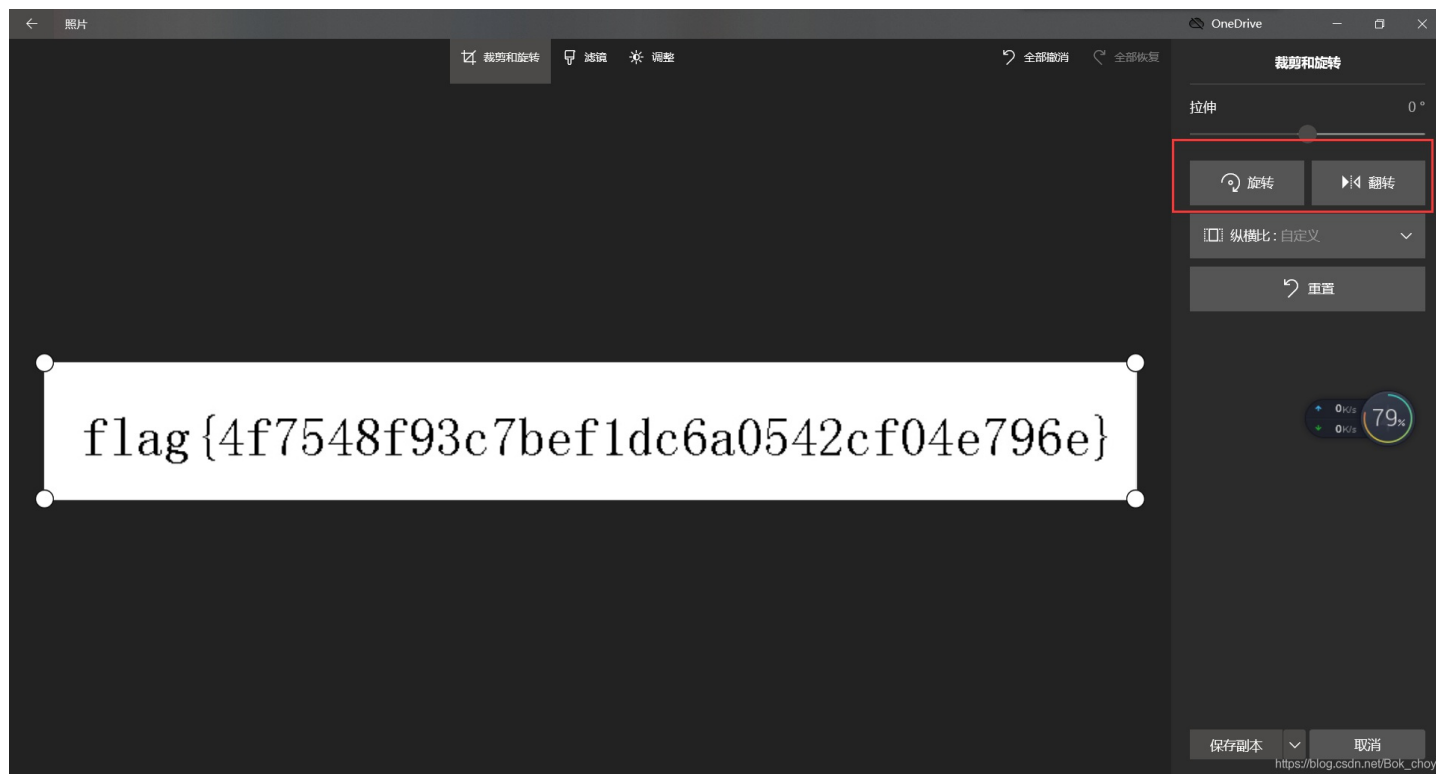
题目附件: 附件1

https://blog.csdn.net/Bok_choy

下载打开是一张翻转倒立的flag图片

{e097e407c5470a0b17ed7c0e7847714}gslf

直接编辑就可以啦



Hear-with-your-Eyes

Hear-with-your-Eyes 👍 3 最佳Writeup由admin提供

难度系数: ★★ 2.0

题目来源: su-ctf-quals-2014

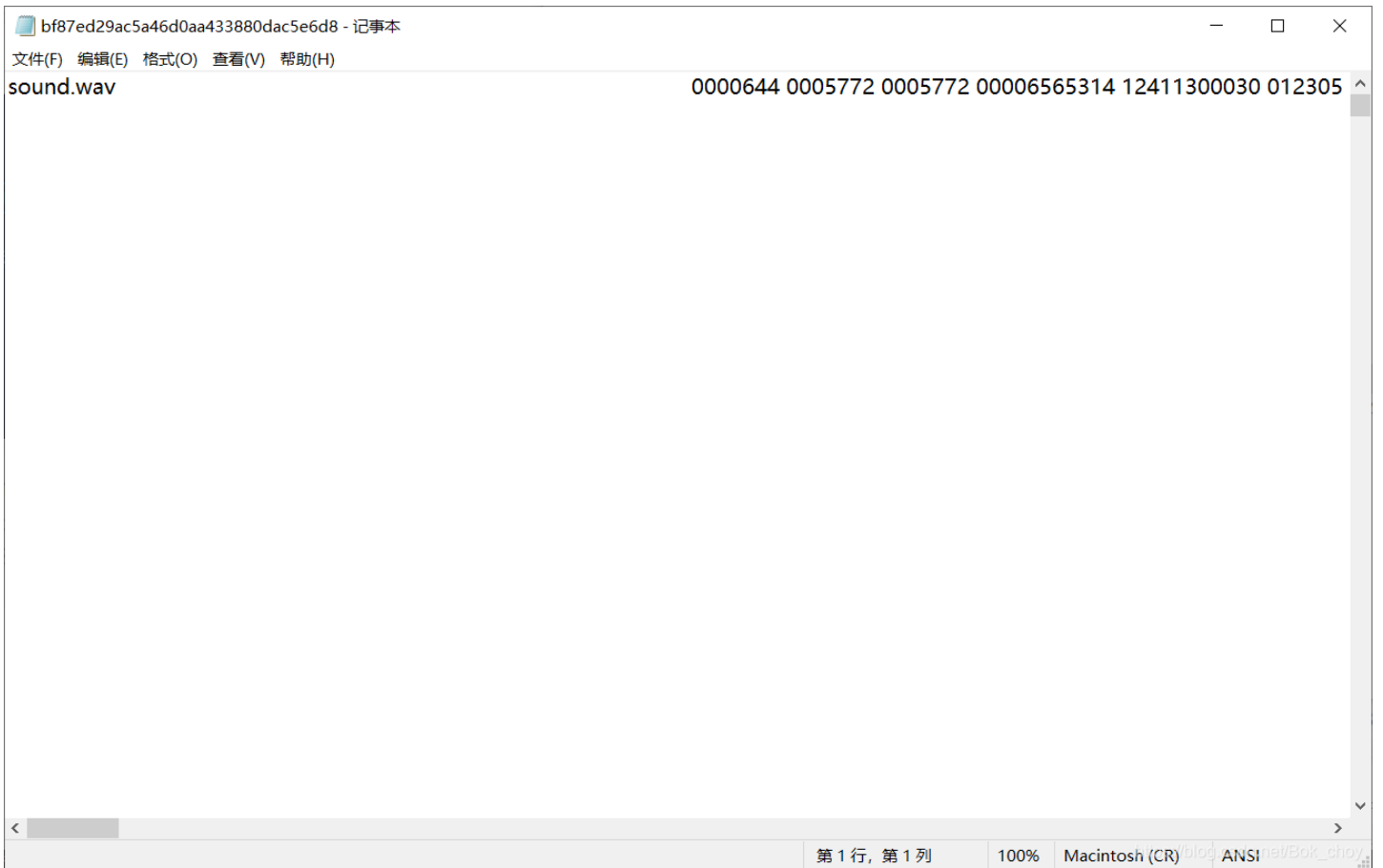
题目描述: 用眼睛听这段音频

题目场景: 暂无

题目附件: 附件1

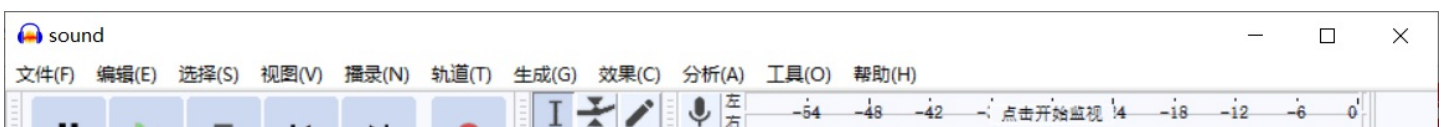
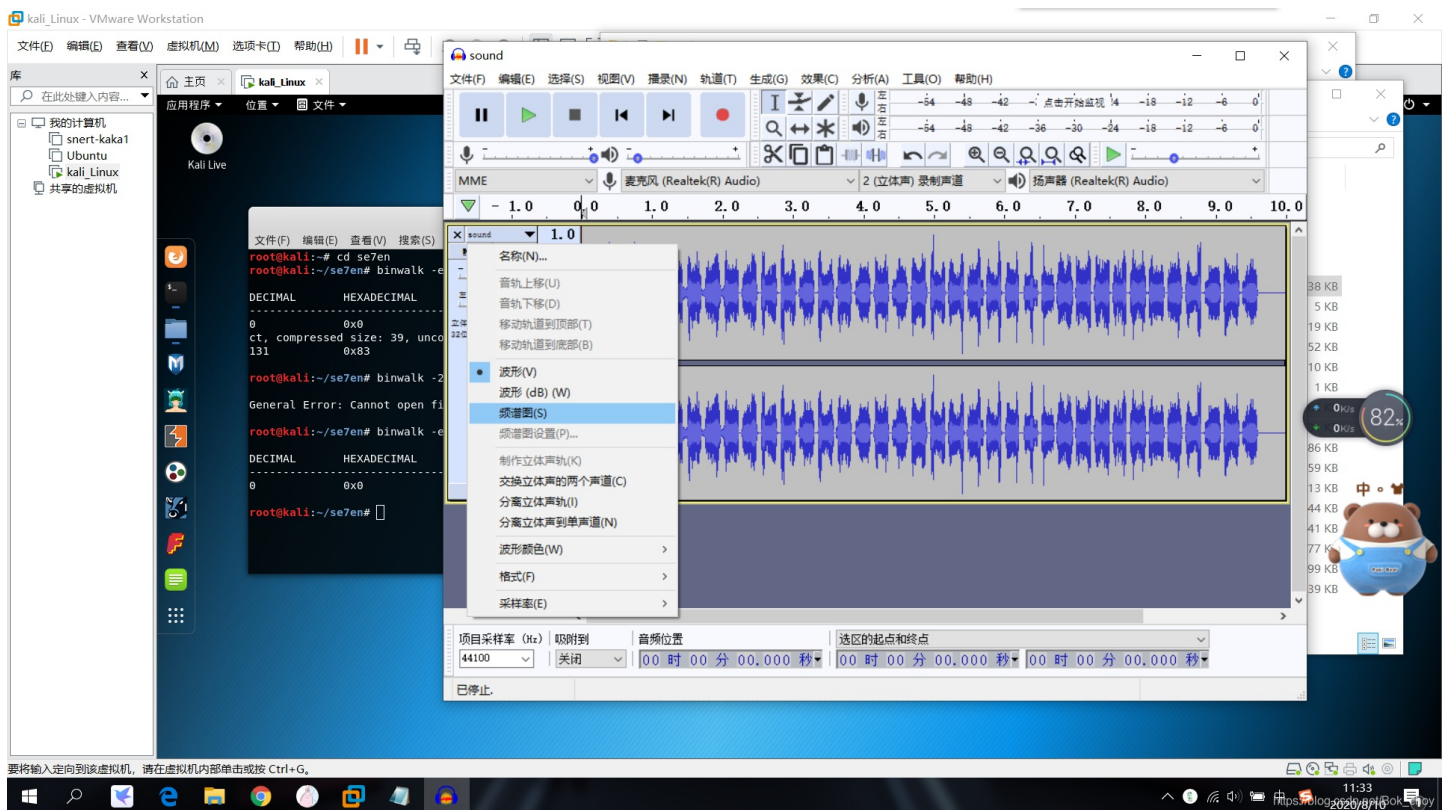
https://blog.csdn.net/Bok_choy

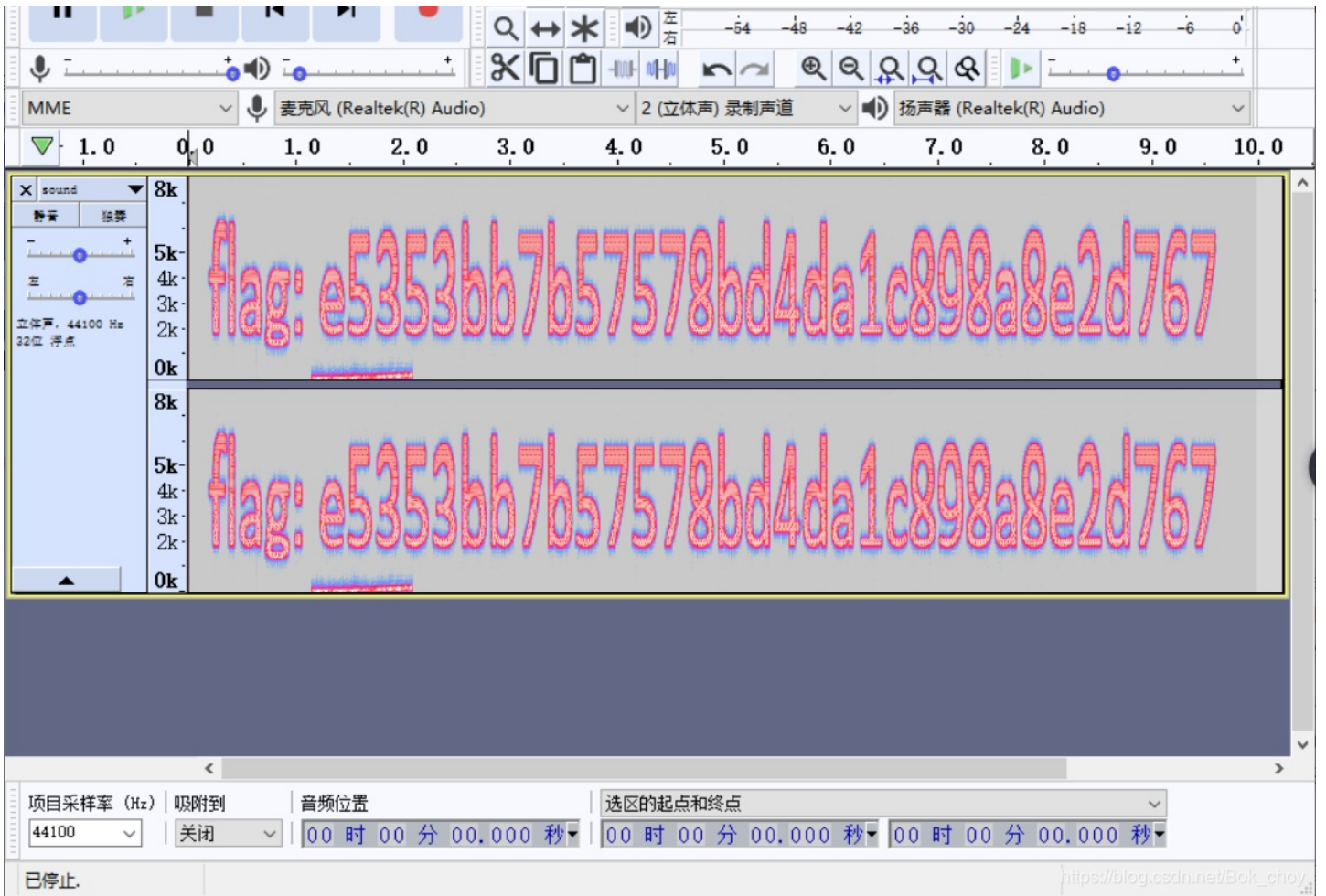
下载解压记事本打开



看见sound.wav再结合题目就知道是音频隐写啦

在虚拟机中提取文件得到sound.wav的文件，然后用工具Audacity打开，选择频谱图得到flag





What-is-this

What-is-this  7 最佳Writeup由MOVI提供

难度系数:  2.0

题目来源: [su-ctf-quals-2014](#)

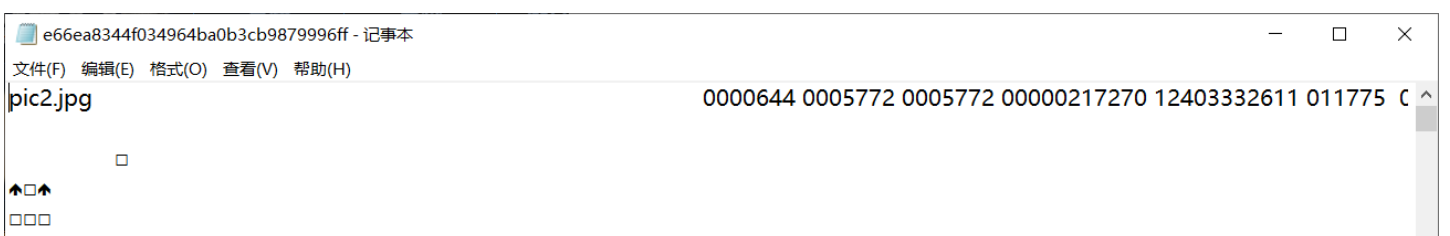
题目描述: 找到FLAG

题目场景: 暂无

题目附件: [附件1](#)

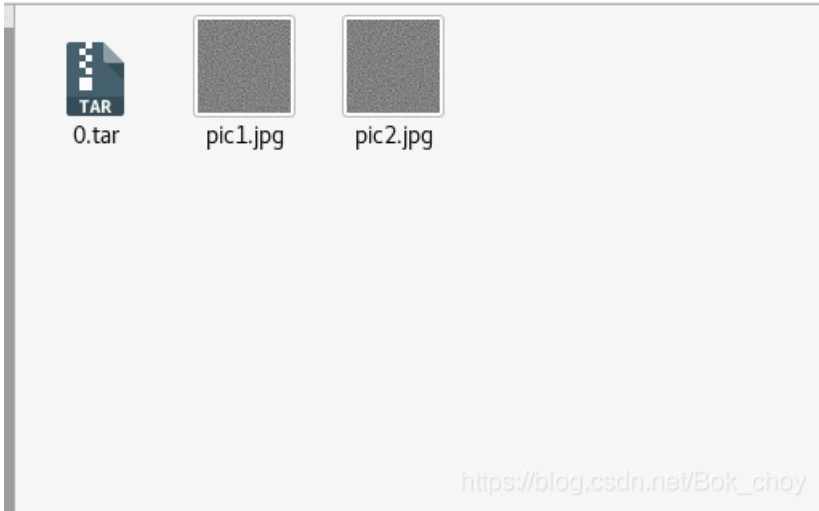
https://blog.csdn.net/Bok_choy

同样用记事本打开看到jpg

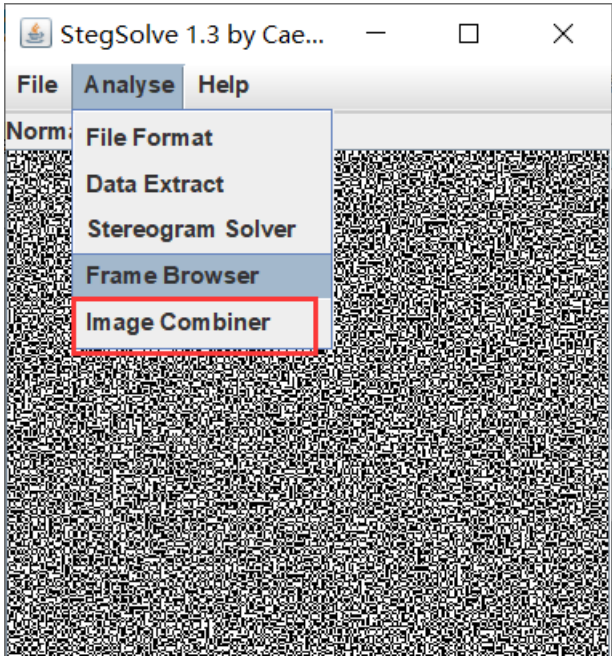


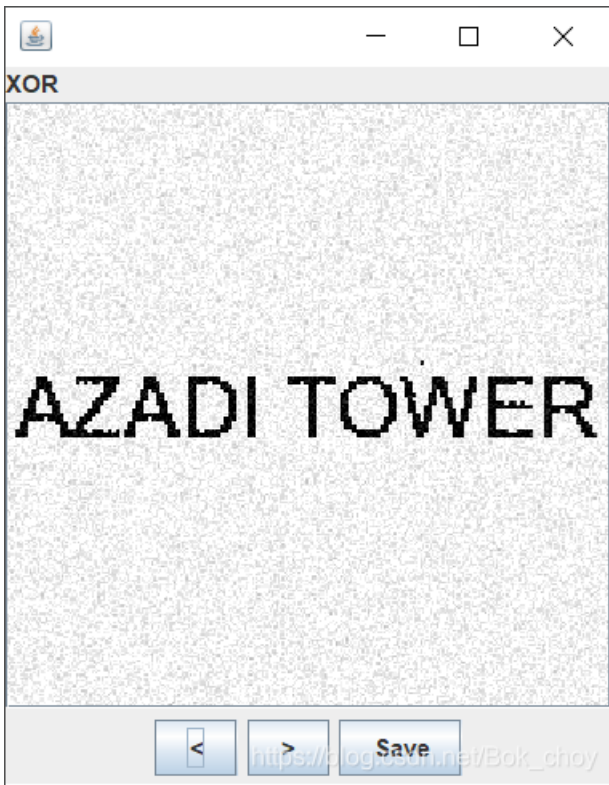
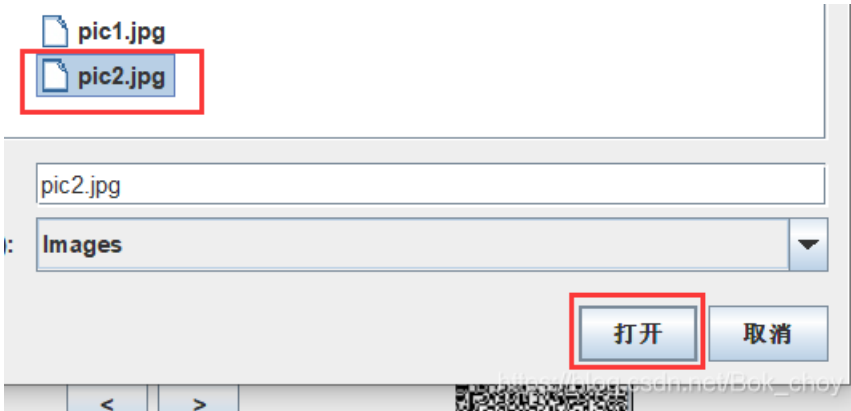

```
U
000000
^0000000000 ?000,0,000 ?0 00000 0000 ?40 00 0000 0 00000 000000000#$% 0"&0!' 6123457C ?000 ?0
5壶0碣0M泄 ?噴屬滔来}枯?鶻?鶻O'紗吳1μ變E鯉腦!?蚩?♣ 柳K笏迷R)?no
w<璦哄?]? Le坪尤漸bu判賄?|鉅ta€穩夙狄?7蛆?狄啓枕0証錠w縉T俚]0蚰咄f40N?MT0t?半? a饶峻<瓊0冷?r褐>毡[?驻#6库獨z禡窈滯
??€霽欄0隨韜演 鈇瓊9 ?Ar龐馱廠渠瘰 LFW^?E??p締 蟻!尘习踏0系 鞋? ?穎??苾r蘭+n狉t?匱?鏗?堵宝 [[?05櫓螯t秤7]秩曝U非?|解儼??頌
踵護鸚Cl3告睚k几 涑o 械蟾?7樞?9sj*蝱w,S p)喘S輕0q~ ,^僂TU?]羨 叩醞養x??滙n赫駁0啞6?7+尉嚙0枳=?0猓螳)嬭_閏鵲曝⊂鸞.:
2?橫^0? ~'? "縱co8??N?绕
輻3汗 櫛?雇C抱q洋蹄??n?坵4環[] WL5*摻W裊漿駐???,神v9? 繁?亂痼(鯉1} 醜旆屨鑿?峽廳Zh損i 折蝗仗>[♣?ブ鄒k阜 ~黃+佻卒0LW~浚
0 禱~御?頃?壘 P8縉?倫 h(頓Som
旗?鯨^啼懼_濼?瑣掌? 0yo29?[R O退鑲貺?元 0=勸唾H#襍0嚙0窺Y# Q好?y薦??炜茲G 0]?G薑J欵?> \^?劬{kS0?A弭NA
扶腕?F(4)嶠0%,駁 0?r?'0 縵=n炕鞍1閩 綯+_商鄱OF鈇]F?H 騰錚\?]樽]l4媼0饒i縹黍>*潤鎧鵬苙 詞指p>=9沓<:歿撐綫m20vy;灌 00S
[:0nnè e?紕RF)0?kk蔗雍柒?J腔#?/T縣蒙? 90? 確4甚 ~?場 5:拗0蠅0v渴膜+營? U託0 卍 Q?娛nOh玕鸚?n8J鎡?=袖揀_O墜晏?6璽? 黠
gm0Y先00釘?書y/抵5投嬭?鶻達?Z'滅?O蔽?职貯117Som 濼?C道莢韵*鯉
]?0d f?0価?駝=距啣香岌菝+M:籽m@ 杔#?Z莫Y?]?跨國墮?謫漸希耐禡(鸞0时<漸=3闢Rt霧0:@閩01?n鈞gd?輶悟G萍乐A]葭%K>
??Q0GQM詭F_虞鎗銜Y狄r觀y0渡斜慄慄 11k?郛劍00鈴m8?豈o??*h靈嘲VUw 錚轟况瘴?消D{挤 GB阡★h續隄?哲0-舶間c 0c9?00?緋赴
濼q止駭w侈彳b?=P(粒B! 詔&讀≥循?梏B ??慶J的M網寫瓶?w0a鷺鴿S 眈? 稂眾參存10{MZ9>Fl?n~9咕?3濼c 0駱?碓鉞惶0
肝G?)?絲0軀\,%鉅搖f隨V?0媿仍|a讞紘姪拾/5"Y0須H0槩× # 惹rD"0u蠅?璣璣劍<俞羶o?眩bN枷b KS WKWQ" 饋埤?鸞c祿?vii]稭0 ii F?)
昇爻X?"n ?h0棟安??9涵穉謎h"W聞杰N濯?眈 鬣又濼4卅?綵稟XF鯨替p0倍墳葬!魄 酷暖r鉤蝨?顛跽00压禱;健!嫵K0懶 薰0哩60+澎0{4f
>悻? 攬"O?5鶻0袂?哀压音"<痾包_p畝
0?壘:緯U翺濼?C?梅WO0肥_d閱E昉迭/<鞞??介2x養(掬0)/榴
```

提取文件



跟上面那个Tom差不多啦，两张相似的图片我们就去作比较





MISCall

MISCall

👍 11

最佳Writeup由我们是来学习的 • Cony提供

难度系数: ★★ 2.0

题目来源: noconname-2014-quals

题目描述: 没有提示

题目场景: 暂无

题目附件: 附件1

https://blog.csdn.net/Bok_choy

没发现什么信息，也不知道文件是什么类型的
在虚拟机跑一下

```
file 文件名
```

得到文件类型是一个压缩包，将文件类型改为bzip2

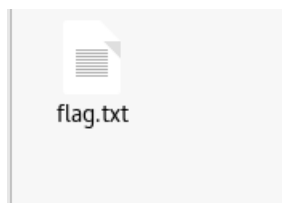
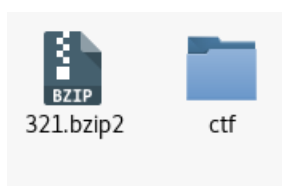
```
root@kali:~/seven# file 123
123: POSIX tar archive (GNU)
root@kali:~/seven# file 321
321: bzip2 compressed data, block size = 900k
root@kali:~/seven#
```

解压压缩包

```
tar xvf 文件
```

```
root@kali:~# tar xvf 321.bzip2
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
```

https://blog.csdn.net/Bok_choy



但是打开后什么也没有.....然后没有然后了

观察目录，发现都有git文件目录，参考大佬的wp，得知考察 git文件泄露、git stash命令

https://blog.csdn.net/qq_46927150/article/details/105880794

查看git记录

```
git log
```

```
root@kali:~# cd ctf# git og
git: 'og' 不是一个 git 命令。参见 'git --help'。

最相似的命令是
lon
root@kali:~# cd ctf# git log
commit bea99b953bef6cc2f98ab59b10822bc42afe5abc (HEAD -> master)
Author: Linus Torvalds <torvalds@klaava.Helsinki.Fi>
Date: Thu Jul 24 21:16:59 2014 +0200

Initial commit
root@kali:~# cd ctf#
```

https://blog.csdn.net/Bok_choy

查看修改列表

```
git stash list
```

```
log
root@kali: /ctf# git log
commit bea99b953bef6cc2f98ab59b10822bc42afe5abc (HEAD -> master)
Author: Linus Torvalds <torvalds@klaava.Helsinki.Fi>
Date: Thu Jul 24 21:16:59 2014 +0200

Initial commit
root@kali: /ctf# git stash list
stash@{0}: WIP on master: bea99b9 Initial commit
root@kali: /ctf#
```

校验列表的存储文件，发现有文件改动s.py

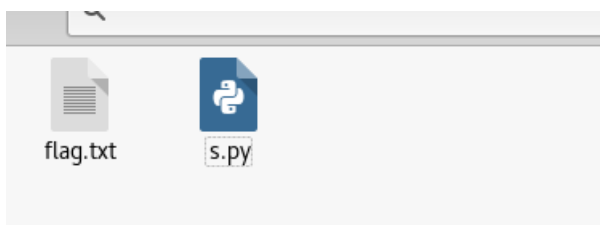
```
git stash show
```

```
log
root@kali: /ctf# git log
commit bea99b953bef6cc2f98ab59b10822bc42afe5abc (HEAD -> master)
Author: Linus Torvalds <torvalds@klaava.Helsinki.Fi>
Date: Thu Jul 24 21:16:59 2014 +0200

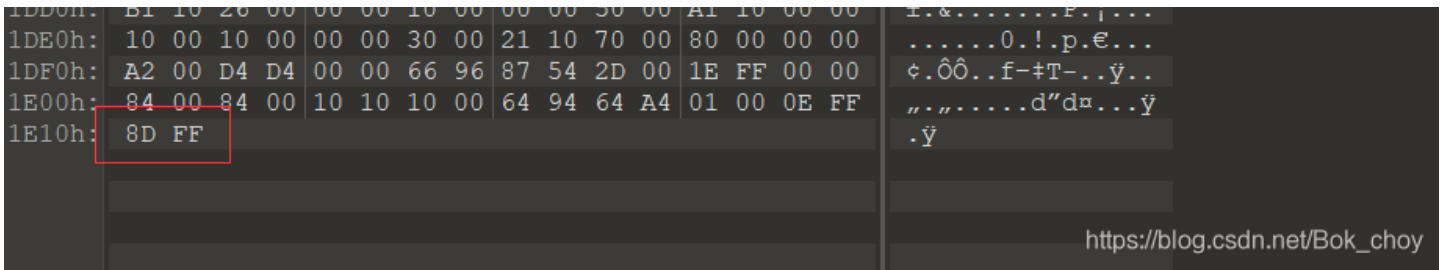
Initial commit
root@kali: /ctf# git stash list
stash@{0}: WIP on master: bea99b9 Initial commit
root@kali: /ctf# git stash show
flag.txt | 25 ++++++
s.py | 4 +++
2 files changed, 28 insertions(+), 1 deletion(-)
root@kali: /ctf# https://blog.csdn.net/Bok\_choy
```

将s.py导出来运行

```
git stash apply
```



```
python s.py
```

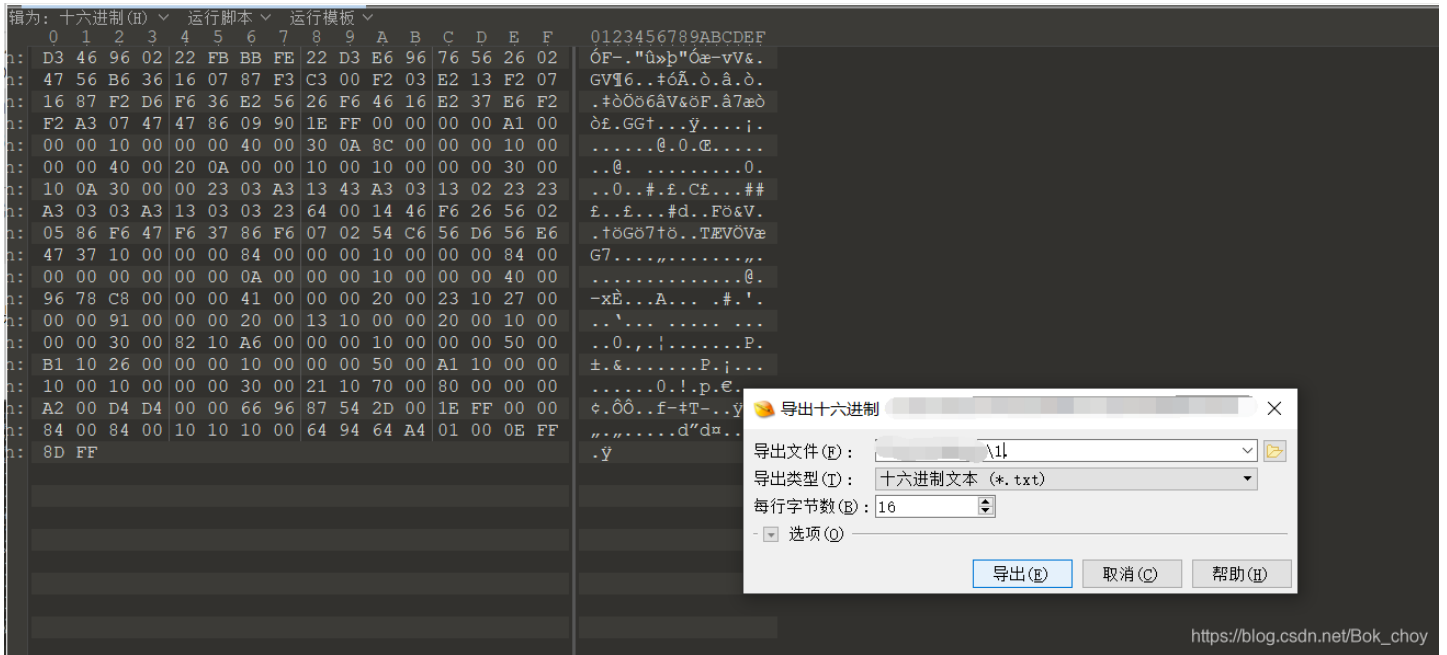



文件头尾信息总结

转载 tcrefreshingbreeze 2018-08-15 14:04:38 1547 收藏 2

JPEG (jpg),	文件头: FFD8FF	文件尾: FF D9
PNG (png),	文件头: 89504E47	文件尾: A
GIF (gif),	文件头: 47494638	文件尾: 00
Archive (zip),	文件头: 504B0304	文件尾: 50 4B

再结合题目，倒过来，就很明显是16进制给倒着输入了，我们需要把他正回来才可以得到正常的图片



然后再倒序输出得到正确的16进制

```
import os

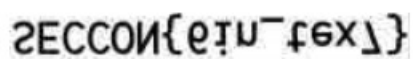
f = open('3.jpg', "rb") # 二进制形式打开
f = f.read()[::-1]
for i in f:
    ans = str(hex(i))[2:][::-1]
    if len(ans) == 1:
        ans = ans + '0'
    print(ans, end='')
```

```
import os

f = open('3.jpg', "rb") # 二进制形式打开
f = f.read()[::-1]
for i in f:
    ans = str(hex(i))[2:][::-1]
    if len(ans) == 1:
        ans = ans + '0'
    print(ans, end='')
```

https://blog.csdn.net/Bok_choy

输出的文件在010editor中导入16进制得到图片



https://blog.csdn.net/Bok_choy

将他翻转得到flag啦

something_in_image

something_in_image

最佳Writeup由admin提供

难度系数:  2.0

题目来源: 2019湖湘杯

题目描述: 暂无

题目场景: 暂无

题目附件: [附件1](#)

https://blog.csdn.net/Bok_choy

放进虚拟机运行命令

```
strings 文件|grep Flag
```



```
Flag.txtt.swx
.Flag.txt.swpe
.Flag.txt.swx
.Flag.txt.swpe
Flag.txtt.swx
.Flag.txt.swpe
Flag.txtt.swx
.Flag.txt.swpe
loft frgmework swx
.Flag.txt.swpe
.Flag.txt.swx
.Flag.txt.swpe
Flag.txtt.swx
Flag.txt
Flag.txt
Flag.txt
Flag.txt
/mnt/test/Flag.txt
Flag{}
Flag{yc4pl0fvjs2k1t7T}
/mnt/test/Flag.txt
Flag{}
Flag{yc4pl0fvjs2k1t7T}
```

https://blog.csdn.net/Bok_choy

Flag{yc4pl0fvjs2k1t7T}

打野

打野

👍 3 最佳Writeup由admin提供

难度系数: ★★ 2.0

题目来源: 强网杯2019

题目描述: 菜你了解CTF圈的实时动态么? flag格式qwxf{}

题目场景: 暂无

题目附件: 附件1

https://blog.csdn.net/Bok_choy

虚拟机跑命令没得到什么信息

```
root@kali:~# binwalk -e 瞅啥.bmp
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	PC bitmap, Windows 3.x format,, 1224 x 689 x 24
681325	0xA656D	Unix path: /M0.L1/M20N42P53Q0.K0.K0.K/-J/-J/-J/-J ,I/-J/-J0.K0.K1/L1/L1/L1/L53P64Q64Q75R86S97T97T97T64Q75R86S:8U<:W><Y@>[@>[GF`GF` GF`HGaKJd
684991	0xA73BF	Unix path: /M1/M0.L0.L0.L1/M31042P0.K0.K/-J.,I.,I- +H-+H-+H-+H-+H-+H.,I.,I/-J/-J/-J31N31N42042053P64Q75R75R53P64Q86S:8U<:W><Y?=Z@>[DC]DC]ED^
1390195	0x153673	Unix path: /C/0D/0D/0D67K78L9:N;<P=>R?@TABVBCWJYH KZJM\L0^MP_NQ`NQ`NQ`KN]IL[HKZGJYFIXCFU@CR<?N<?N;>M9<K8;J69H69H58G58G/2A03B14C25D 58G8;J:=L
1404862	0x156FBE	Unix path: /B20C31D-/A-/A-/A-/A/1C13E24F35G46H57I7 9K:<N>@RACUCEWDFXDGUDGUEHVFIWGXGJXFIWFIWFIWCFT?BP<?M:=K9<J69G47E03B/2A.1@-0?-0? -0?-0?.1@
1415857	0x159AB1	Unix path: /D./D89N56K34I45J67L67L34I/0E.0B.0B.0B/ 1C14C36E58F69G8;I8;I9=H:>I=AL@DOCGREITHISFGQFGQHISKLVKLVFGQBCMABL@AK=>H;<F89C67A 56@56@31D
1419881	0x15AA69	Unix path: /-/0-/0-/0-/0+0_01/1213424546757868968 968979:9:<:<:=:>?=@99?>>DEEK11P11PEEK??E:~@99?77=228...2(((((++1--34896:~9=>

stegsolve跑一下无果

运行该命令得到flag

```
zsteg 瞅啥.bmp
```



```
root@kali: # zsteg 倒立屋.png
imagedata .. text: "\t\t\t\r\r\r\r\r\r\r\r"
b1,rgb,lsb,xy .. text: "IsCc_2019"
b2,r,msb,xy .. text: "t^y\t_{!i0"
b2,g,msb,xy .. text: "UUUUUU`\rUUUU"
b2,b,msb,xy .. text: "UUUUUU`\rUUUU"
b2,rgb,msb,xy .. text: "jZ]?0]k0"
b4,r,lsb,xy .. text: "#UwcDS#z"
b4,r,msb,xy .. text: ["f" repeated 8 times]
b4,g,lsb,xy .. text: "w17ffd2T3EB"
b4,g,msb,xy .. text: "wwwwwwwww3{"
b4,b,lsb,xy .. text: "ffffwww"
b4,b,msb,xy .. text: "ffffffffffff\j"
b4,rgb,lsb,xy .. text: "iVugVUUU6"
b4,bgr,lsb,xy .. text: "YevWUUUV" https://blog.csdn.net/Bok_choy
```

因为是倒立，所以要反过来，flag是9102_cCsl

2017_Dating_in_Singapore

2017_Dating_in_Singapore 👍 4 最佳Writeup由admin提供 WP 建议

难度系数: ★ ★ 2.0

题目来源: XCTF 3rd-HITB CTF-2017

题目描述: 01081522291516170310172431-050607132027262728-0102030209162330-02091623020310090910172423-02010814222930-0605041118252627-0203040310172431-0102030108152229151617-04050604111825181920-0108152229303124171003-261912052028211407-04051213192625

题目场景: 暂无

题目附件: 附件1

https://blog.csdn.net/Bok_choy

直接上链接= https://blog.csdn.net/qq_42016346/article/details/104234416

simple_transfer

simple_transfer

👍 6 最佳Writeup由B301 • dals提供

难度系数: ★★ 2.0

题目来源: XCTF 3rd-HITB CTF-2017

题目描述: 文件里有flag, 找到它。

题目场景: 暂无

题目附件: 附件1

https://blog.csdn.net/Bok_choy

用binwalk查看一下

```
root@kali:~# binwalk -e 111.pcap
```

DECIMAL	HEXADECIMAL	DESCRIPTION
339380	0x52DB4	PDF document, version: "1.5"
339454	0x52DFE	Zlib compressed data, default compression
340171	0x530CB	Zlib compressed data, default compression
6380104	0x615A48	Zlib compressed data, default compression
6385002	0x616D6A	Zlib compressed data, default compression

用foremost将PDF提取出来

HITB{b3d0e380e9c39352c667307d010775ca}

https://blog.csdn.net/Bok_choy

Erik-Baleog-and-Olaf

Erik-Baleog-and-Olaf

👍 7 最佳Writeup由HeliantHuS提供

难度系数: ★★2.0

题目来源: tinyctf-2014

题目描述: 暂无

题目场景: 暂无

题目附件: 附件1

https://blog.csdn.net/Bok_choy

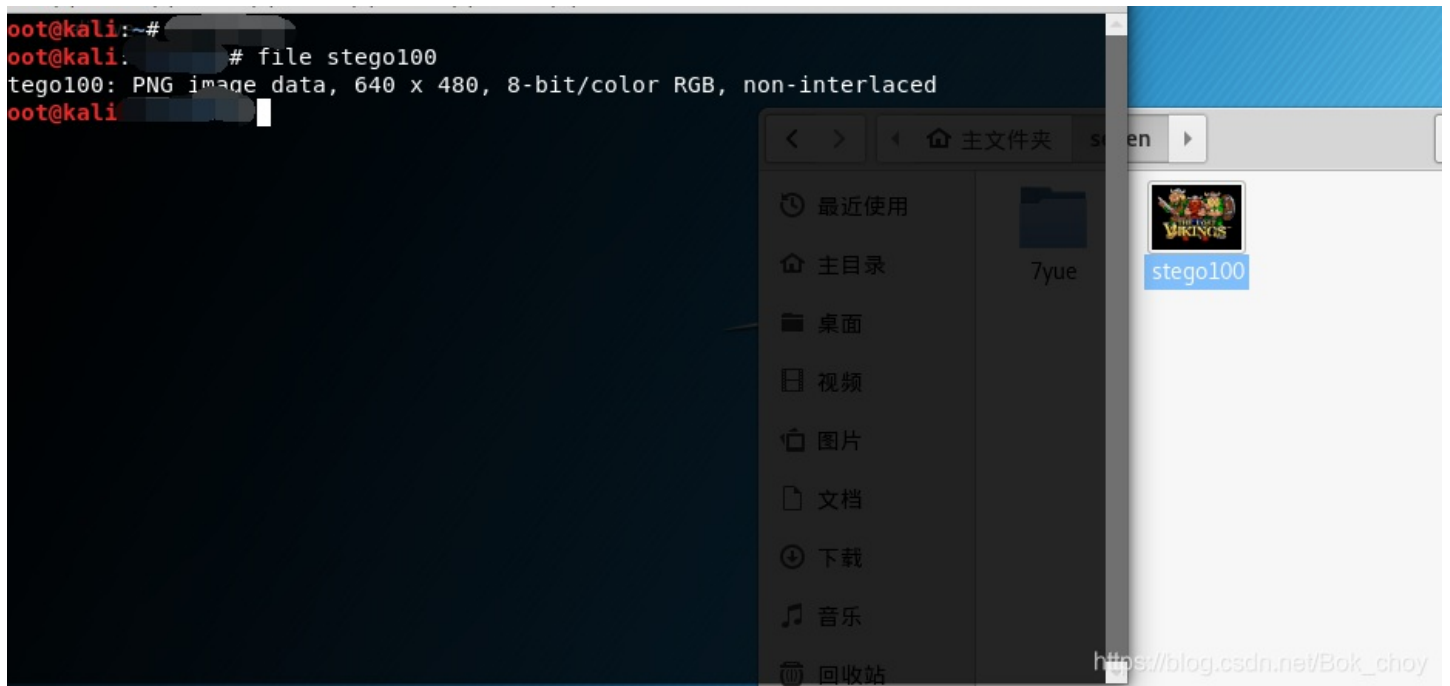
第一步打开记事本，尾端有个链接，尝试进入却打不开

```
00霽2?彘 #tEXthint: http://i.imgur.com/22kUrzm.png > IEND 晒?
```

第 1 行, 第 1 列 100% Windows (CRLF)

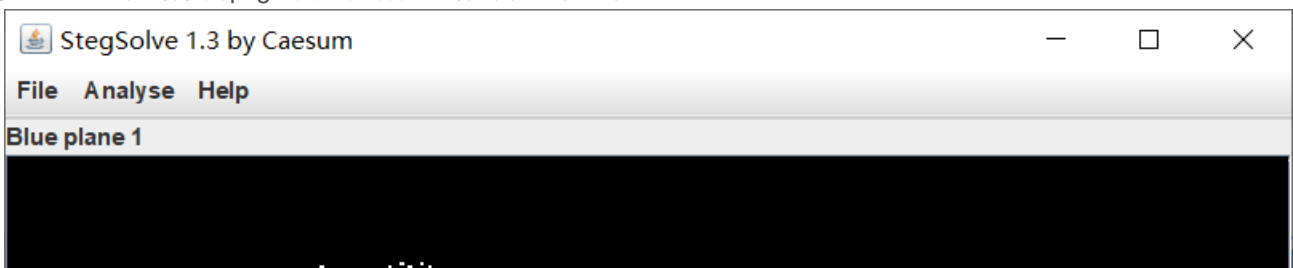
查看文件类型

```
oot@kali:~#  
oot@kali:~# file stego100  
stego100: PNG image data, 640 x 480, 8-bit/color RGB, non-interlaced  
oot@kali:~#
```



https://blog.csdn.net/Bok_choy

是png类型，于是后缀改为png，用工具打开，看到了一个二维码





二维码扫码扫不上，看大佬说要变化通道找到合适的通道才能扫码，也有说找个清晰一点的二维码重新查看然后扫码，我太菜了没找到.....

查看隐写，还是这个链接，但是同样在虚拟机中也进不去

```
root@kali: ~ # zsteg stego100
meta hint .. text: "http://i.imgur.com/22kUzrm.png"
root@kali: ~ # strings stego100 | grep -i flag
```

出题人应该是想打开链接下载原图然后就可以把两个图作比较最后得到flag，可惜链接打不开了
复制一个flag: flag{#justdiffit}

hit-the-core

hit-the-core 👍 8 最佳Writeup由sins7 • giun提供

难度系数: ★★ 2.0

题目来源: alexctf-2017

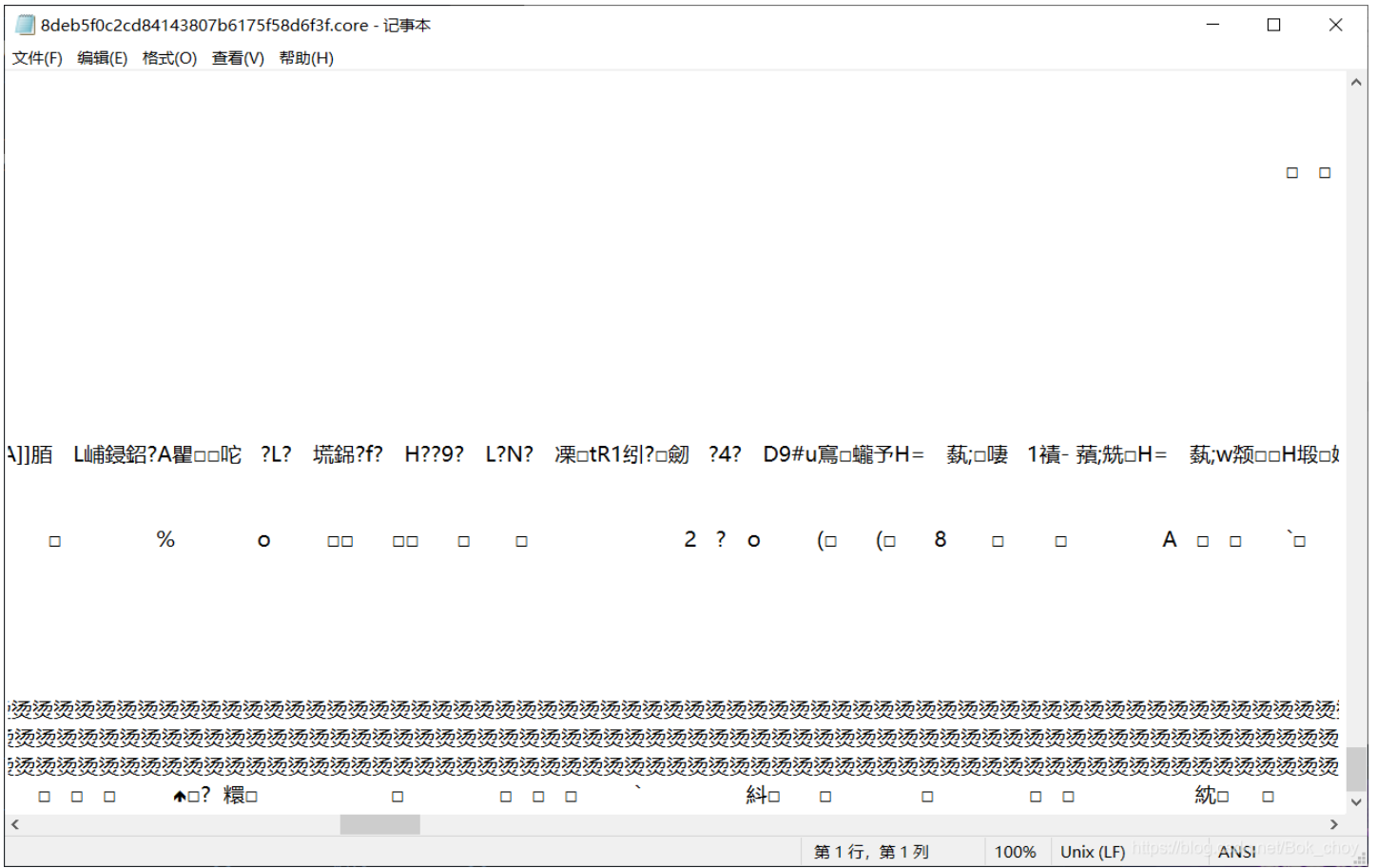
题目描述: 暂无

题目场景: 暂无

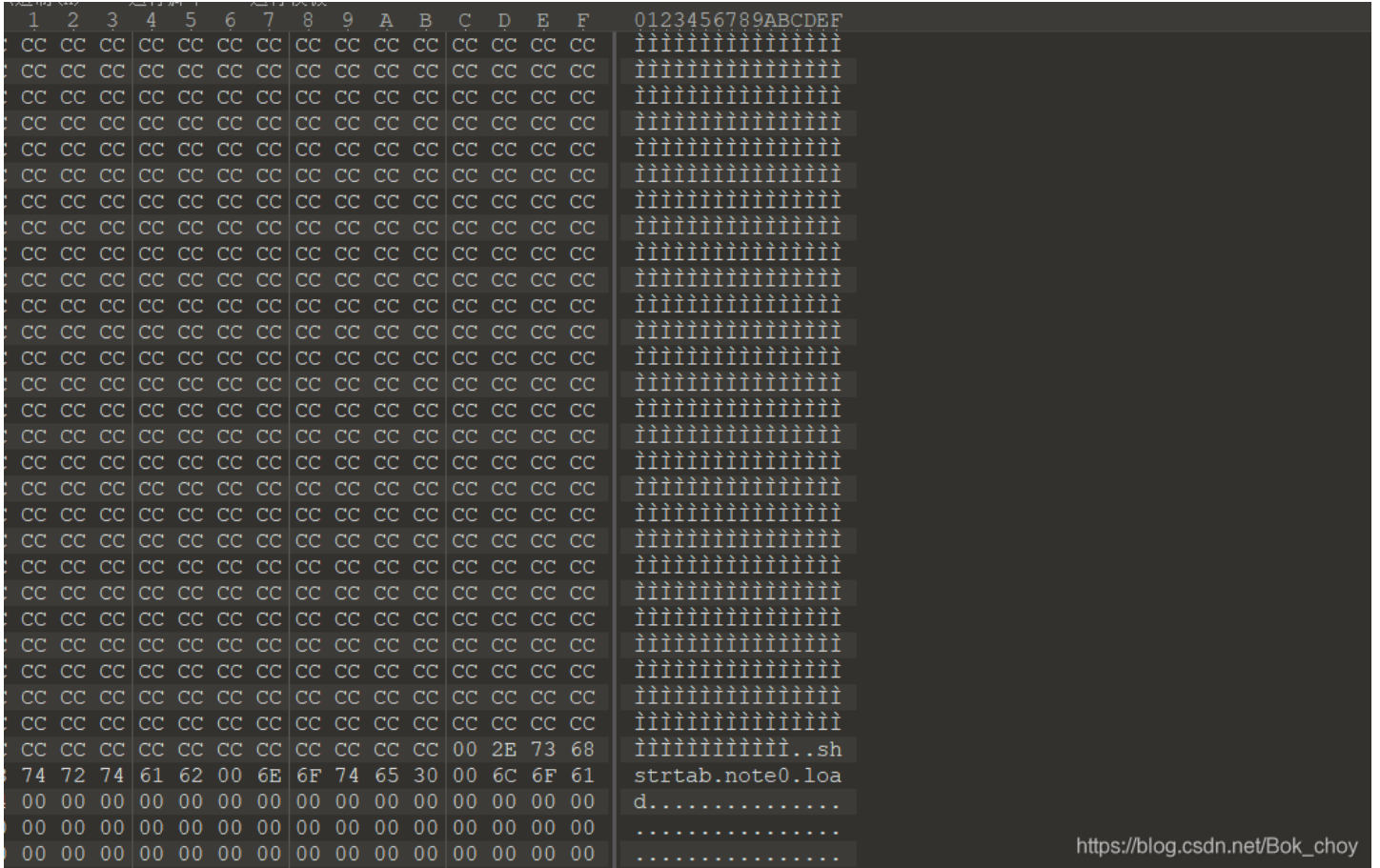
题目附件: 附件1

https://blog.csdn.net/Bok_choy

用记事本打开发现有很多烫烫烫，怀疑是一种加密方式



用010editor打开



无果

然后在kali里面查看是否有特殊的字符串

用string命令可以查看




```
AWAVA
AUATL
[]A\A]A^A
cvqAeqacLtqazEigwiXobxrCrtuiTzahfFreqc{bnjrKwgk83kgd43j85ePgb_e_rwqr7fvbmHjkl03t
ews_hmkogooyf0vbnk0ii87Drfgh_n_kiwutfb0ghk9ro987k5tfb_hjiouo087ptfcv}
(q9e
```

在众多字符串中看见这个有点像flag的，然后观察字符串，每五个字母为一个小组的话每个小组第四个都是大写字母，且遇到了第四个为{，然后我们都以这样的规律，每五组的第四个符号提取出来，得到flag
ALEXCTF{K33P_7H3_g00D_w0rk_up}

glance-50

glance-50  10 最佳Writeup由 **Kyrie • KyrieKiki** 提供

难度系数:  ★★ 2.0

题目来源: [mma-ctf-2nd-2016](#)

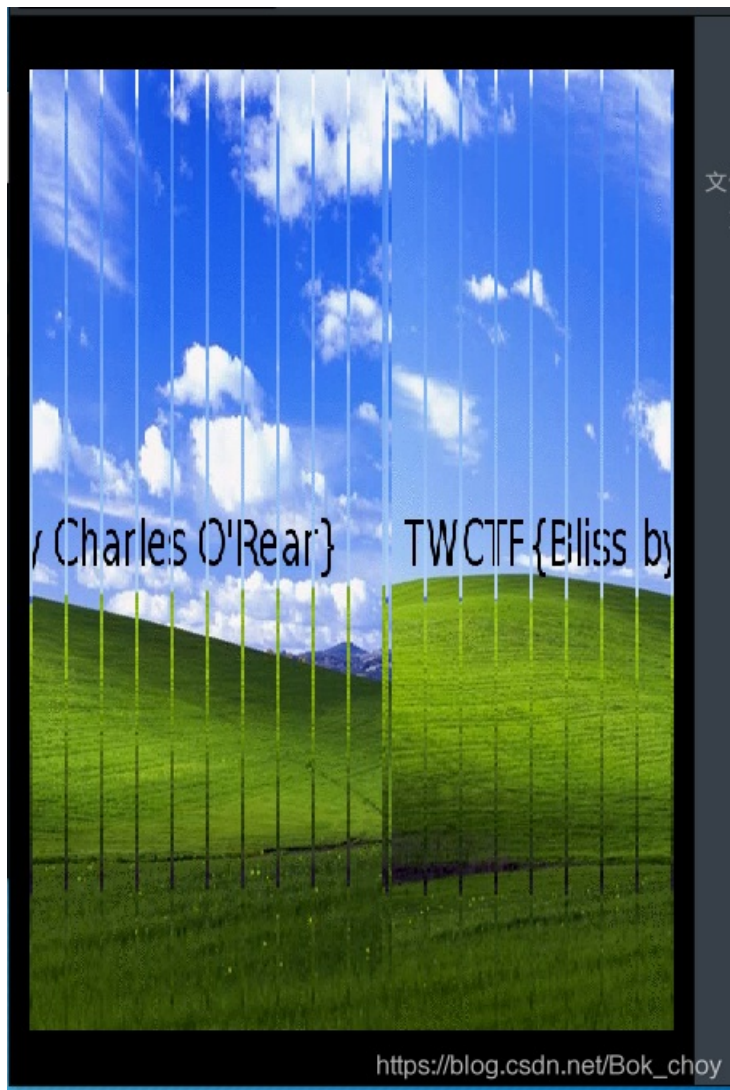
题目描述: 暂无

题目场景: 暂无

题目附件: [附件1](#)

https://blog.csdn.net/Bok_choy

https://blog.csdn.net/zz_Caleb/article/details/89490494



一个动态图片分解网站，直接出flag

<https://tu.sioe.cn/gj/fenjie/>

Ditf

Ditf 最佳Writeup由admin提供

难度系数: ★★ 2.0

题目来源: 安恒9月赛

题目描述: 暂无

题目场景: 暂无

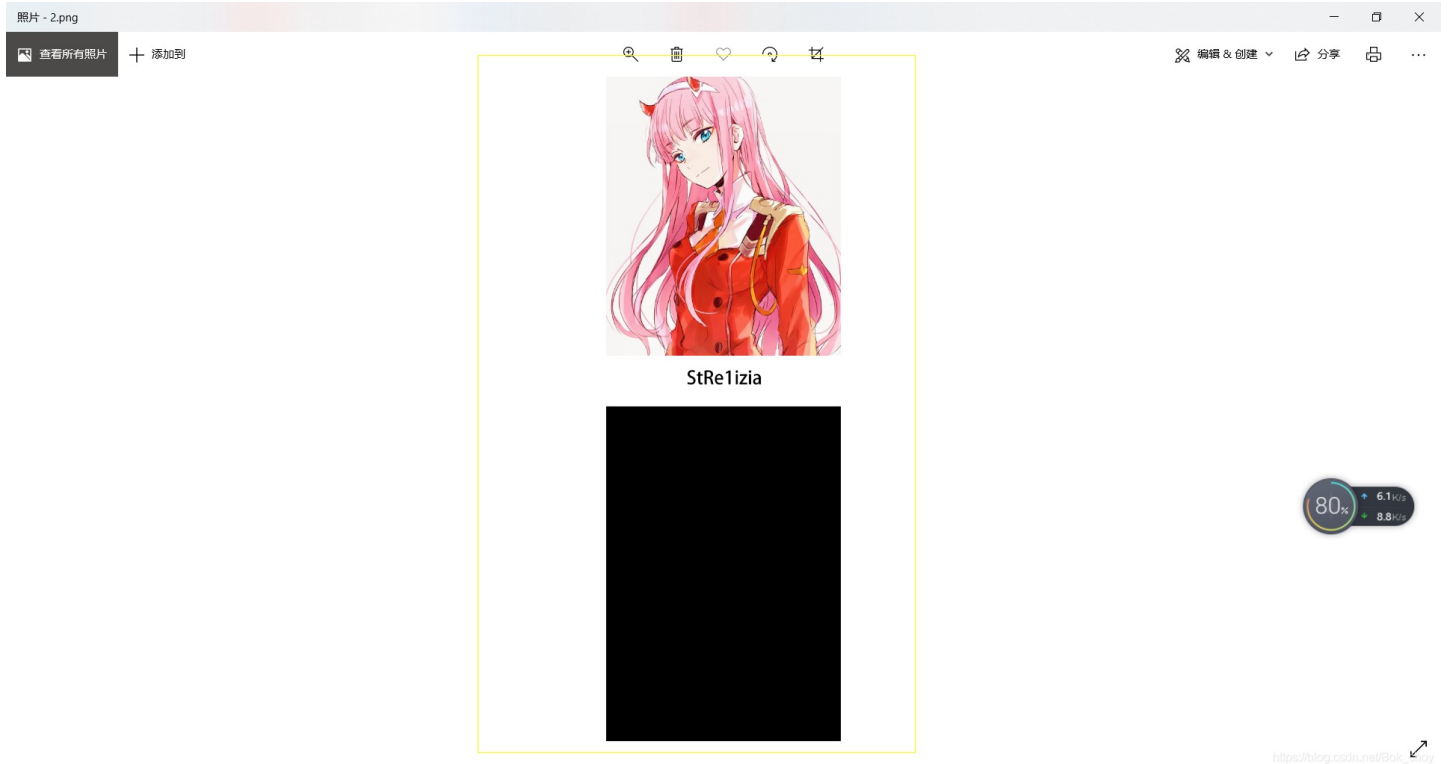
题目附件: 附件1

https://blog.csdn.net/Bok_choy

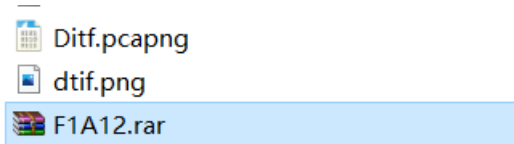
下载附件后提取出来，有一个png图片和一个压缩包，图片没有附带其他文件，也没有隐写，压缩包是加密的，尝试着暴力破解没有破解出来。后来还是回到了图片本身，图片应该藏有压缩包密码，改写图片的高度将图片的高度调大一点，可以查看是否藏

有密文

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	%PNG.....IHDR
00	00	03	9E	00	00	0A	3C	08	02	00	00	00	38	16	5A	...ž...8.Z
34	00	00	00	09	70	48	59	73	00	00	0B	13	00	00	0B	4...pHYs.....
13	01	00	9A	9C	18	00	00	06	D4	69	54	58	74	58	4D	...šœ...ôITxTXM
4C	3A	63	6F	6D	2E	61	64	6F	62	65	2E	78	6D	70	00	L:com.adobe.xmp.
00	00	00	00	3C	3F	78	70	61	63	6B	65	74	20	62	65	...<?xpacket be
67	69	6E	3D	22	EF	BB	BF	22	20	69	64	3D	22	57	35	gin="i»;" id="w5
4D	30	4D	70	43	65	68	69	48	7A	72	65	53	7A	4E	54	M0MpCehiHzreSzNT
63	7A	6B	63	39	64	22	3F	3E	20	3C	78	3A	78	6D	70	czkc9d"?)> <x:xmp
6D	65	74	61	20	78	6D	6C	6E	73	3A	78	3D	22	61	64	meta xmlns:x="ad
6F	62	65	3A	6E	73	3A	6D	65	74	61	2F	22	20	78	3A	obe:ns:meta/" x:
78	6D	70	74	6B	3D	22	41	64	6F	62	65	20	58	4D	50	xmptk="Adobe XMP
20	43	6F	72	65	20	35	2E	36	2D	63	31	34	32	20	37	Core 5.6-c142 7
39	2E	31	36	30	39	32	34	2C	20	32	30	31	37	2F	30	9.160924, 2017/0
37	2F	31	33	2D	30	31	3A	30	36	3A	33	39	20	20	20	7/13-01:06:39
20	20	20	20	20	22	3E	20	3C	72	64	66	3A	52	44	46	"> <rdf:RDF
20	78	6D	6C	6E	73	3A	72	64	66	3D	22	68	74	74	70	xmlns:rdf="http
3A	2F	2F	77	77	77	2E	77	33	2E	6F	72	67	2F	31	39	https://blog.csdn.net/Bok_choy
20	20	2F	20	22	2F	22	22	2D	72	64	66	2D	72	70	6F	88/02/22_rdf_gvr



得到了压缩包密码，进行解压



然后分析流量，尝试搜索flag，无果，搜索ctf，无果，搜索png（此处查看了其他大佬的wp），

No.	Time	Source	Destination	Protoc	Length	Info
7...	20.409...	123.206.131.1...	192.168.31.59	TCP	1458 80 → 33307	[ACK] Seq=81946 Ack=723 Win=31360 Len=1404 [TCP segment of a reassembled PDU]
7...	20.409...	123.206.131.1...	192.168.31.59	TCP	1458 80 → 33307	[ACK] Seq=83350 Ack=723 Win=31360 Len=1404 [TCP segment of a reassembled PDU]
7...	20.412...	123.206.131.1...	192.168.31.59	TCP	1458 80 → 33307	[ACK] Seq=84754 Ack=723 Win=31360 Len=1404 [TCP segment of a reassembled PDU]
7...	20.412...	123.206.131.1...	192.168.31.59	TCP	1458 80 → 33307	[ACK] Seq=86158 Ack=723 Win=31360 Len=1404 [TCP segment of a reassembled PDU]
7...	20.412...	123.206.131.1...	192.168.31.59	TCP	1458 80 → 33307	[ACK] Seq=87562 Ack=723 Win=31360 Len=1404 [TCP segment of a reassembled PDU]
7...	20.412...	123.206.131.1...	192.168.31.59	TCP	1458 80 → 33307	[ACK] Seq=88966 Ack=723 Win=31360 Len=1404 [TCP segment of a reassembled PDU]
7...	20.339...	123.206.131.1...	192.168.31.59	TCP	1458 80 → 33307	[ACK] Seq=8938 Ack=723 Win=31360 Len=1404 [TCP segment of a reassembled PDU]
7...	20.412...	123.206.131.1...	192.168.31.59	TCP	1458 80 → 33307	[ACK] Seq=90370 Ack=723 Win=31360 Len=1404 [TCP segment of a reassembled PDU]
7...	20.412...	123.206.131.1...	192.168.31.59	TCP	1458 80 → 33307	[ACK] Seq=91774 Ack=723 Win=31360 Len=1404 [TCP segment of a reassembled PDU]
7...	20.422...	123.206.131.1...	192.168.31.59	TCP	1458 80 → 33307	[ACK] Seq=93178 Ack=723 Win=31360 Len=1404 [TCP segment of a reassembled PDU]
7...	20.422...	123.206.131.1...	192.168.31.59	TCP	1458 80 → 33307	[ACK] Seq=94582 Ack=723 Win=31360 Len=1404 [TCP segment of a reassembled PDU]
7...	20.422...	123.206.131.1...	192.168.31.59	TCP	1458 80 → 33307	[ACK] Seq=95986 Ack=723 Win=31360 Len=1404 [TCP segment of a reassembled PDU]
7...	20.422...	123.206.131.1...	192.168.31.59	TCP	1458 80 → 33307	[ACK] Seq=97390 Ack=723 Win=31360 Len=1404 [TCP segment of a reassembled PDU]

7...	20.422...	123.206.131.1...	192.168.31.59	TCP	1458 80 → 33307 [ACK] Seq=98794 Ack=723 Win=31360 Len=1404 [TCP segment of a reassembled PDU]
1...	29.454...	123.206.131.1...	192.168.31.59	TCP	54 80 → 33307 [FIN, ACK] Seq=683127 Ack=723 Win=31360 Len=0
7...	21.629...	123.206.131.1...	192.168.31.59	TCP	1458 80 → 33307 [PSH, ACK] Seq=275698 Ack=723 Win=31360 Len=1404 [TCP segment of a reassembled PDU]
9...	24.480...	123.206.131.1...	192.168.31.59	TCP	1458 80 → 33307 [PSH, ACK] Seq=630910 Ack=723 Win=31360 Len=1404 [TCP segment of a reassembled PDU]
6...	20.304...	123.206.131.1...	192.168.31.59	TCP	66 80 → 33307 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1424 SACK_PERM=1 WS=128
6...	20.305...	192.168.31.59	123.206.131.1...	HTTP	432 GET / HTTP/1.1

追踪http流，看见有一串特殊字符，明显的base64加密

```

GET / HTTP/1.1
Host: 123.206.131.120
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9

HTTP/1.1 200 OK
Date: Sun, 01 Jul 2018 09:46:19 GMT
Server: Apache/2.4.18 (Ubuntu)
Last-Modified: Sun, 01 Jul 2018 09:45:26 GMT
ETag: "c6-56fecf0c66879-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 177
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  </head>
  <body>
    
    ZmxhZ3tPel80bmRfSGlyMF9sb3ZzX0ZvcjN2ZXJ9
  </body>
</html>

GET /kiss.png HTTP/1.1
Host: 123.206.131.120
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36
Accept: image/webp,image/apng,image/*,*/*;q=0.8
Referer: http://123.206.131.120/
Accept-Encoding: gzip, deflate

```

然后进行解密得到flag

flag{Oz_4nd_Hir0_lov3_For3ver}	ZmxhZ3tPel80bmRfSGlyMF9sb3ZzX0ZvcjN2ZXJ9
--------------------------------	--

flag{Oz_4nd_Hir0_lov3_For3ver}

4-1

4-1

最佳Writeup由admin提供

难度系数: ★ ★ 2.0

题目来源: **WDCTF-2017**

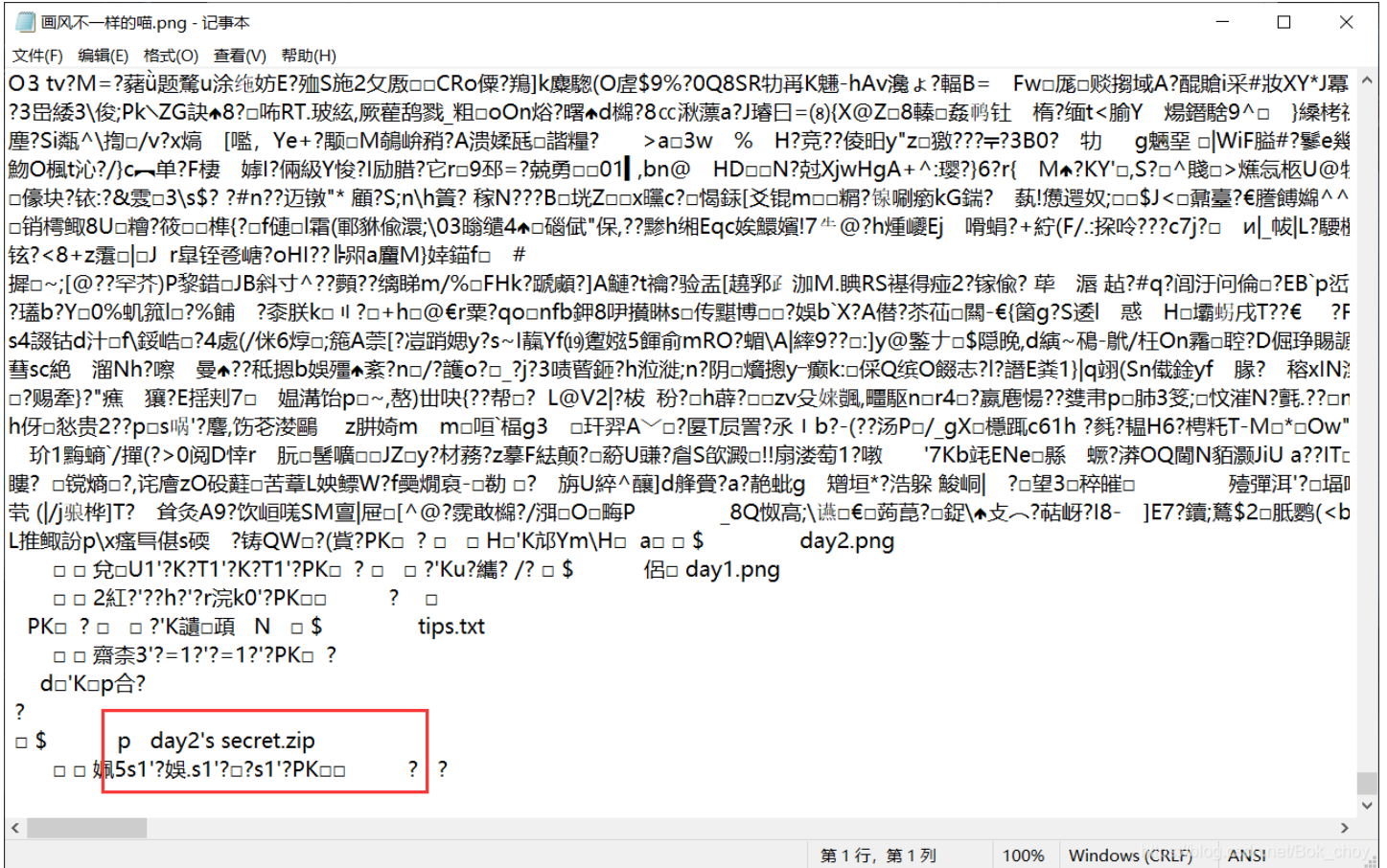
题目描述: 暂无

题目场景: 暂无

题目附件: **附件1**

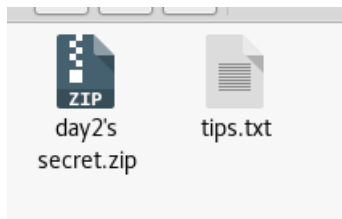
https://blog.csdn.net/Bok_choy

用记事本打开看见有zip的压缩包



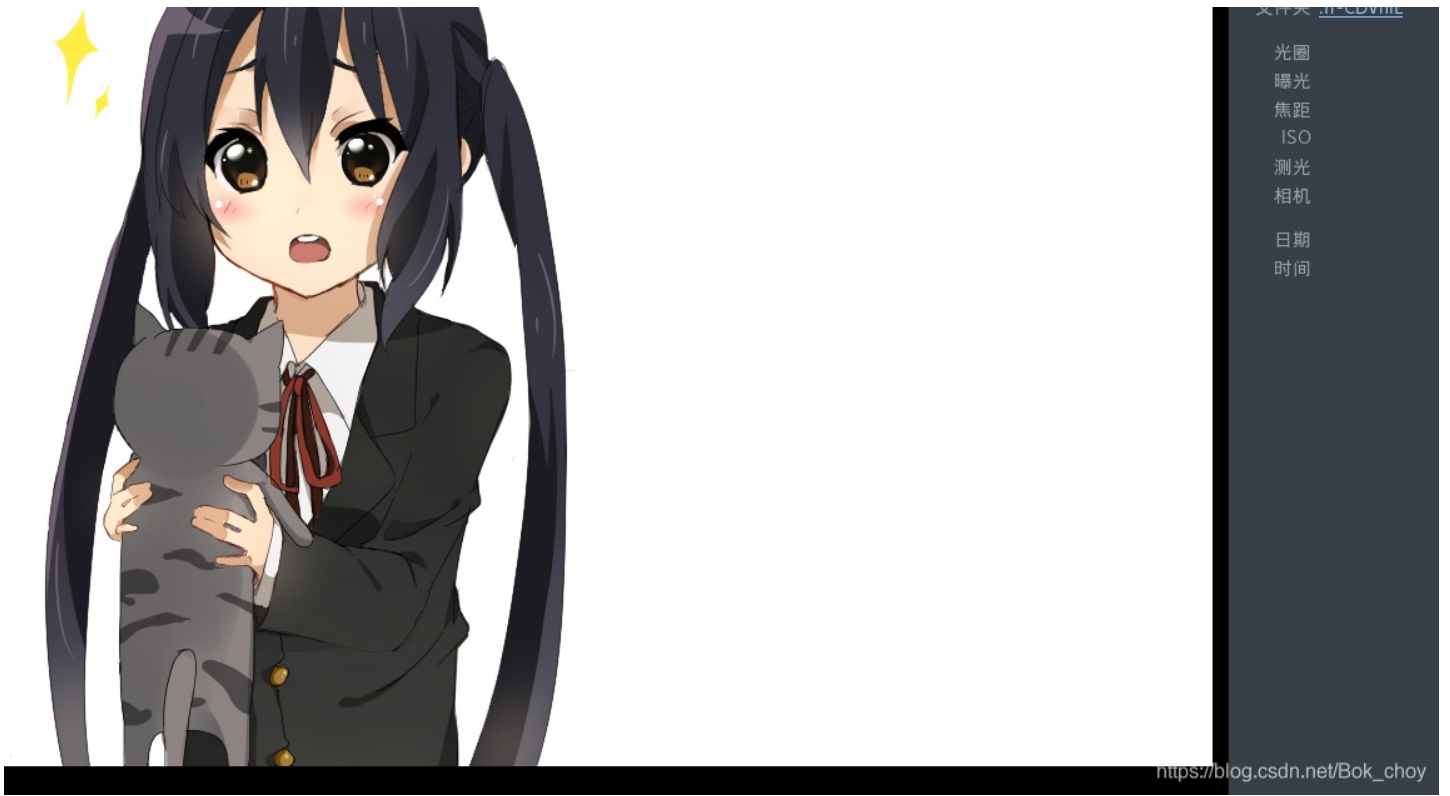
经过提取文档，发现有一个tips

```
Although two days doing the same things, but day2 has a secret than day1
- A1\A3 -|
```



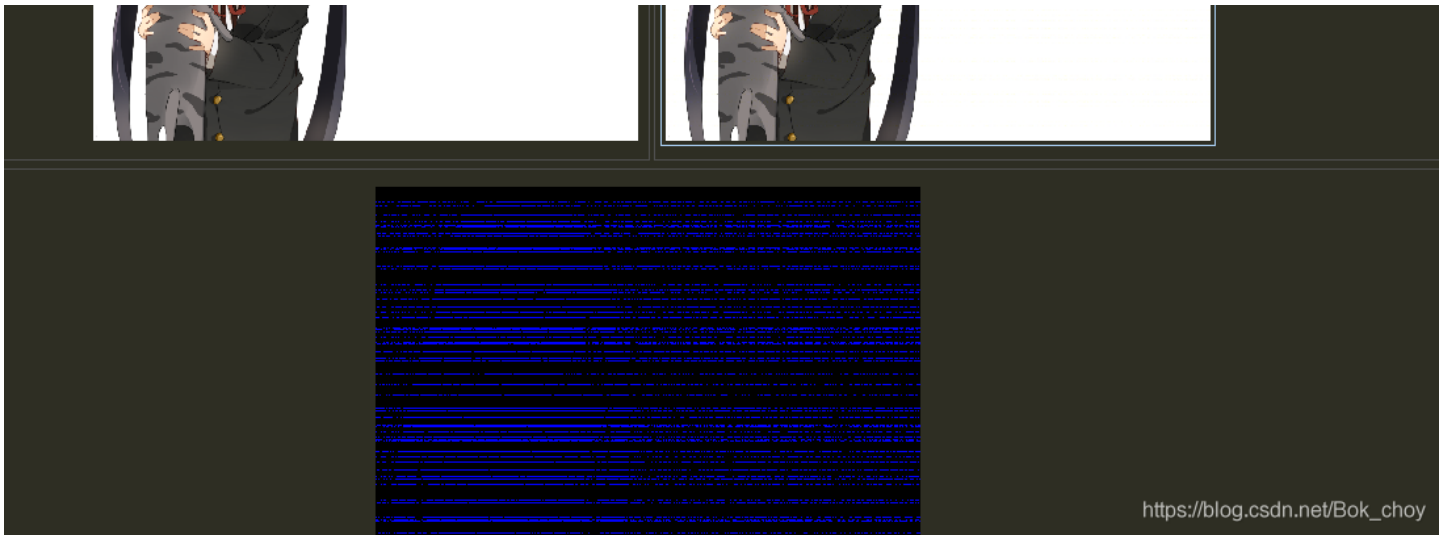
打开压缩包，里面有两张图，根据提示，第二张图是经过隐写的，两张图对比，可以看见day2的图文件大许多





一开始看见两个相似的图片，就会进行比较，看看是否有信息，但是这次没有





于是回到第二张图，查看有什么信息然后什么也没有查到.....

后来得知还有一种隐写方式是盲水印('·_·')

https://blog.csdn.net/qq_43504939/article/details/100673360?utm_medium=distribute.pc_relevant.none-task-blog-baidulandingword-1&spm=1001.2101.3001.4242

wdfлаг{My_c4t_Ho}

适合作为桌面

适合作为桌面

最佳Writeup由sins7 • giun提供

难度系数: ★★ 2.0

题目来源: 世安杯

题目描述: 暂无

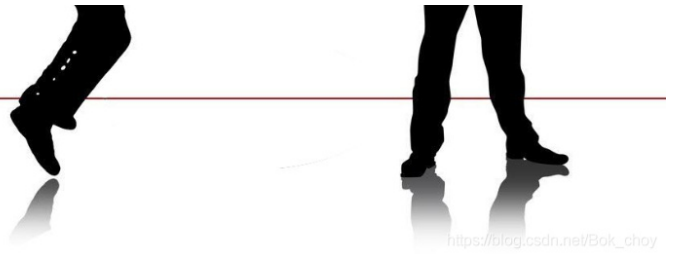
题目场景: 暂无

题目附件: 附件1

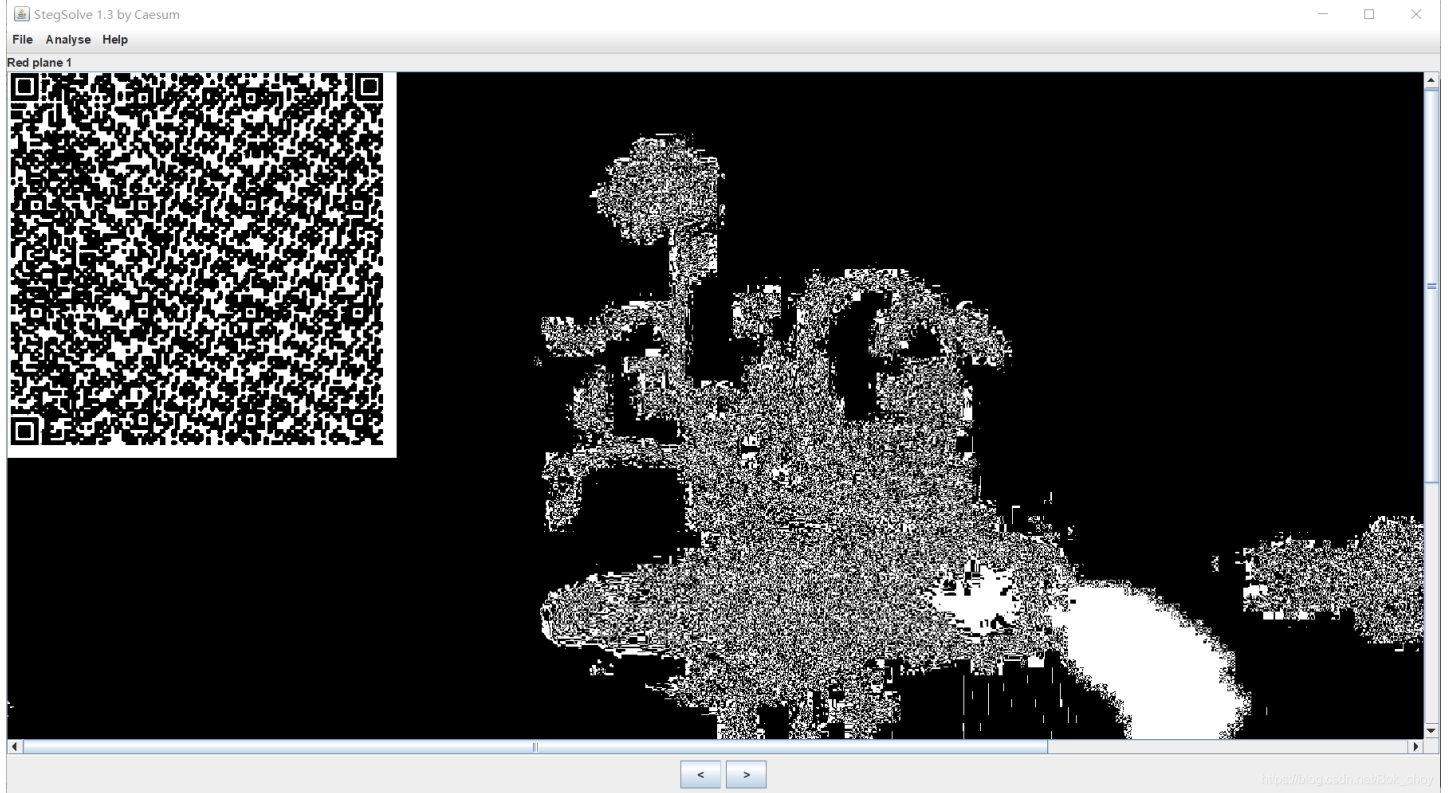
https://blog.csdn.net/Bok_choy

打开图片觉得蛮怪异的，第一想法是反色，然后操作一下得到新的图，正常多了嘻嘻~（虽然还不知道有什么用）





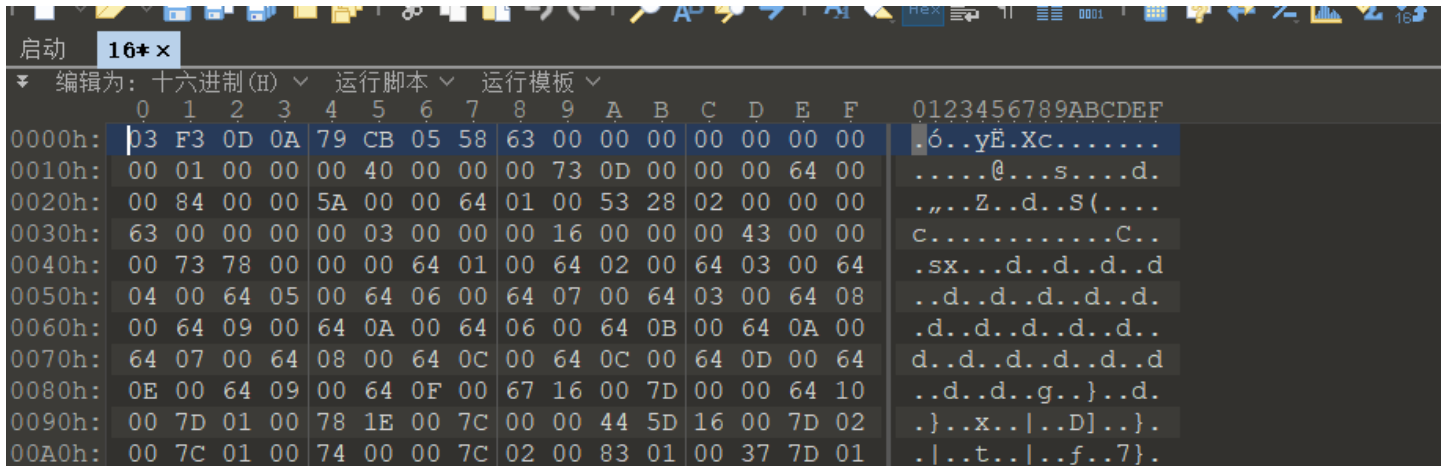
之前有关于图片处理的方法都用了之后没发现什么信息，不敢相信，于是再来一遍，终于发现（之前太过于粗心没注意到）



扫描之后得到一段信息

```
03F30D0A79CB0558630000000000000000100000040000000730D0000006400008400005A00006401005328020000006  
300000000030000001600000043000000737800000064010064020064030064040064050064060064070064030064080064  
0900640A00640600640B00640A00640700640800640C00640C00640D00640E00640900640F006716007D00006410007D  
0100781E007C0000445D16007D02007C01007400007C0200830100377D0100715500577C010047486400005328110000  
04E696600000696C00000069610000006967000000697B000000693300000069380000006935000000693700000069300  
000006932000000693400000069310000006965000000697D00000074000000002801000000740300000063687228030000  
0074030000007374727404000000666C6167740100000069280000000028000000007304000000312E70795203000000010  
00000730A0000000001480106010D0114014E2801000000520300000028000000002800000000280000000073040000003  
12E7079740800000003C6D6F64756C653E010000007300000000
```

感觉像是16进制，想到之前有一道题reverse it，做题的过程中尝试过这种方法，于是我们导入16进制



00B0h:	00 71 55 00	57 7c 01 00	47 48 64 00	00 53 28 11	.qU.W ..GHd..S(. ..Nif...il...ia
00C0h:	00 00 00 4E	69 66 00 00	00 69 6C 00	00 00 69 61	...ig...i{...i3.
00D0h:	00 00 00 69	67 00 00 00	69 7B 00 00	00 69 33 00	..i8...i5...i7..
00E0h:	00 00 69 38	00 00 00 69	35 00 00 00	69 37 00 00	.i0...i2...i4...
00F0h:	00 69 30 00	00 00 69 32	00 00 00 69	34 00 00 00	i1...ie...i}...t
0100h:	69 31 00 00	00 69 65 00	00 00 69 7D	00 00 00 74(....t....ch
0110h:	00 00 00 00	28 01 00 00	00 74 03 00	00 00 63 68	r(....t....strt.
0120h:	72 28 03 00	00 00 74 03	00 00 00 73	74 72 74 04	...flagt....i(..
0130h:	00 00 00 66	6C 61 67 74	01 00 00 00	69 28 00 00	..(....s....l.py
0140h:	00 00 28 00	00 00 00 73	04 00 00 00	31 2E 70 79	R.....s.....
0150h:	52 03 00 00	00 01 00 00	00 73 0A 00	00 00 00 01	H.....N(....R.
0160h:	48 01 06 01	0D 01 14 01	4E 28 01 00	00 00 52 03	... (.... (.... (..
0170h:	00 00 00 28	00 00 00 00	28 00 00 00	00 28 00 00	..s....l.pyt....
0180h:	00 00 73 04	00 00 00 31	2E 70 79 74	08 00 00 00	<module>....s...
0190h:	3C 6D 6F 64	75 6C 65 3E	01 00 00 00	73 00 00 00	.
01A0h:	00				

https://blog.csdn.net/Bok_choy

看见文本后面有一个pyt，猜测是py文件，于是保存为py文件，但是但是.....不会做了
后面看大佬wp，说是要保存为pyc文件并解码（oh我是个`□`脑子）

```
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

def flag():
    str = [
        102,
        108,
        97,
        103,
        123,
        51,
        56
```

```
50,  
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
```

在线反汇编链接<https://tool.lu/pyc/>

运行之后得到

```
flag{38a57032085441e7}  
  
Process finished with exit code 0
```

flag{38a57032085441e7}

心仪的公司

心仪的公司

👍 2 最佳Writeup由admin提供

难度系数: ★★ 2.0

题目来源: 世安杯

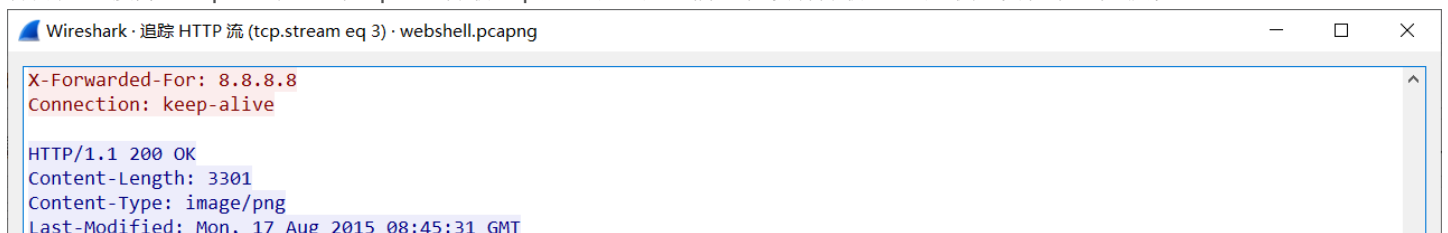
题目描述: 暂无

题目场景: 暂无

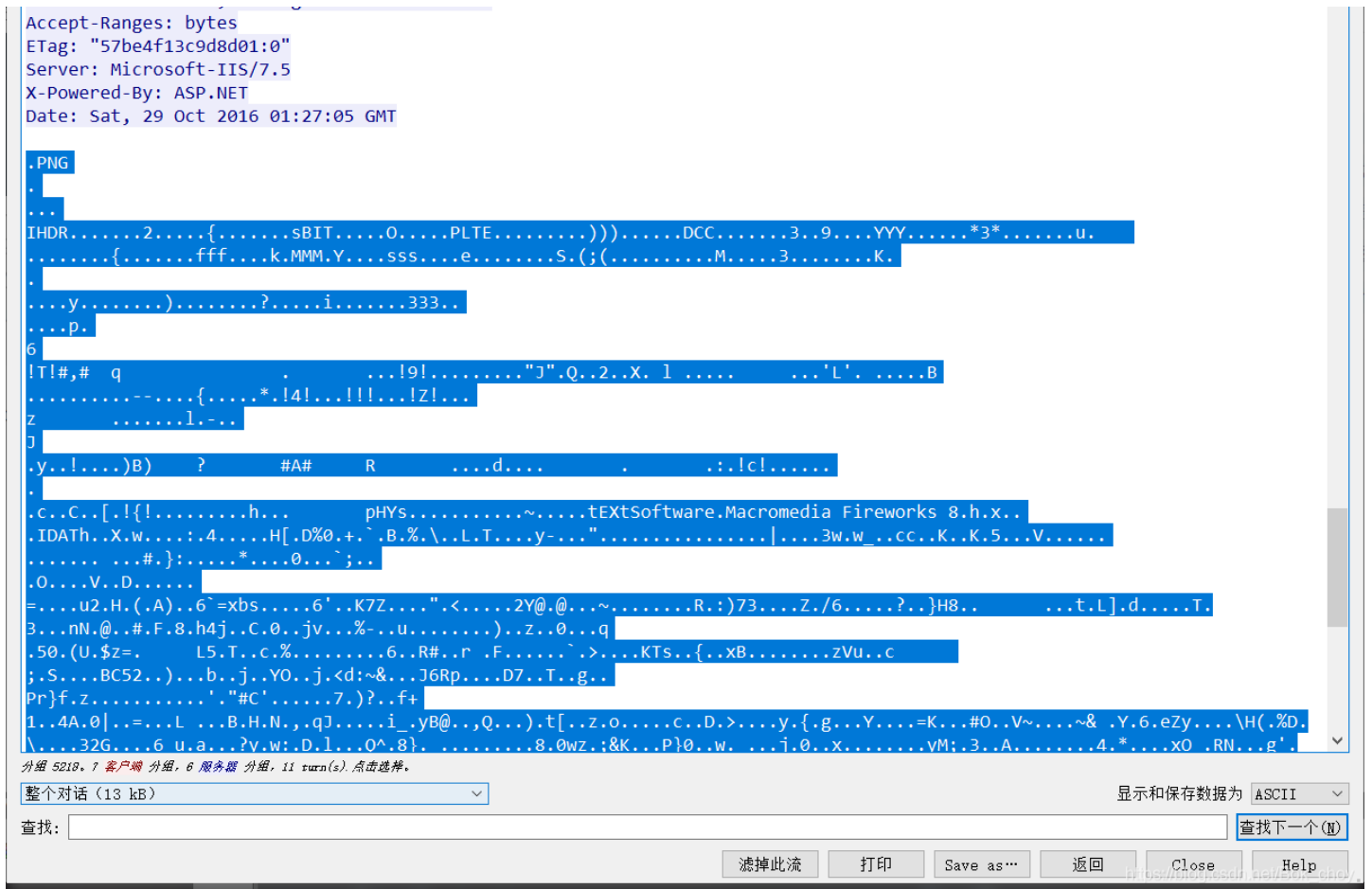
题目附件: 附件1

https://blog.csdn.net/Bok_choy

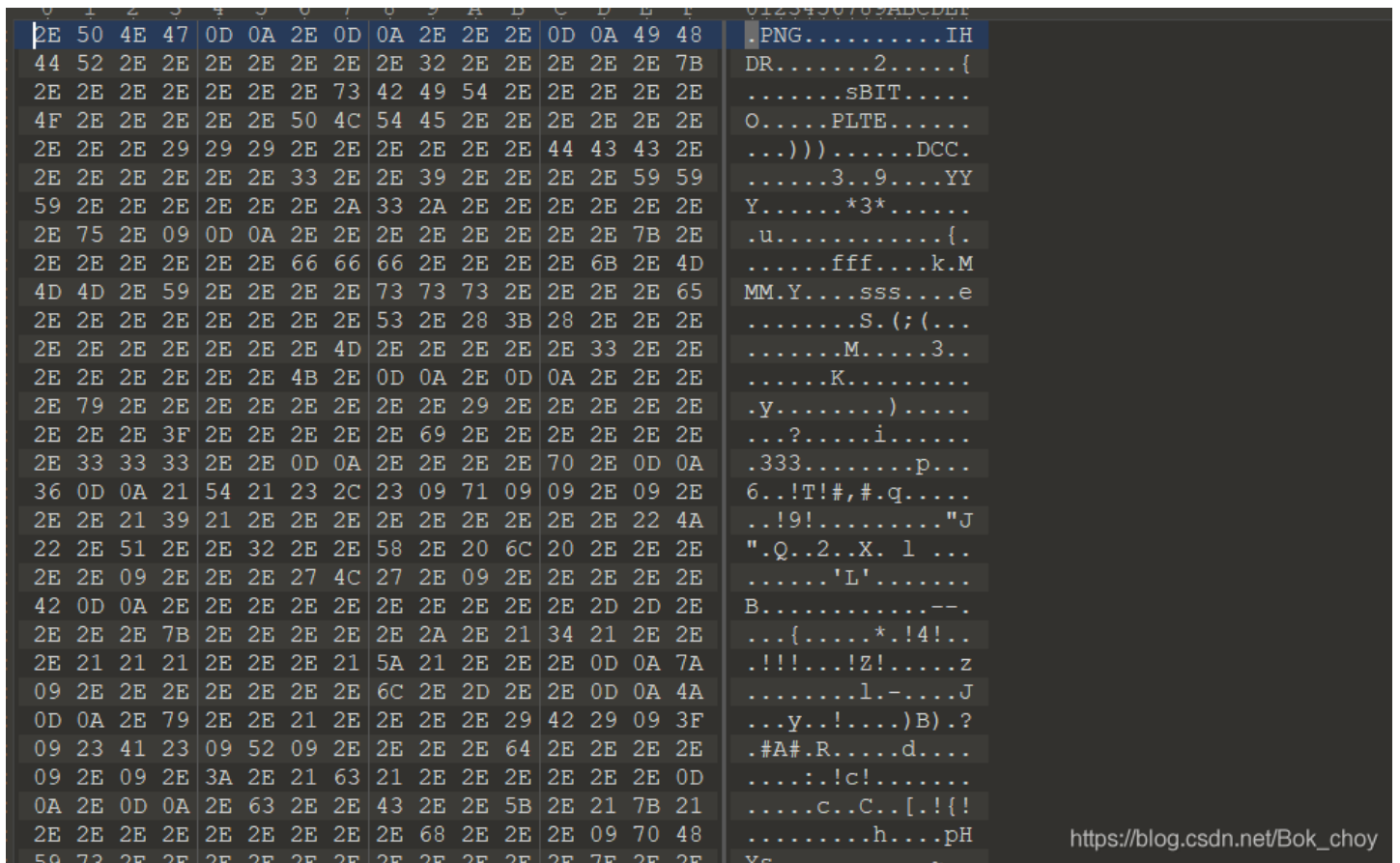
打开后直接筛选http，然后追踪http流，分析http流（永远在乱搞，从没有分析），在最后发现了一大段信息



```
Wireshark · 追踪 HTTP 流 (tcp.stream eq 3) · webshell.pcapng  
X-Forwarded-For: 8.8.8.8  
Connection: keep-alive  
HTTP/1.1 200 OK  
Content-Length: 3301  
Content-Type: image/png  
Last-Modified: Mon, 17 Aug 2015 08:45:31 GMT
```



然后用010打开，应该是png文件，但是文件头和png文件有出入，png文件头是89 50 4E 47,文件尾也是有出入的，png文件尾是AE 42 60 82



尝试修改了下文件头和文件尾，失败了.....弄了好久看了好多资料，都失败了
 可能被自己误导了，重新来一次，这次换追踪TCP流



来康康大佬们的wp

心仪的公司

[目标]

流量分析

[环境]

无

[工具]

wireshark

[分析过程]

ip.addr == 192.168.1.0/24过滤内网ip

192.168.1.111大量访问外网, 猜测是192.168.1.108做反向代理攻击

过滤192.168.1.108追踪tcp流得到flag

```
shell s@. (.Nk.UuUI.D.....o.....c...i.H.m.j...x...t.... ?.]b.....9.;.....n5...fj$.6.r...0.b.....D,..WMa.B....+a<...bh.....&..<~8.hgvq;.N..w.....q;.N.v...L.;.6i..N..Ls.....q;.N.....i....c...fl4g:{ftop_Is_Waiting_4_y}` https://blog.csdn.net/Bok\_choy
```

真好, 我不会~菜的理直气壮555

使用 linux strings 命令查看

对{}进行匹配

```
strings webshell.pcapng | grep {
```

```
function settable(tablename,doing,page) {
    if (doing) {
        if (page) {
function mssqlinfo(dbname) {
!sf{
Qij{
fl4g:{ftop_Is_Waiting_4_y}
!{6S
Je,      {d
root@kali:/mnt/hgfs/shares# strings webshell.pcapng | grep {
```

得到 flag

```
fl4g:{ftop_Is_Waiting_4_y}
```

https://blog.csdn.net/Bok_choy

啊这啊这, 我怎么没想到55555555