

[BUGKU][CTF][MISC][2020] MISC writeup持续更新中

原创

[CryptWinter](#) 于 2020-12-30 22:44:13 发布 2747 收藏 4

分类专栏: [CTF](#) 文章标签: [BUGKU 2020 MISC writeup 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/dadongwudi/article/details/111998874>

版权



[CTF 专栏收录该内容](#)

17 篇文章 2 订阅

订阅专栏

CTF 总结

ctf基本操作: <https://blog.csdn.net/mafucan/article/details/106886421>

zip: <https://www.anquanke.com/post/id/86211>

找软件推荐网址

1. <https://www.52pojie.cn/>

2. 没有的话百度一下 去找百度云

工具下载:

1. stegSolve 隐写分析 需要配置Java环境 <http://www.caesum.com>

MISC 1 签到题目

步骤: 扫码关注

MISC 2 这是一张单纯的图片

关键字: 信息隐藏

步骤:

1. 右键, 用记事本或者是用 .notepad++ 打开拉到最下面, 看到 `key{you are right}`

2. 复制下来HTML实体化转换 得出flag

MISC 3 隐写

关键字: winhex

知识点:

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。

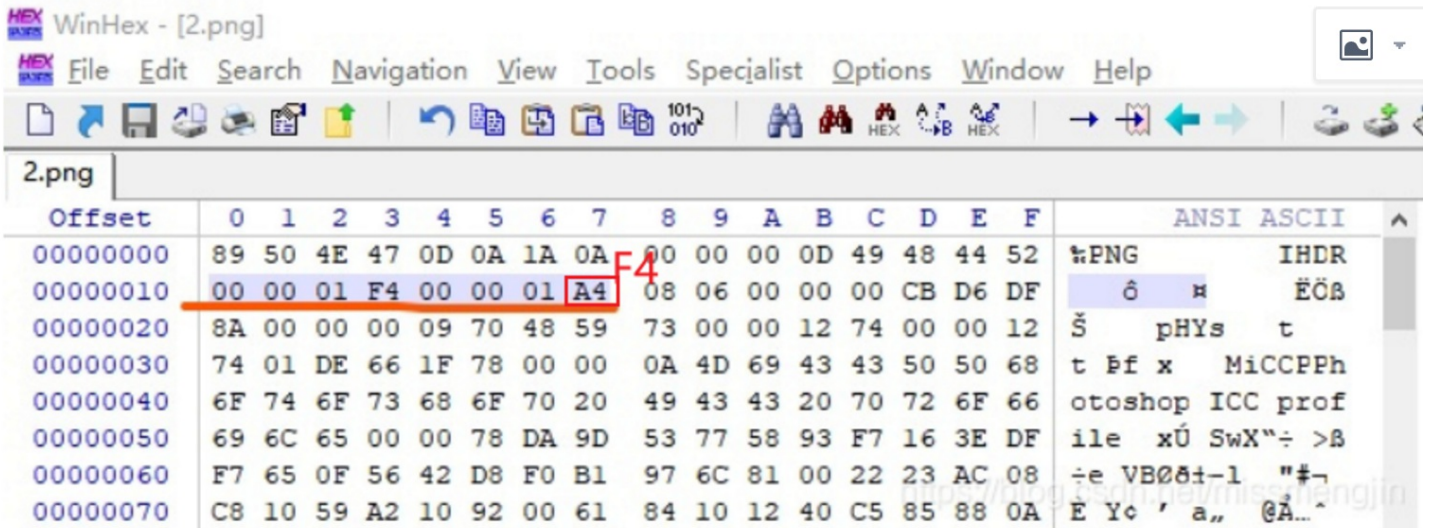
域的名称	字节数	说明
Width	4 bytes	图像宽度, 以像素为单位
Height	4 bytes	图像高度, 以像素为单位
Bit depth	1 byte	图像深度: 索引彩色图像: 1, 2, 4或8 灰度图像: 1, 2, 4, 8或16 真彩色图像: 8或16
ColorType	1 byte	颜色类型: 0: 灰度图像, 1, 2, 4, 8或16 2: 真彩色图像, 8或16 3: 索引彩色图像, 1, 2, 4或8 4: 带 α 通道数据的灰度图像, 8或16 6: 带 α 通道数据的真彩色图像, 8或16
Compression method	1 byte	压缩方法(LZ77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)

<https://blog.csdn.net/qsj/c39629343>

步骤:

1.安装winhex (setup安装在电脑上即可使用) <https://www.52pojie.cn/thread-999432-1-1.html>

2.修改高度 A4->F4,使图片显示完整



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

MISC 4 telnet

关键字: winhex winshark

步骤:

4.1 winhex直接打开

4.2 wireshark打开pac文件 一切都是协议 telnet也不例外 通过telnet与服务器主机进行交互 telnet包可以通过追踪tcp流找到

MISC 5 眼见为实

关键字: word zip

知识点: word文档本质上是个压缩包

步骤: 拿到后解压得到docx文档 打开显示内存被占满 无法打开 拖进010etidor瞅瞅是什么牛鬼蛇神 魔数PK开头 猜测是zip文件 将docx后缀改为zip后解压 之后得到一堆文件 全局搜索一些flag

MISC 6 啊哒

关键字: binwalk hex

步骤:

1.1

1.binwalk 查看, 是文件包含, 该文件包含一个jpg文件和一个zip文件。

2.foremost分离文件, 分离出一个jpg和一个zip文件,

3.打开zip文件, 有flag.txt文件, 解压缩提示需要密码。

4.查看图片详细信息, 相机型号为16进制, 转字符后得到解压缩密码。

5.打开flag.txt, 得到flag。

2.1解压, 用hex打开, 看到flag.txt,改后缀zip, 发现需要解压密码, 查看图片详情16进制转字符为密码, 解压成功得到一个flag.txt文件

MISC 7 又一张图片, 还单纯吗

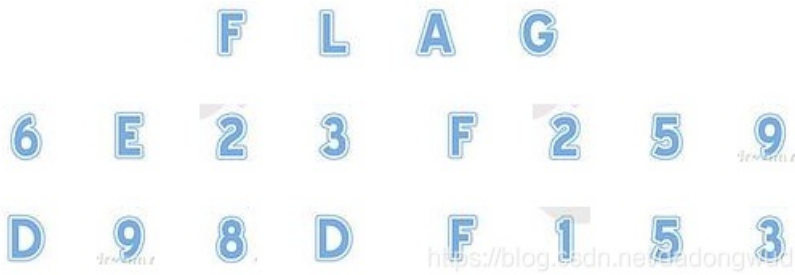
关键字: binwalk foremost

步骤:

1.binwalk file.jpg

2.foremost -T file.jpg

3.重新打开 发现FLAG



SYC{6E23F259D98DF153}

MISC 14 白哥的鸽子

关键字： hex 栅栏密码

知识点： 图片 FFD8开头， FFD9结尾

步骤：

1.末尾有嫌疑 fg2ivyo}{2s3_o@aw__rcl@

2.<https://www.qqxiuzi.cn/bianma/zhalanmima.php>

```
FF D9 66 67 32 69 76 79 6F 7D 6C 7B 32 73 33 5F 6F 40 61 77 ÈbEb%î [Ü¶s äÿÙfg2ivyo}l{2s3_o@aw__rcl@
```

MISC 15 linux

关键字： linux cat grep

知识点：

1.cat（英文全拼：concatenate）命令用于连接文件并打印到标准输出设备上。

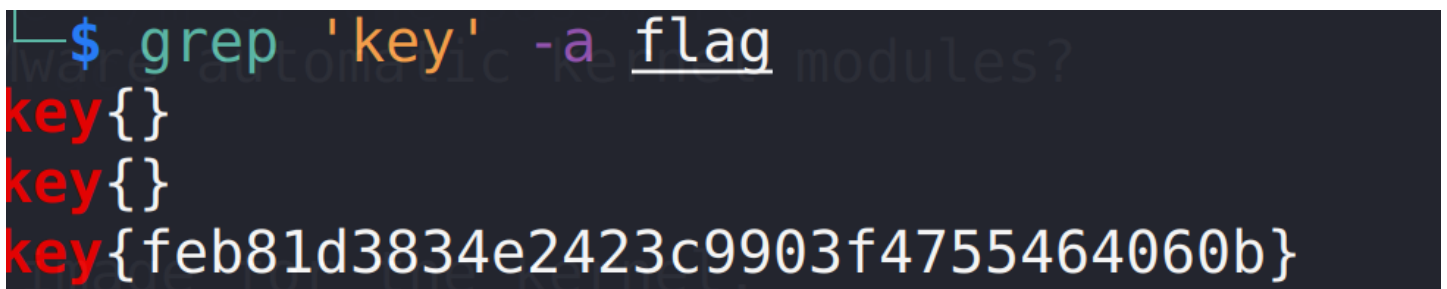
2.Grep 是 Global Regular Expression Print 的缩写，它搜索指定文件的内容，匹配指定的模式，默认情况下输出匹配内容所在的行。注意，grep 只支持匹配而不能替换匹配到的内容。

步骤：

1.解压缩linux.zip,解压缩1.tar.gz,test目录下有一个flag文件。

2.1cat flag，即可见flag。

2.1grep 'key' -a flag



MISC 16 富强民主

关键字： 在线编码

步骤：<http://www.atoolbox.net/Tool.php?id=850>

MISC 17 cisco 为未解决

关键字: cisco

步骤:

- 1.文件2.txt内容为AES加密, 通过解密可知要改1.txt后缀为pka(思科文件后缀)
- 2.用思科打开, 进到交换机CLI的特权执行模式, 密码为flag, 用show running-config命令查看配置即可找到flag

MISC 18 隐写3

关键字: hex 修改高度

步骤:

- 1.大白不完整
- 2.winhex打开 修改高度

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	00	00	02	A7	00	00	01	00	08	06	00	00	00	6D	7C	71
35	00	00	00	01	73	52	47	42	00	AE	CE	1C	E9	00	00	00	04	67	41	4D	41	08	80	B1	8F	0B	FC	61	05	00	00
00	09	70	48	59	73	00	00	0E	C4	00	00	0E	C4	01	95	2B	0E	1B	00	00	FF	A9	49	44	41	54	78	5E	EC	BD	07
A0	A5	57	59	EE	FF	EE	BE	4F	9B	DE	93	4C	7A	0F	84	24	24	60	0C	04	A5	2B	20	45	10	10	BB	88	8A	A8	57
BD	FC	EF	BD	7A	F5	5A	AE	7A	BD	5E	CB	BD	2A	62	05	04	69	52	04	E9	01	42	48	48	42	7A	EF	7D	52	A6	CF

3.出现flag



MISC 19 做个游戏

关键字: jd-gui

步骤:

- 1.解压后发现是个jar, 无法运行
- 2.使用jd-gui (需要下载), 搜索, 找到flag后base 64

The screenshot shows the JD-GUI interface. On the left, a tree view shows the class structure. In the center, the source code of 'PlaneGameFrame.class' is displayed, with a search for 'flag' highlighting a line: `println(g, "flag{RGFqURhbG1fSm1ud2FuQ2hpamk=}");`. A red arrow points from the search results on the right to this line. The search results show '1 matching entry:' and 'cn.bjxt.plane.PlaneGameFrame.class'.

3.base64解码 flag{DajiDali_JinwanChiji}

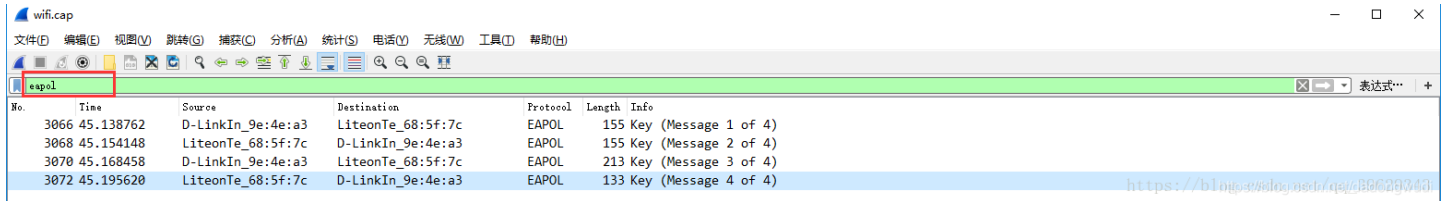
MISC 20 想蹭网先解开密码

关键字:

crunch https://blog.csdn.net/qq_42025840/article/details/81125584

aircrack-ng

知识点: wireshark查找, 发现很多802.11协议, wifi认证的话重点是在WPA的四次握手, 也就是eapol协议的包, 使用规则过滤



The image shows a Wireshark interface with a packet capture named 'wifi.cap'. The filter is set to 'eapol'. The packet list shows four EAPOL Key messages (1, 2, 3, and 4) between source 'D-LinkIn_9e:4e:a3' and destination 'LiteonTe_68:5f:7c'. The packet details pane shows the structure of an EAPOL Key message, including fields like 'Key' and 'MIC'.

No.	Time	Source	Destination	Protocol	Length	Info
3066	45.138762	D-LinkIn_9e:4e:a3	LiteonTe_68:5f:7c	EAPOL	155	Key (Message 1 of 4)
3068	45.154148	LiteonTe_68:5f:7c	D-LinkIn_9e:4e:a3	EAPOL	155	Key (Message 2 of 4)
3070	45.168458	D-LinkIn_9e:4e:a3	LiteonTe_68:5f:7c	EAPOL	213	Key (Message 3 of 4)
3072	45.195620	LiteonTe_68:5f:7c	D-LinkIn_9e:4e:a3	EAPOL	133	Key (Message 4 of 4)

步骤: 环境Linux Kali

1.crunch

```
└─$ crunch 11 11 -t 1391040%% -o dictionary.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
crunch: 100% completed generating output
```

Annotations in the image: '最短' (shortest) points to the first '11', '最长' (longest) points to the second '11', '匹配字段' (match field) points to '-t', '输出' (output) points to '-o', and '文件名' (filename) points to 'dictionary.txt'.

2.aircrack-ng

```
└─$ aircrack-ng wifi.cap -w dictionary.txt
Reading packets, please wait...
Opening wifi.cap
Read 4257 packets.

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

Index number of target network ? 3
```


Reading packets, please wait...

Opening wifi.cap

Read 4257 packets.

1 potential targets

Aircrack-ng 1.6

[00:00:01] 9480/10000 keys tested (6842.83 k/s)

Time left: 0 seconds

94.80%

KEY FOUND! [13910407686]

Master Key : CD 43 57 AD 21 96 7A F4 56 C1 43 DB 56 AF D4 5E
09 37 4F BF 03 CF 58 06 15 62 4E CD D2 68 BB 2D

Transient Key : BF B0 E3 FC 40 0C 81 B6 08 E7 EA AC 04 BA A0 14
F2 1B 50 FD C1 4C AC 69 84 3B 5E 31 48 72 97 95
8F BD F7 FF 50 3A 2D CA 68 50 9F 10 09 93 EC 7D
B4 BE 34 63 76 C6 A8 9B 99 B3 E6 B7 29 08 E1 9D

EAPOL HMAC : DD 75 EA C9 CF F8 13 5B B0 F9 35 FA 8E 97 52 B3

MISC 21 zip伪加密

关键字：zip伪加密

知识点：zip伪加密是在文件头的加密标志位做修改，进而再打开文件时识被别为加密压缩包

一个 ZIP 文件由三个部分组成：

压缩源文件数据区+压缩源文件目录区+压缩源文件目录结束标志

压缩源文件数据区+压缩源文件目录区+压缩源文件目录结束标志

压缩源文件数据区：
50 4B 03 04: 这是头文件标记 (0x04034b50)
14 00: 解压文件所需 pkware 版本
00 00: 全局方式位标记 (有无加密)
08 00: 压缩方式
5A 7E: 最后修改文件时间
F7 46: 最后修改文件日期
16 B5 80 14: CRC-32校验 (1480B516)
19 00 00 00: 压缩后尺寸 (25)
17 00 00 00: 未压缩尺寸 (23)
07 00: 文件名长度
00 00: 扩展记录长度

压缩源文件目录区：
50 4B 01 02: 目录中文件文件头标记(0x02014b50)
3F 00: 压缩使用的 pkware 版本
14 00: 解压文件所需 pkware 版本
00 00: 全局方式位标记 (有无加密, 这个更改这里进行伪加密, 改为09 00打开就会提示有密码了)
08 00: 压缩方式
5A 7E: 最后修改文件时间
F7 46: 最后修改文件日期
16 B5 80 14: CRC-32校验 (1480B516)
19 00 00 00: 压缩后尺寸 (25)
17 00 00 00: 未压缩尺寸 (23)
07 00: 文件名长度
24 00: 扩展字段长度
00 00: 文件注释长度
00 00: 磁盘开始号
00 00: 内部文件属性
20 00 00 00: 外部文件属性
00 00 00 00: 局部头部偏移量

压缩源文件目录结束标志：
50 4B 05 06: 目录结束标记
00 00: 当前磁盘编号
00 00: 目录区开始磁盘编号
01 00: 本磁盘上纪录总数
01 00: 目录区中纪录总数
59 00 00 00: 目录区尺寸大小
3E 00 00 00: 目录区对第一张磁盘的偏移量
00 00: ZIP 文件注释长度

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

步骤：

1.压缩文件结构

1.1 查看文件，压缩源文件数据区+压缩源文件目录区+压缩源文件目录结束标志

```
50 4B 03 04 14 00 09 00 08 00 50 A3 A5 4A 21 38 76 65 19 00 00 00 17 00 00 00 08 00 00 00 66 6C  
61 67 2E 74 78 74 4B CB 49 4C AF 76 4C C9 35 F4 D3 75 32 72 D7 CD 0E D5 0D 8E F2 0C A8 05 00 50  
4B 01 02 1F 00 14 00 09 00 08 00 50 A3 A5 4A 21 38 76 65 19 00 00 00 17 00 00 00 08 00 24 00 00  
00 00 00 00 00 20 00 00 00 00 00 66 6C 61 67 2E 74 78 74 0A 00 20 00 00 00 00 01 00 18  
00 0F F5 04 D5 9A C5 D2 01 46 1F CB 8A 9A C5 D2 01 46 1F CB 8A 9A C5 D2 01 50 4B 05 06 00 00 00  
00 01 00 01 00 5A 00 00 00 3F 00 00 00 00 00
```

1.2 红色框住的50

4B 是压缩源文件数据区的头文件标记，它对应的红色框柱的 09 00 并不影响加密属性。

绿色框住的50 4B 是压缩源文件目录区，它对应的绿色框柱的 09 00 影响加密属性，当数字为奇数是为加密，为偶数时不加

2 ZipCenOp.jar

2.1 下载flag.zip ZipCenOp.jar文件 放在同一个文件夹

下载地址: <https://pan.baidu.com/s/1GHcUYA36X9reZL7rcmWNfA> 提取码: uqyn

2.2 java -jar ZipCenOp.jar r flag.zip , 运行后均能不需要输入密码就能打开文件

正常

```
C:\ctf>java -jar ZipCenOp.jar r flag.zip
success 1 flag(s) found
C:\ctf>
```

我的

```
D:\CTF\MISC>java -jar ZipCenOp.jar r flag.zip
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by zip.CenOp$1 (rsrc:./) to method java.nio.DirectByteBuffer.cleaner()
WARNING: Please consider reporting this to the maintainers of zip.CenOp$1
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
Exception in thread "main" java.lang.reflect.InvocationTargetException
    at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.base/java.lang.reflect.Method.invoke(Method.java:567)
    at org.eclipse.jdt.internal.jarinjarloader.JarRsrcLoader.main(JarRsrcLoader.java:58)
Caused by: java.lang.NoClassDefFoundError: sun/misc/Cleaner
    at zip.CenOp$1.run(CenOp.java:95)
    at java.base/java.security.AccessController.doPrivileged(AccessController.java:310)
    at zip.CenOp.clean(CenOp.java:89)
    at zip.CenOp.operate(CenOp.java:80)
    at zip.CenOp.main(CenOp.java:32)
    ... 5 more
Caused by: java.lang.ClassNotFoundException: sun.misc.Cleaner
    at java.base/java.net.URLClassLoader.findClass(URLClassLoader.java:436)
    at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:588)
    at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:521)
    ... 10 more
```

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

参考链接: <https://www.pianshen.com/article/3780175978/>

MISC 23 细心的大象

关键字: binwalk foremost

知识点: dd命令

<https://www.cnblogs.com/ginvip/p/6370836.html>

步骤: windows+kali

1.图片按照以往经验 binwalk查看

2.用dd命令 或用foremost分解出来得到一个大象的图片及一个压缩包（有密码）

```
(d@kali) - [~/mnt/hgfs/share]
$ binwalk 1.jpg
DECIMAL      HEXADECIMAL     DESCRIPTION
-----
0            0x0             JPEG image data, EXIF standard
12           0xC             TIFF image data, big-endian, offset of first image directory: 8
5005118     0x4C5F3E       PARity archive data
6391983     0x6188AF       RAR archive data, version 4.x, first volume type: MAIN_HEAD

(d@kali) - [~/mnt/hgfs/share]
$ dd if=1.jpg of=1.rar skip=6391983 bs=1
记录了16301+0 的读入
记录了16301+0 的写出
16301字节 ( 16 kB , 16 KiB) 已复制 , 2.5484 s , 6.4 kB/s

(d@kali) - [~/mnt/hgfs/share]
$ foremost -T 1.jpg
Processing: 1.jpg
|*|
```

3.查看大象图片的详情，有个备注（经过base64加密的，先解码得到那个压缩包的密码）

base64编码

base16、base32、base64

TVNEUzQ1NkFTRDEyM3p6

编码 base64

MSDS456ASD123zz

1.jpg 属性

属性	值
说明	
标题	出题人已经跑路了
主题	出题人已经跑路了
分级	☆☆☆☆☆
标记	
备注	TVNEUzQ1NkFTRDEyM3p6
来源	
作者	Bugku
拍摄日期	2017/8/10 11:53
程序名称	sagit-user 7.1.1 NMF26X V8.2.26.0.NCACNE...
获取日期	
版权	
图像	
图像 ID	
分辨率	3016 x 4032
宽度	3016 像素
高度	4032 像素
水平分辨率	72 dpi
垂直分辨率	72 dpi
位深度	24
压缩	

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

4.解压之后得到一张图片，用winhex打开，修改宽和高的值一样，然后保存。再打开图片即可在图片上看到flag.



BUGKU{a1e5a5A}

MISC 24 爆照

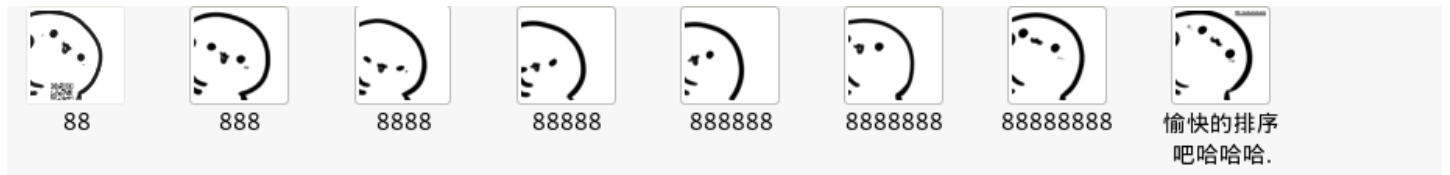
关键字: binwalk foremost

知识点:

步骤: windows+kali

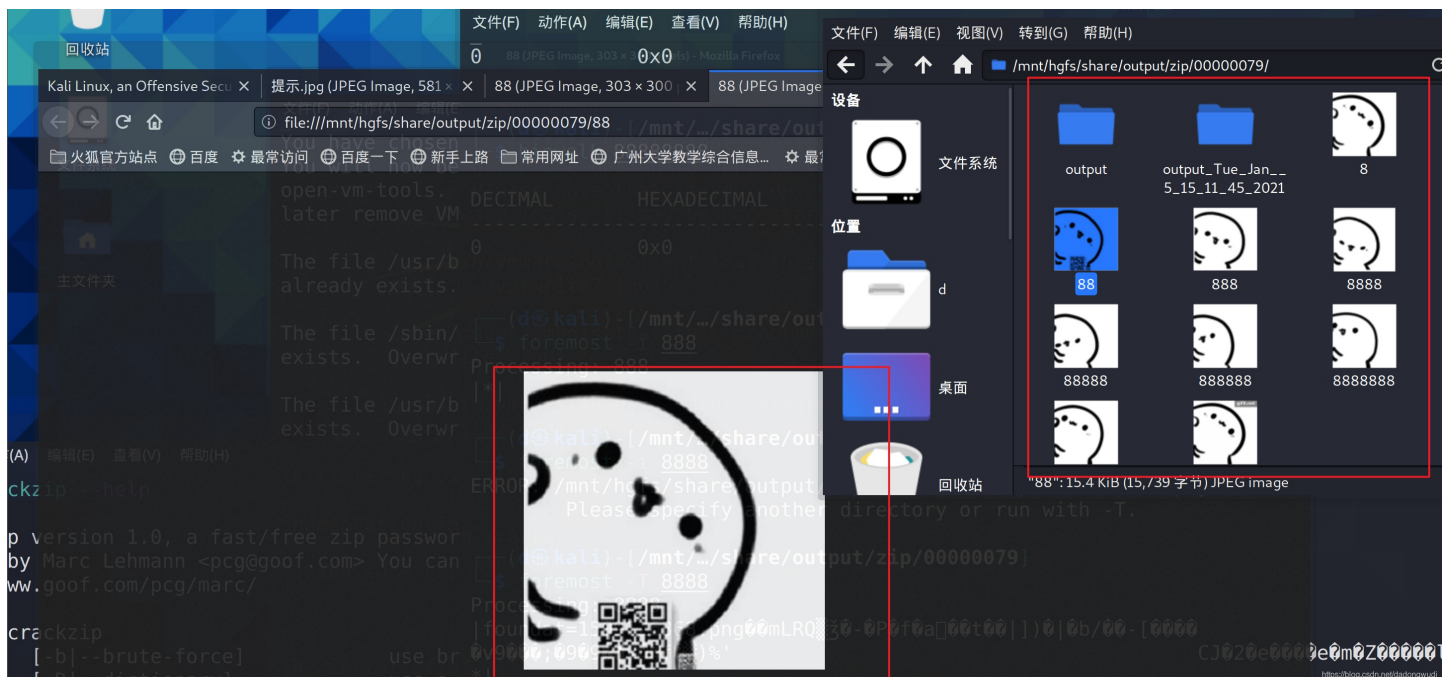
1.binwalk分析原文件， foremost分离，解压后发现多个文件

2.在win下不知道是什么文件 但是在linux下系统会自动辨别 所以我们复制到kali里面



3.binwalk逐个分析文件 发现前三个有修改的痕迹， 88， 888， 8888

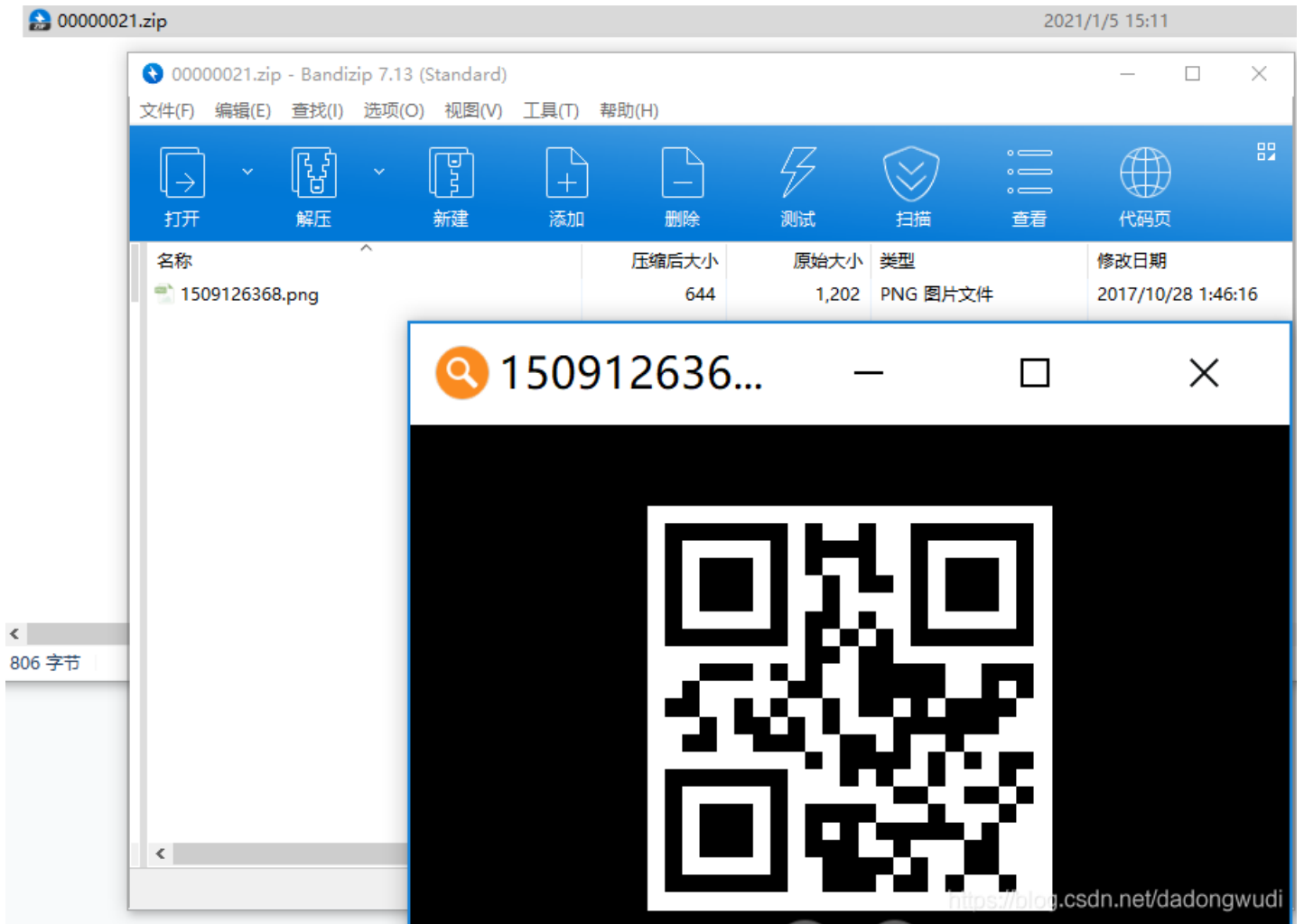
88



base编码

base16、base32、base64

The diagram illustrates the process of base64 encoding. It starts with the text `c2lsaXNpbGk=` in a box. A red arrow points down to a '编码' (Encode) button with a dropdown menu set to 'base64'. Another red arrow points down to a box containing the decoded text `silisili`. To the right, a screenshot of a file's properties dialog for `00000000.jpg` shows the '备注' (Comments) field containing `c2lsaXNpbGk=`.



MISC 25 猫片

关键字: LSB BGR ntfstreamseditor StegSolve

知识点:

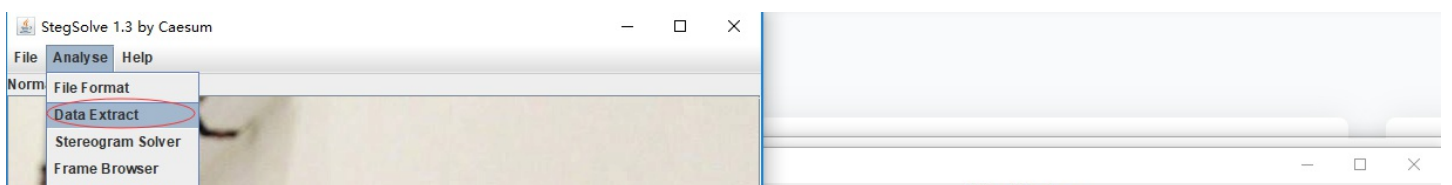
1. chr() 用一个范围在 range (256) 内的 (就是0~255) 整数作参数, 返回一个对应的字符。
2. ord() 函数是 chr() 函数 (对于8位的ASCII字符串) 或 unichr() 函数 (对于Unicode对象) 的配对函数, 它以一个字符 (长度为1的字符串) 作为参数, 返回对应的 ASCII 数值, 或者 Unicode 数值, 如果所给的 Unicode 字符超出了你的 Python 定义范围, 则会引发一个 TypeError 的异常
3. uncompile 反编译 pyc
pip install uncompile
uncompile6 文件名

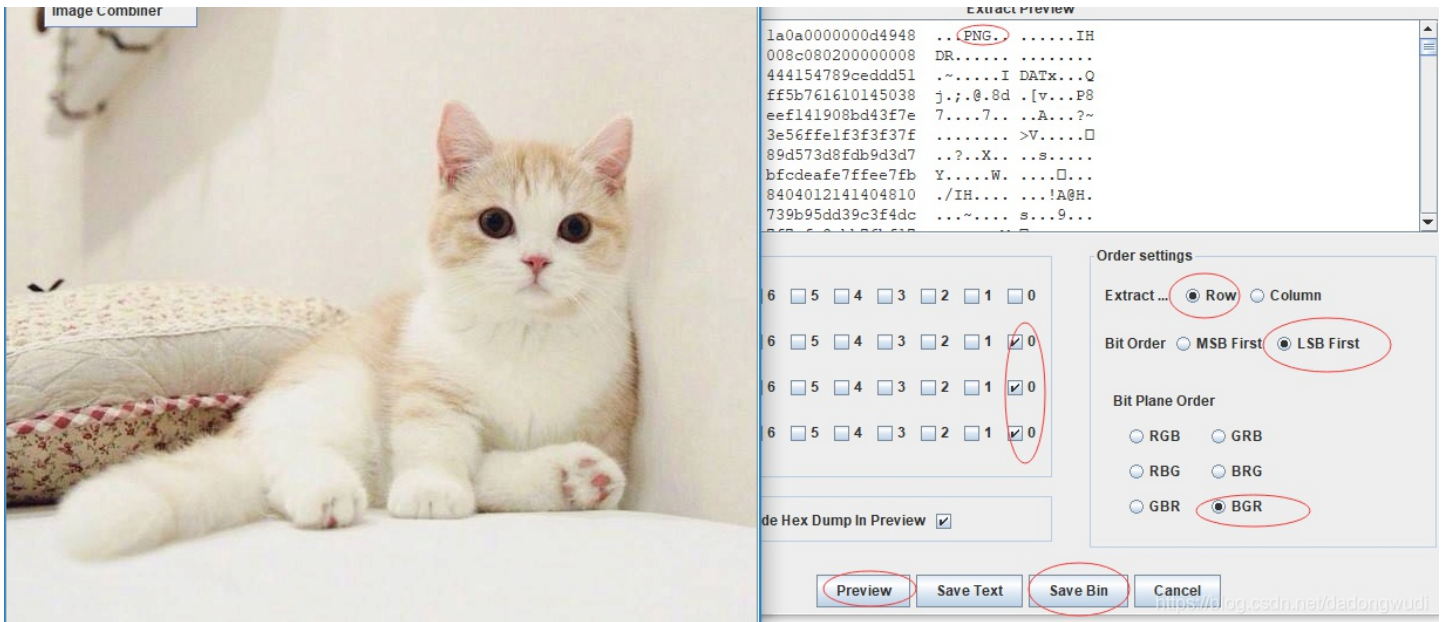
步骤: windows+java+python

1. hint: LSB BGR 想到信息隐藏, 使用 StegSolve 提取

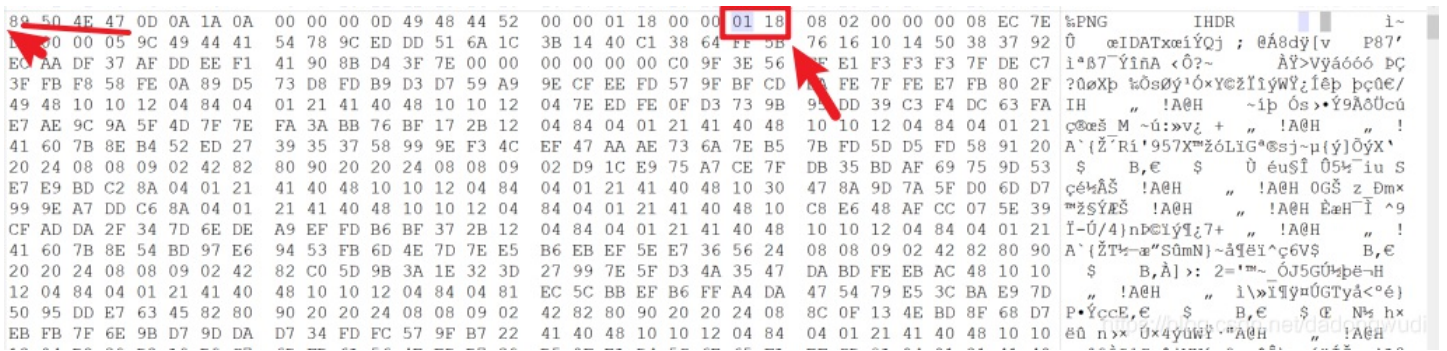
Alpha 为什么不勾, 因为是透明度, 如果改为 0 就变成全透明, 图片就什么也看不见, RGB 是全为 0, 让它避免有其他颜色亮度的干扰, 如果 RGB 任意一个改为不为 0, 那么这张图片的代码就会出现错误, 导致他不是一个图片文件。

当我们改好的时候 可以看见是一个 png 文件

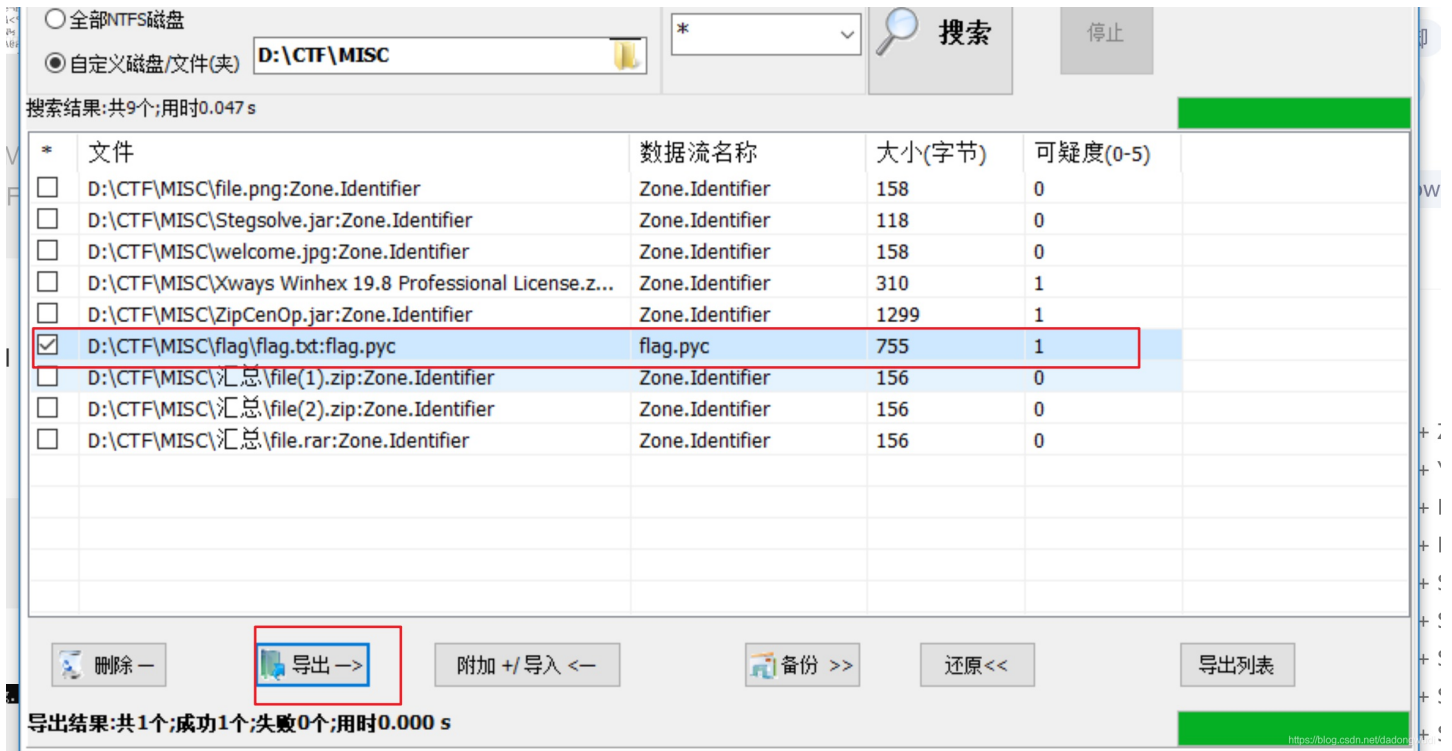




2.打不开图片。winhex打开，选中后右键删除头ffe，保存，发现需要修改高度，发现是一个二维码，扫描是一个百度云链接



3.NTFS提示要用ntfstreamsedtiior 一定要用winrar并且用ntsf 导出pyc文件
 ntfstreamseditior 下载链接: <https://bbs.kafan.cn/thread-460084-1-1.html>
 winrar 下载链接: <https://www.52pojie.cn/>



4.pyc编译成py

4.1 uncompile

```
(base) PS D:\CTF\MISC> pip install uncompile
```

```
(base) PS D:\CTF\MISC> uncompile6 .\D:\CTF\MISC\flag\flag.txt!flag.pyc >flag.py
```

4.2在线

<https://tool.lu/pyc/>

```
import base64

def encode():
    flag = '*****'
    ciphertext = []
    for i in range(len(flag)):
        s = chr(i ^ ord(flag[i]))
        if i % 2 == 0:
            s = ord(s) + 10
        else:
            s = ord(s) - 10
        ciphertext.append(str(s))

    return ciphertext[::-1]

ciphertext = [
    '96',
    '65',
    '93',
    '123',
    '91',
    '97',
    '22',
    '93',
    '70',
    '102',
    '94',
    '132',
    '46',
    '112',
    '64',
    '97',
    '88',
    '80',
    '82',
    '137',
    '90',
    '109',
    '99',
    '112']
```

5.根据加密过程写脚本并运行得到flag

```

def decode():
    ciphertext = [
        '96',
        '65',
        '93',
        '123',
        '91',
        '97',
        '22',
        '93',
        '70',
        '102',
        '94',
        '132',
        '46',
        '112',
        '64',
        '97',
        '88',
        '80',
        '82',
        '137',
        '90',
        '109',
        '99',
        '112']
    ciphertext.reverse()
    flag = ''
    for i in range(len(ciphertext)):
        if i % 2 == 0:
            s = int(ciphertext[i]) - 10
        else:
            s = int(ciphertext[i]) + 10
        s=chr(i^s)
        flag += s
    return flag

def main():
    flag = decode()
    print(flag)

if __name__ == '__main__':
    main()

```

5.1在线运行

<https://c.runoob.com/compile/6>

5.2

```

(base) PS D:\CTF\MISC> python 4.py
flag {Y@e_C13veR_C1Ever!}

```

参考链接: <https://www.cnblogs.com/Paranoid-4/p/9502196.html>

MISC 26 多彩 超出能力范围

关键字: YSL

步骤:

参考链接: <https://blog.csdn.net/carlzi/article/details/82753478>

参考链接: <https://www.secpulse.com/archives/69465.html>


```

f = open('flag.txt')
temp = []
while True:
    k = f.read(3)
    if k:
        temp.append(k)
    else:
        break

f.close()
for i in temp:
    num = '00' + i
    num = int(num, base=0)
    num = chr(num)
    print(num, end='')

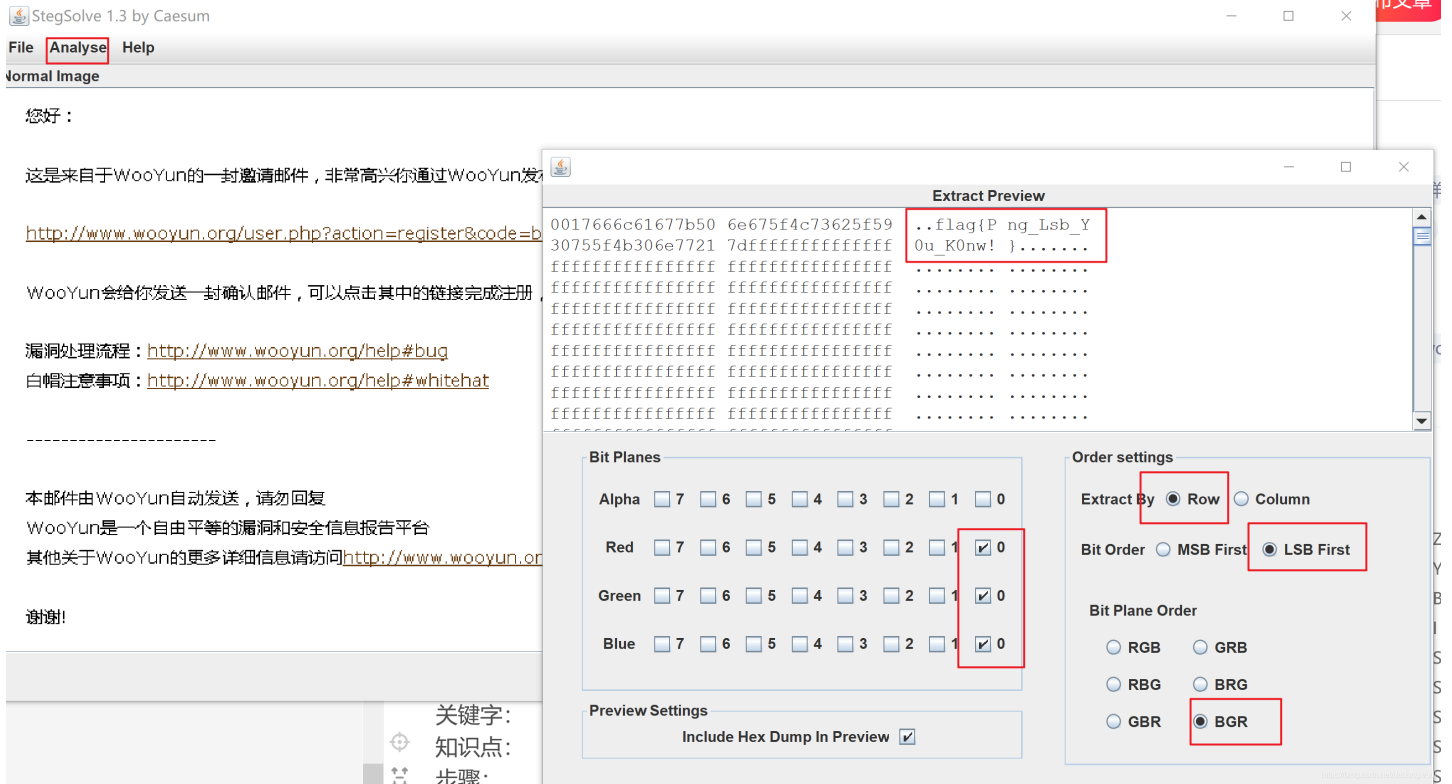
```

MISC 29 乌云的邀请码

关键字: StegSolve 隐写

步骤: windows

- 1.打开 StegSolve java -jar Stegsolve.jar
- 2.提取图片里面的信息



MISC 30 神秘的文件

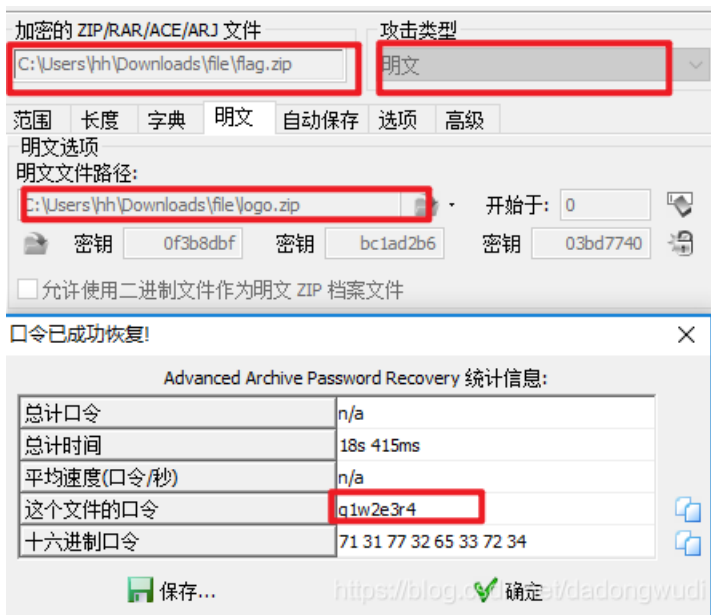
关键字：明文攻击 ARCHPR

知识点：

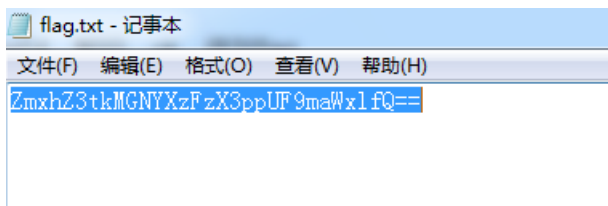
步骤：windows+Kali

1.压缩包和png图片，压缩包是加密的，并且那个图片存在于压缩包，于是将logo.png使用winRAR加密成zip，然后对比他们的CRC32,发现一模一样！考点就是明文攻击。

明文攻击（不懂的参考<http://www.cnblogs.com/ECJTUACM-873284962/p/9387711.html>）



2.binwalk查看，foremos分离后，找到flag.txt，打开得到一串base64，解码，ok，得到flag



MISC 论剑

关键字: winhex binwalk

知识点:

1.base64, base32, base16

快速判断方法: 大写小写都有是base64,只有大写是base32,只有大写且最大只到F是base16

2.常见的文件头

7z

文件头标识: 37 7A BC AF 27 1C

JPEG/JPG

文件头标识: ff, d8 (SOI) (JPEG 文件标识)

文件结束标识: ff, d9 (EOI)

PNG

文件头标识: 89 50 4E 47 0D 0A 1A 0A

GIF

文件头标识: 47 49 46 38 39(37) 61 GIF89(7)a

BMP

文件头标识: 42 4D--- BM

HTML (html)

文件头标识: 68746D6C3E

ZIP Archive (zip)

文件头标识: 504B0304 PK

RAR Archive (rar)

文件头标识: 52617221

3.jpeg格式详解https://blog.csdn.net/yun_hen/article/details/78135122

步骤: win+kali

1.winhex到中部发现一串二进制 进行进制转换, 发现类似7z文件头37 7A BC AF, 修改保存

这个图片中隐藏的不是常见的zip压缩包, 是7z压缩包

位置就在二进制码的旁边

008{

标蓝的位置是7z的一个标志, 但是这里文件头损坏了, 所以binwalk没有分析出来

正常的7z头: 7z

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

进制转换

ASCII与2进制、10进制和16进制之间相互转; 2进制、8进制、10进制、16进制及任意进制相互转换

进制转换 (常用) 进制转换 (任意)

ASCII = 进制

文本: mynameiskey!!!hhh

二进制: 01101101 01110001 01101110 01100001 01101101 01100101 01101001 01110011 01101011 01100101 01111001 00100001 00100001 00100001 01101000 01100000 01101000

清空

file.jpg
D:\share
文件大小: 21.5 KB
21,990 字节
状态: 原始的
撤销级数: 0
反向撤销: 暂无信息
创建时间: 2021/01/07 16:28:13
最后写入时间: 2021/01/07 16:27:46
属性: A
图标: 0
模式: 文本
偏移地址: 十六进制
每页字节数: 47x32=1504
当前窗口: 11
窗口名称: 11

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

2.foremost无法分离出7z, 用dd分离

```
(kali) - [~/mnt/hgfs/share]
$ binwalk file.jpg
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	JPEG image data, JFIF standard 1.01
9591	0x2577	7-zip archive data, version 0.4
17569	0x44A1	JPEG image data, JFIF standard 1.01

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

```
(kali) - [~/mnt/hgfs/share]
$ dd if=file.jpg of=7.zip skip=9591 bs=1
^[[A记录了12399+0 的读入
记录了12399+0 的写出
12399字节 ( 12 kB , 12 KiB) 已复制 , 2.24452 s , 5.5 kB/s
find more info on
(kali) - [~/mnt/hgfs/share]
$ dd if=file.jpg of=1.jpg skip=17569 bs=1
记录了4421+0 的读入
记录了4421+0 的写出
4421字节 ( 4.4 kB , 4.3 KiB) 已复制 , 0.933102 s , 4.7 kB/s
```

3.用之前的密码修改解密压缩文件 得到一张图片



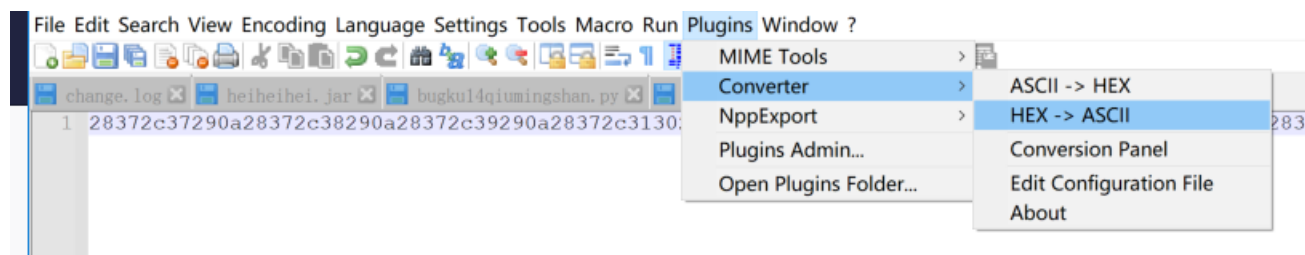
MISC 图穷匕见

关键字: winhex gnuplot

知识点: jpg的文件尾FF D9

步骤:

1.winhex查看, jpg的文件尾FF D9, 将之后的数据保存到txt中, 使用的是notepad++中的插件Converter进行解码



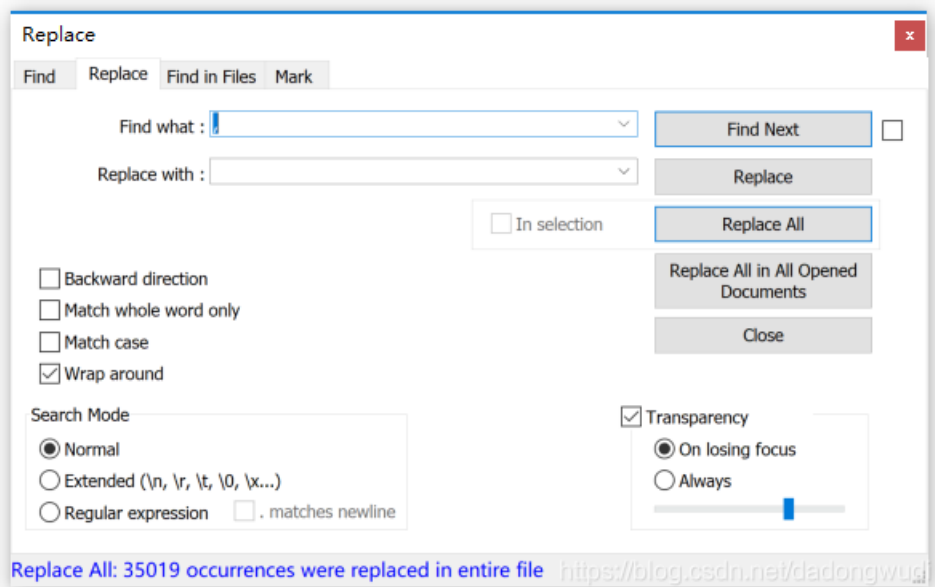
2.去除(",)" 匹配gnuplot格式

2.1 直接replace

2.2 python脚本

```
with open('paintpaintpaint.txt','r') as file:
    fw = open('f1.txt','w')
    while 1:
        lines = file.readlines()
        if not lines:
            break
        for line in lines:
            fw.write(line.replace('(',')').replace(')',',').replace(',',' '))
file.close
fw.close
```

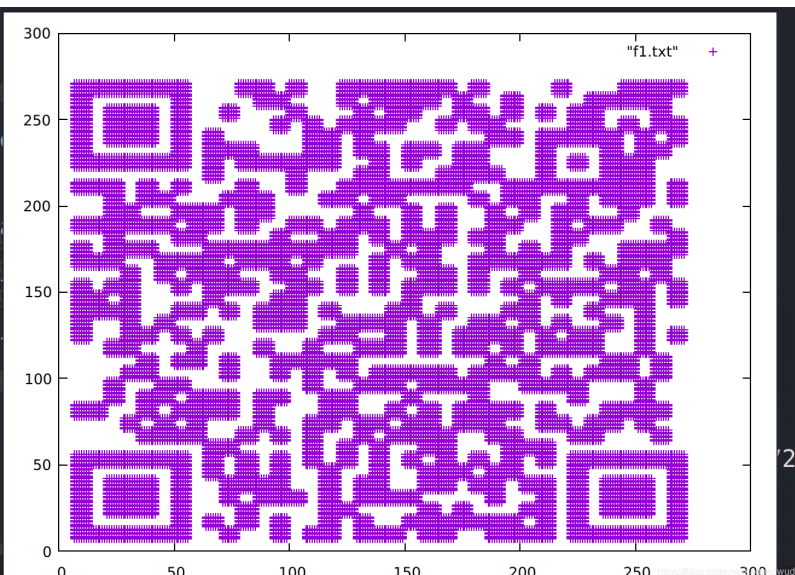
271 174
271 175
271 176
271 177
271 178
271 179
271 180
271 181
271 182
271 183
271 184
271 185
271 201
271 202
271 203
271 204
271 205
271 206
271 207
271 208
271 209
271 210
271 211
271 212
271 213
271 214



3.安装

gnuplot 使用gnuplot, 扫码得到flag
sudo apt-get install gnuplot

```
(kali) - [~/mnt/hgfs/share]
$ gnuplot
GNUPLLOT
Version 5.4 patchlevel 1 last mo
Copyright (C) 1986-1993, 1998, 2004
Thomas Williams, Colin Kelley and m
gnuplot home: http://www.gnuplo
faq, bugs, etc: type "help FAQ"
immediate help: type "help" (plo
Terminal type is now 'qt'
gnuplot> plot "f1.txt"
gnuplot> qt.qpa.xcb: QXcbConnection: XCB er
, minor code: 0
```



MISC convert

关键字: convert

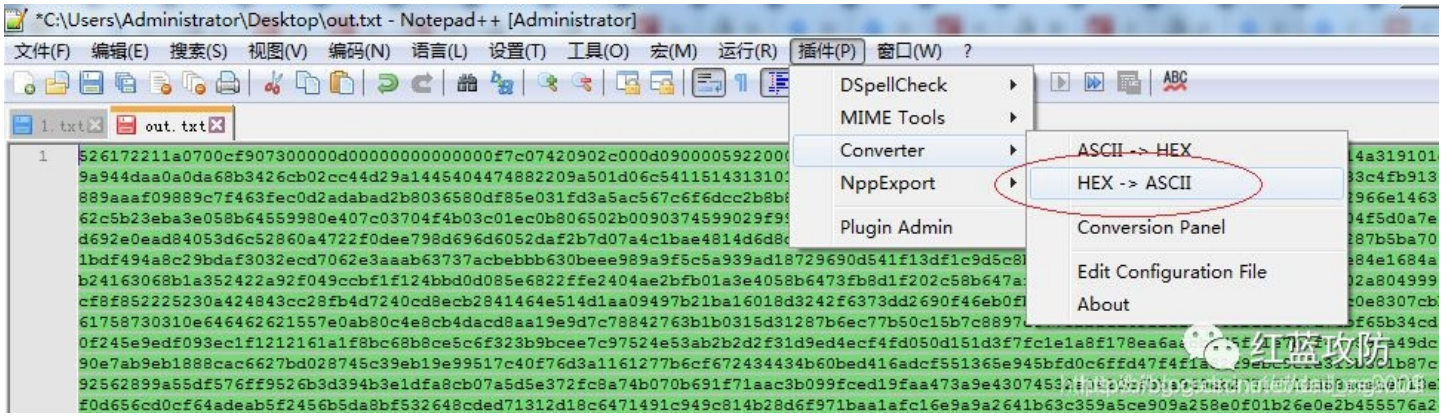
知识点:

步骤: win anaconda

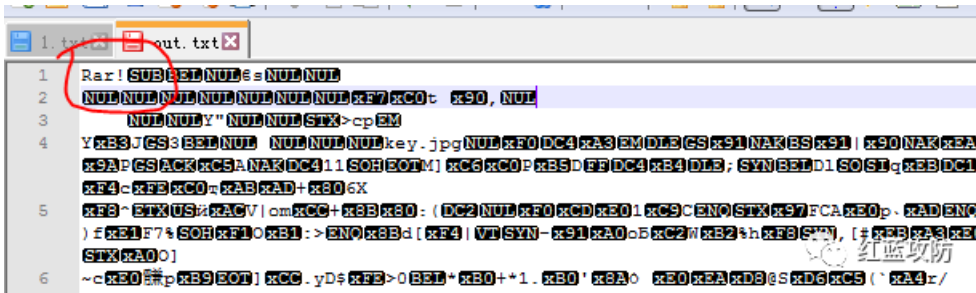
1.脚本2进制分组转换成16进制


```
#1
f1 = open('1.txt', 'r')
fw = open('2.txt', 'w')
fw.write(hex(int(str(f1.read()), 2)))
f1.close()
fw.close()
#2
f1=open('1.txt', 'r')
#二进制转10进制
oct1=int(f1.read(), 2)
#十进制转16进制
hex1=hex(oct1)
#将十六进制文件写入文件
f2=open('out.txt', 'w')
f2.write(hex1)
#关闭文件
f1.close()
f2.close()
```

2. 转成16进制之后，删除开头的0x，然后在进行十六进制转ASCII码操作(可以直接用notepad++的插件)



3. 转成ASCII码之后，看到开头Rar!，很明显的rar文件头，修改文件后缀为.rar



4. 解压后得到图片key.jpg，属性查看

base编码

base16, base32, base64

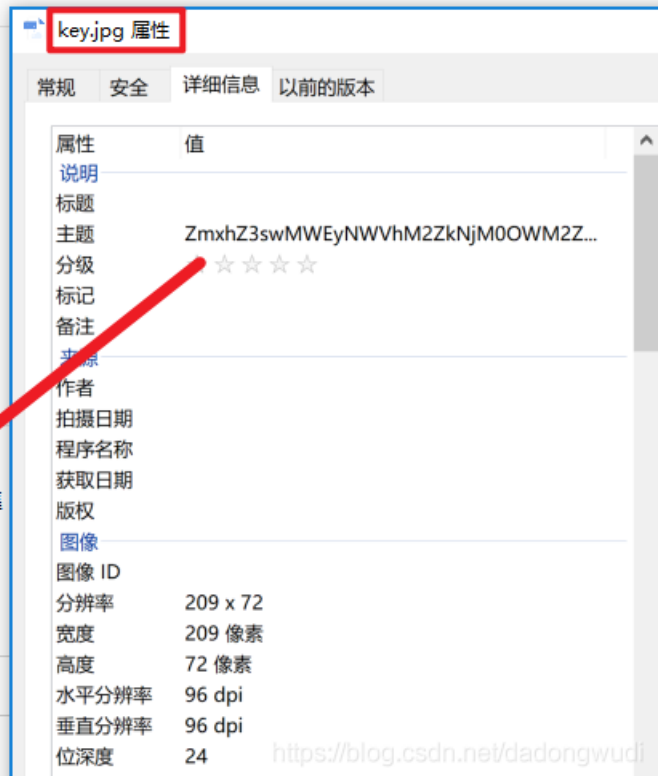
ZmxhZ3swMWEyNWVhM2ZkNjM0OVM2ZTYzNWExZDExOTZlNzVmYn0=

编码

base64

字符集

flag{01a25ea3fd6349c6e635a1d0196e75fb}



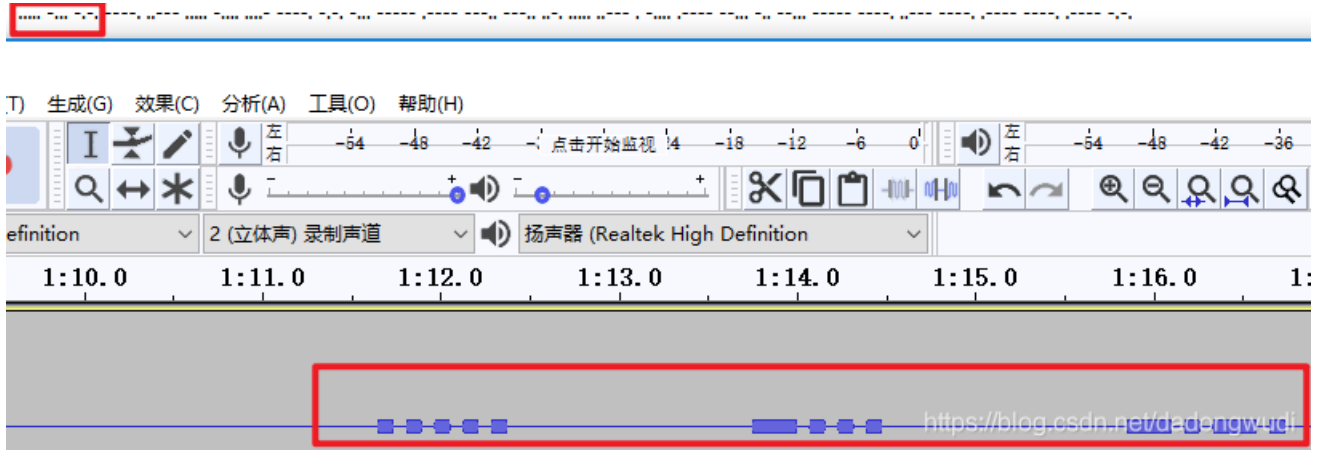
<https://blog.csdn.net/dadongwudi>

MISC 听首音乐

关键字: audacity 摩斯密码

步骤:

1. 下载, 解析



2. 认定为摩斯密码:

在线转换一下就是5BC925649CB0188F52E617D70929191C

MISC 好多数值

关键字: PIL RGB

步骤:

1. 看到的第一反映就是RGB值, 需要通过这些值来重构图片, 但是仅有像素的RGB值是不行的, 我们还必须要知道此图像的尺寸, 即每行每列各有多少像素

粗略的研究一下这些数据就能够发现, 在其中的某一段呈现出有规律型的变化

计算出行列 $3219-3098+1=122$ $61366/122=502$

3094	255, 255, 255	3216	255, 255, 255
3095	255, 255, 255	3217	255, 255, 255
3096	255, 255, 255	3218	255, 255, 255
3097	<u>255, 255, 255</u>	<u>3219</u>	<u>255, 255, 255</u>
3098	<u>255, 255, 253</u>	3220	255, 255, 248
3099	255, 253, 251	3221	255, 254, 235
3100	255, 249, 235	3222	255, 249, 211

尾

61365	255, 255, 255
61366	<u>255, 255, 255</u>
61367	

2. 重构图片, 可使用python中的PIL中的Image库 (不要命名为PIL.py, 否则cannot import name 'Image' from 'PIL')

```

from PIL import Image

x = 502
y = 122

c = Image.new("RGB", (x,y))
file_object = open('1.txt', 'r')

for i in range(0,x):
    for j in range(0,y):
        text = file_object.readline()
        textarr = text.split(',')
        print(textarr)
        c.putpixel([i,j],(int(textarr[0]),int(textarr[1]),int(textarr[2])))

c.show()

```

flag(youc@n'tseeme)

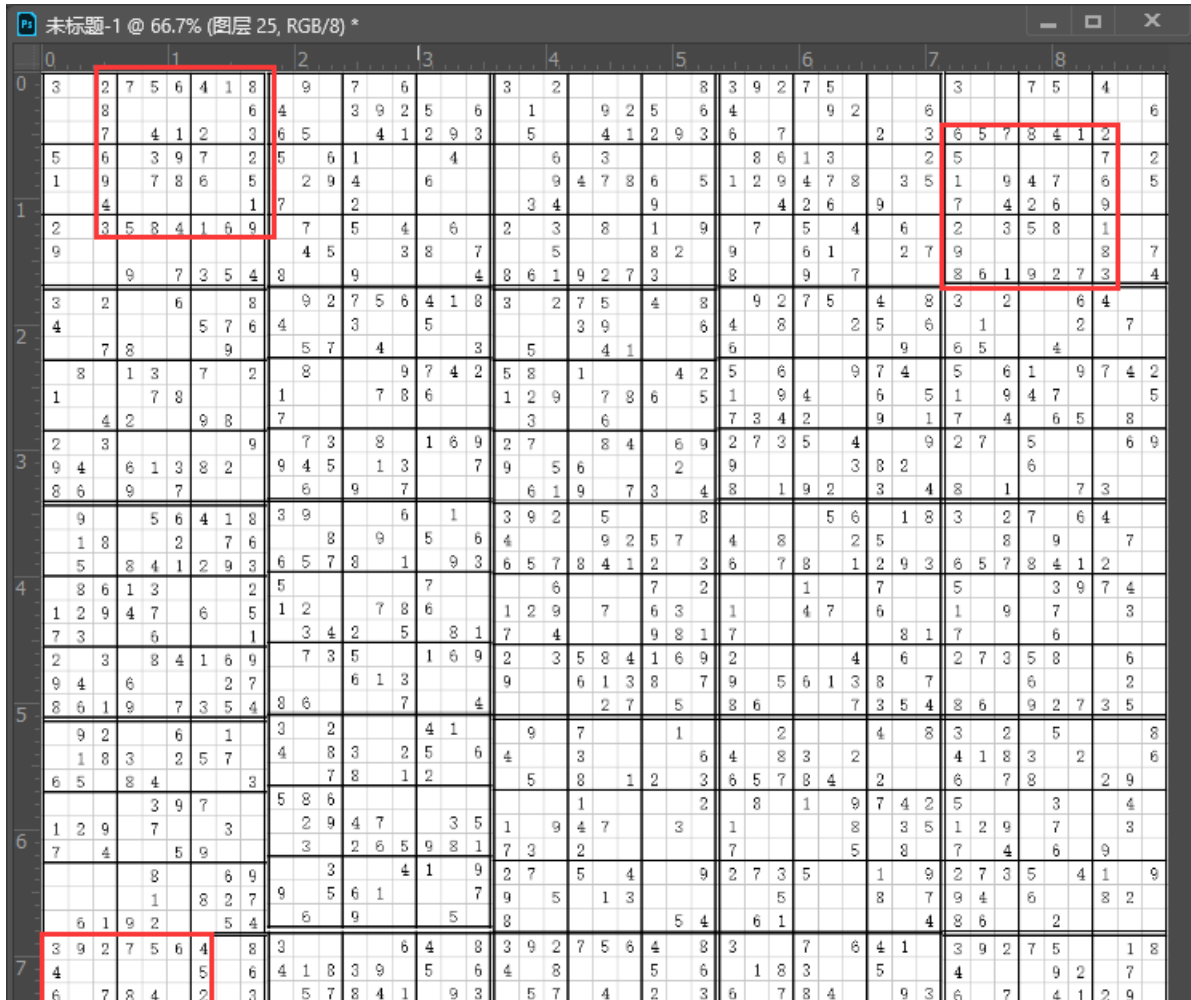
MISC 很普通的数独

关键字： 二维码

知识点： 常见二维码 除了右下角 其他角都有框框

步骤：

1.解压观察，换位置调整，第21张换到第1张的位置，第1张换到第5张的位置，第五章换到第21张的位置，还原,脚本还原二维码



5	6	1	3	7	2	5	8	1	9	7	2	5	6		7	4	2	5	8	3		2	5		3		2					
1	9	4	7	6		2	9	4			5	1	2	9	4	7	8	6	5	1	9		8		1	2	9	4	7	8	3	5
7				9		1	7	3	2	6	5	9	1	7	4				1		4	2	5	9		3	4	2	5	9	1	
2	7	3	5	8	4	1			8	4	1	9	2			4	1	9	7	5	8		6		7	7	8	8		9		
8	6		2	7	3			4	5		3	2	7	9	4	5		1	3		7	4	5		2	7	4	6	1			
8	6		2	7	3		6	9	7		4	8	6	1	2	7	3	5	4	8	1	9	7	3	5		9	7	3	4		

66.67% 文档:3.05M/100.7M > 未标题-1 @ 66.7% (图层 21, RGB/8) *

0	1	2	3	4	5	6	7	8																										
0	3	9	2	7	5	6	4	8	4	9	7	6		3	2			8	3	9	2	7	5			3	2	7	5	6	4	1	8	
1	4				5		6	4	4	9	2	3	9	2	5	6		1	4	8	7	9	2		6		8	7	4	1	2	3	6	7
2	6	7	8	4	2	3	6	5		4	1	2	9	3		5		4	1	2	9	3	6	7		2	3	6						
3	5	6	1	3	7	2	5	6	1		4				6	3			8	6	1	3		2	5	6	6	3	9	7	2			
4	1	9	4	7	6		2	9	4		6			9	4	7	8	6	5	1	2	9	4	7	8	3	5	1	9	4	7	8	6	5
5	7			9		1	7		2					3	4		9			4	2	6	9		4		2	3	5	8	4	1	6	9
6	2	7	3	5	8	4	1	9	7	5	4	6		2	3	8	1	9	7	5	4	6		2	3	5	8	4	1	6	9			
7	8	6		2	7	3		8	4	5		3	8	7		5		8	2	9		6	1	2	7	9		9	7	3	5	4		
8	3	2		6		8	9	2	7	5	6	4	1	8	3	2	7	5	4	8	9	2	7	5	4	8	3	2		6	4			
9	4			5	7	6	4		3		5		5		3	9		6	4	8		2	5	6		1			2	7				
0	8		1	3	7	2	1	8		9	7	4	2	5	8	1		4	2	5	6		9	7	4	5	5	6	1	9	7	4	2	
1			4	2		9	8	7		7	8	6		1	2	9	7	8	6	5	1	9	4	6	5	1	9	4	7	8	6	5		
2	2	3			8	2	9	7	3		8	1	6	9	2	7	8	4	6	9	2	7	3	5	4	9	2	7	5			6	9	
3	9	4		6	1	3	8	2	9	4	5	1	3	7	9	5	6		2	9		3	8	2	9		6							
4	8	6	9	7			8		6	9	7			6	1	9	2	7	3	4	8	1	9	2	3	4	8	1			7	3		
5	9		5	6	4	1	8	3	9			6	1	3	9	2	5		8		5	6	1	8	3	2	7	6	4					
6	1	8		2	7	6	6	8		9	5	6		4	4	8	9	2	5	7	4	8	2	5		8	9		9	7				
7	5		8	4	1	2	9	3	6	5	7	8	1	9	3	6	5	7	8	4	1	2	9	3	6	5	7	8	4	1	2	9	3	6
8	8	6	1	3	7	2	5		5		7			6		7	2			1		7					5							
9	1	2	9	4	7	6	5		1	2		7	8	6		1	2	9	7	6	3		1		4	7	6	1	9	7	3			
0	7	3		6		1			3	4	2	5	8	1	7	4		9	8	1	7			8	1	7		8	1	7	6			
1	2	3		8	4	1	6	9	7	3	5	1	6	9	2	3	5	8	4	1	6	9	2	3	5	8	4	1	6	9	2	3	5	8
2	9	4		6			2	7	7	3	5	1	6	9	9	6	1	3	8	7	9	5	6	1	3	8	7	9	5	6	1	3	8	7
3	8	6	1	9	7	3	5	4	8	6		7		4	8	6		2	7	5	8	6		7	3	5	4	8	6	9	2	7	3	5
4	9	2		6	1		3	2	4	1				9	7		1		2		4	8	3	2	5	6	4	1	8	3	2	5	6	
5	1	8	3	2	5	7	4	8	3	2	5	6		4	8	3	2	5	6	4	8	3	2	5	6	4	8	3	2	5	6	4	8	
6	6	5		8	4		3		7	8	1	2		5	8	1	2	3	6	5	7	8	4	2	2	6	5	7	8	4	2	9	6	
7	1	2	9	7		3	5	8	6		7			1		8	1	9	7	4	2	5	6	4	2	5	6	4	1	8	3	4		
8	7	4		5	9		3		2	9	4	7	3	5	1	9	4	7		3	1		8	3	5	1	2	9	7	6	9	3		
9			8		6	9	3		3		4	1	9	2	7	5	4		9	2	7	3	5		1	9	2	7	3	5	4	1	9	
0	6	1	9	2		5	4		9	5	6	1		7	9	5	1	3	8	7	9	5	6	1	3	8	7	9	5	6	1	3	8	7
1	3		7	5	4		6		8					8		5	4		6	1			4		6	1		8	6	9	2			
2	6	5	7	8	4	1	2		4	1	8	3	9	5	6	4	8	3	9	2	7	5	6	4	8	3	9	2	7	5	6	4	8	
3	5				7	2	5	8	1	9	7	2	5	6		7	4	2	5	8		3		2	5	8	1	8	3	9	2	7		
4	1	9	4	7	6	5	2	9	4		5	1	2	9	4	7	8	6	5	1	9	4	6	5	1	9	4	7	8	6	5	1	9	
5	7	4	2	6	9		7	3	2	6	5	9	1	7	4			1		4	2	5	9		4	2	5	9	3	4	2	5	9	
6	2	3	5	8	1				4	5		3	2	7	9	4	5		1	3		7	4	5		6	7	8	8					
7	9				8		7		4	5		3	2	7	9	4	5		1	3		7	4	5		6	7	8	8					
8	8	6	1	9	2	7	3	4	6	9	7		4	8	6	1	2	7	3	5	4	8	1	9	7	3	5	4	8	6	9	2	7	3

66.67% 文档:3.05M/104.0M > 未标题-1 @ 66.7% (图层 21, RGB/8) *

111111010101000101000011111000010111111
100000101100111101010011101100011001001000001
101110101110011111010011111101000101001011101
101110101101100010001010000011110001101011101
10111010001110010000111110111111011101011101
100000101100100000011000100001110100001000001
11111110101010101010101010101010101110111111
00000000001100110100100011010011001110000000
11001110010010000111111100100101000000101111
10100100101111111101110101011110101101001100
100000111100100100000110001101001101010001010
001100010011010001010011000100000010110010000
010110101010001111110100011101001110101101111
100011000100011100111011101101100101101110001
001100110100000000010010000111100101101011010
101000001011010111110011011111101001110100011
110111110111011001101100010100001110000100000
110101000010101000011101101101110101101001100
010011111110001011111010001000011011101101100
0110010110010101011000111101001100001010010
010111111111010111111101101101111111111100
011110001100000100001000101000100100100011110
111110101110011100111010110100110100101010010
110010001011101011101000111100000011100010000
1010111101110011101111111100001010111110010
110100011000111000100111101101111101000100010
111101111110001001000011010110001111110111110
011001010101000110010100010001000101101010001
011101110101101101100100001101101000111101001
110110001001101100010101101111110100101100110
000011100111000000000100001010101111100010010
111010010011110011101110010100001011111010010
101001100010111111110100000100001010101010100
000010011001001101110101001111100101111101101
000010111101110001101011000001000101110100110
011110011010100010100000011011000001110010000
10011010010000110111111101100101110111110011
000000001111110101101000101011100100100011010
111111100011111011011010101101110011101011110
100000101110101101101000111110010001100010001
10111010101110000111111101101001000111111011
101110100110111101101000001001101100011101101
101110100000011101100001101010110010010010001
100000101011001011111011001011000011010110000
111111101010101001111011110101101110000101101

```

from PIL import Image
x = 45
y = 45

im = Image.new('RGB', (x, y))
white = (255, 255, 255)
black = (0, 0, 0)

with open('数独.txt') as f:
    for i in range(x):
        ff = f.readline()
        for j in range(y):
            if ff[j] == '1':
                im.putpixel((i, j), black)
            else:
                im.putpixel((i, j), white)
im.save("数独.jpg")

```

2. 不断base64

得到二维码，扫描得到

Vm0xd1NtUXlWa1pPVldoVFIUSINjRIJVVGtOamJGWNlWMjFHVIUxV1ZqTldNakZMWcxS1lxTnNhRmhoTVZweVdWUkdXbVZHWkhOWGJGcHBWa1paZWxacpEUmhNVXBYVW14V2FHVnFRVGS9

base64解密一次

Vm1wSmQyVkZOVWhTYTJSfFRUTkNjbFZyV21GVU1WVjNWmjFHYW1KR1NsaFhhMVpyWVRGwMVGZHNxbFppVkZZelZrZDRhMUpXUmXWaGVqQtk=

base64解密两次

VmpJd2VFNUhSa2RpTTNCcIVrWmFUMVV3V21GamJGSlhXa1ZrYTFZeFdsWIZiVFYzVkd4a1JWRIVhejA9

base64解密三次

VjlweE5HRkdiM3BrUkZaT1UwWmFjBFJXWkVka1YxWIZVbTV3VGxkRVFUaz0=

base64解密四次

V20xNGFGb3pkRFZOU0ZaclRWZEdkV1ZVUm5wTldEQtk=

base64解密五次

Wm14aFozdDVNSfZrTVdGdWVURnpNWDA9

base64解密六次

ZmxhZ3t5MHVhMWFueTFzMX0=

base64解密七次

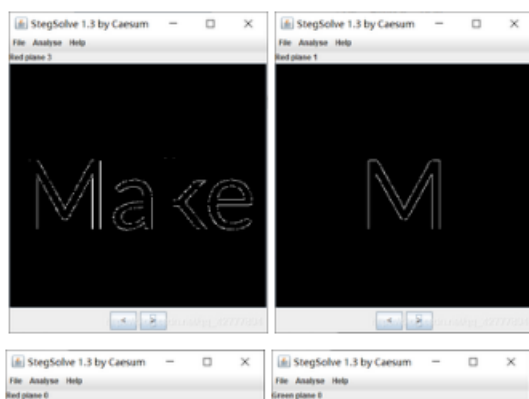
flag{y0ud1any1s1}

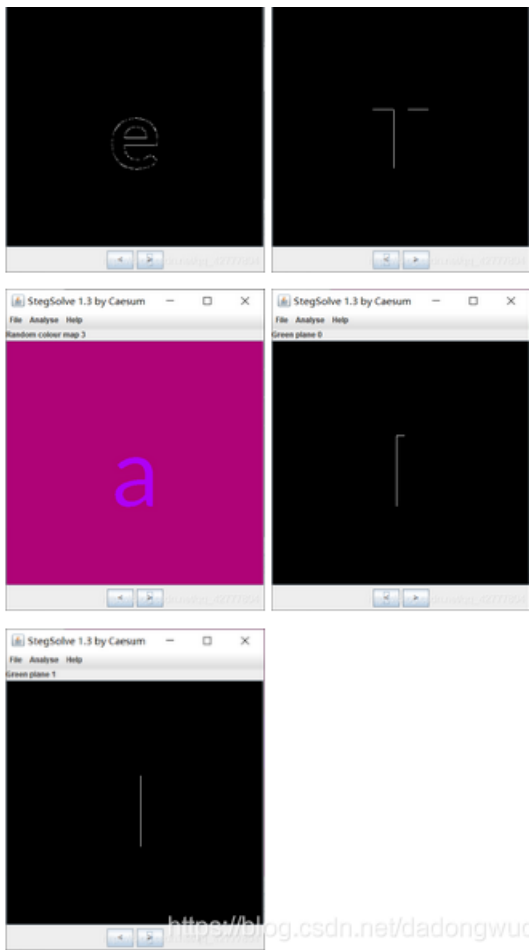
MISC color

关键字: stegsolve hex

步骤:

1. 组合起来 Make Me Tall





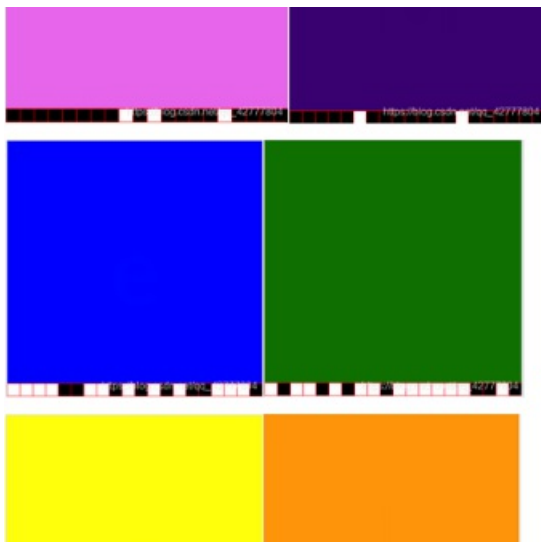
2.发现提交不了，hex修改高度

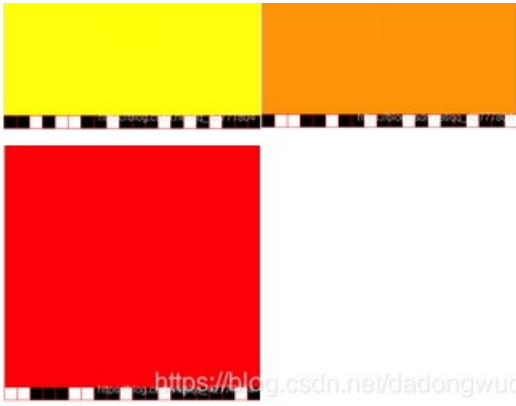
启动 1. png+ x

编辑为: 十六进制(H) 运行脚本 运行模板

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000h:	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52
010h:	00	00	01	90	00	00	01	FF	08	06	00	00	00	D0	D1	F7
020h:	A8	00	00	00	04	67	41	4D	41	00	00	D8	EB	F5	1C	14
030h:	AA	00	00	00	20	63	48	52	4D	00	00	87	0F	00	00	8C
040h:	0D	00	00	F9	93	00	00	84	E5	00	00	7B	82	00	00	EB
050h:	75	00	00	3F	B4	00	00	22	58	75	6B	5E	9C	00	00	04
060h:	18	69	43	43	50	6B	43	47	43	6F	6C	6F	72	53	70	61
070h:	63	65	47	65	6E	65	72	69	63	52	47	42	00	00	48	C7
080h:	8D	55	5D	68	1C	55	14	3E	BB	73	67	23	24	CE	53	6C

https://blog.csdn.net/dadongwudi





3.最后一行 白0 黑1 转化

```
'11111111010111101111',  
'11111011111110111111',  
'00001100101010110001',  
'01001010010000001101',  
'11010011011101010111',  
'10011011011010110110',  
'00111001101101111101'
```

发现第一列从上到下的二进制码刚好对应的ascii码

1100110 对应 f

```
|102 |66 |https://01100110|f|dad|ngwudi
```

```
c1 = '11111111010111101111'  
c2 = '11111011111110111111'  
c3 = '00001100101010110001'  
c4 = '01001010010000001101'  
c5 = '11010011011101010111'  
c6 = '10011011011010110110'  
c7 = '00111001101101111101'  
  
flag = ''  
  
for i in range(0,20):  
    c = c1[i]+c2[i]+c3[i]+c4[i]+c5[i]+c6[i]+c7[i]  
    flag += chr(int(c,2))  
  
print flag
```

参考链接: https://blog.csdn.net/qq_42777804/article/details/98974287

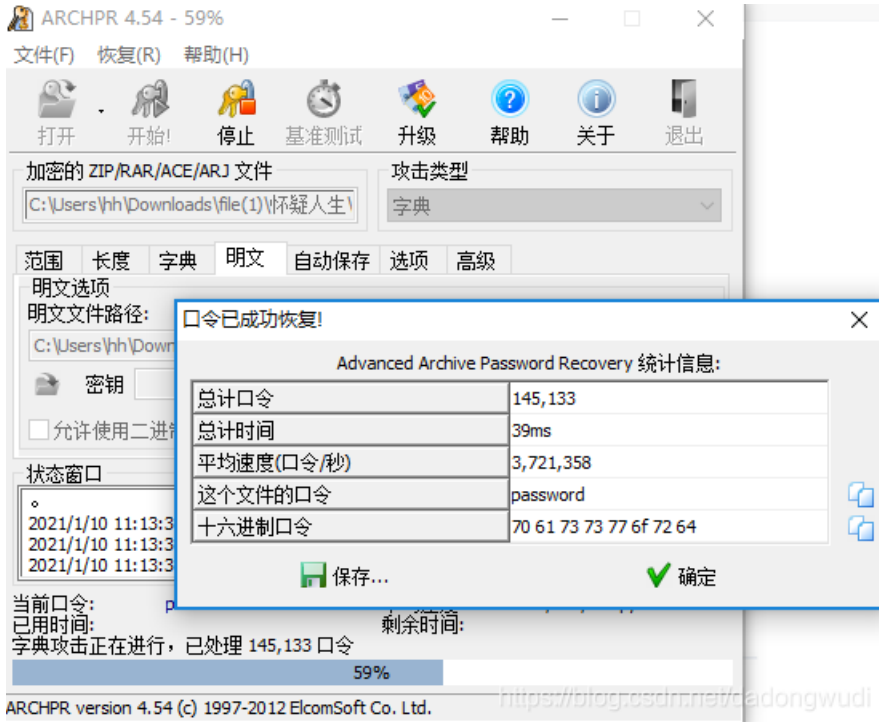
MISC 怀疑人生

关键字: archpr foremost psytec QR code editor

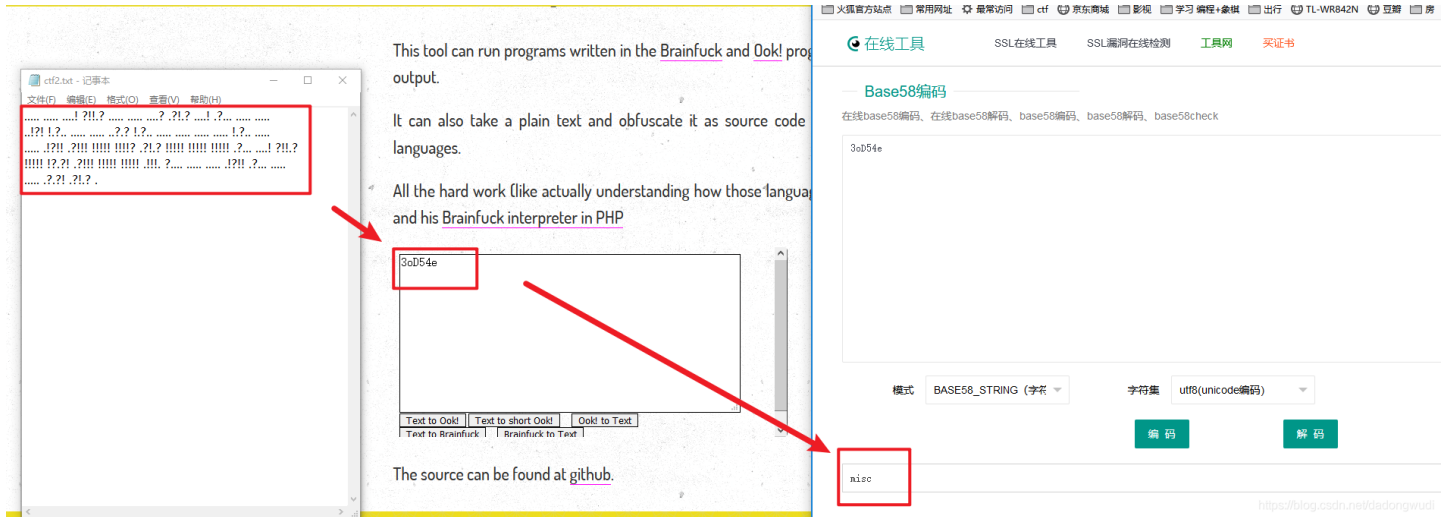
知识点:

步骤:

1.ctf1, 使用archpr字典或暴力破解 密码是password, 解压后base64, [ascii转unicode],(http://www.json.cn/unicode),得到 flag{hacker



2.foremost解压, ook编码, base58转码,得到 misc



3.pyotec QR code editor二维码识别 得到12580}



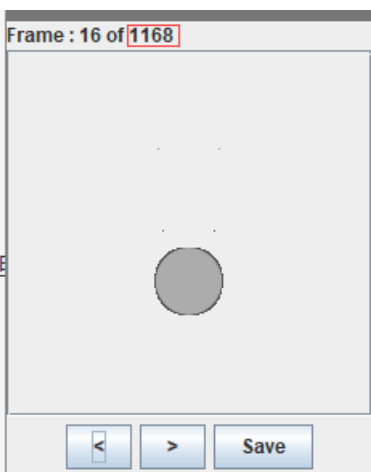
flag{hackermisc12580}

MISC 红绿灯

关键字: Stegsolve PIL

步骤

1.Stegsolve分析发现一共有1168帧图片



2.一帧一帧查看发现多数是红色和绿色，偶尔有黄色且（每8个红绿后跟一个黄），可以推测红色和绿色对应二进制0和1，黄色作为分隔，这样第一个黄灯之前数值为01100110或10011001，而01100110二进制转换成ascii对应字符就是'f',依次可以验证前四个字符为flag

其中有红色，绿色，还有少部分的黄色，相对应的应该对应1、0、空格，再转换成字符串即可，脚本如下。

用画图定位到红绿灯所在坐标（是一个区域，随便取一个判断颜色值来确定该帧是红灯还是绿灯），红灯为(115,55)，绿灯为(115,150)，输出该坐标下的颜色值发现为红灯是颜色值是251，为绿灯时颜色值是186

```
from PIL import Image

im = Image.open("1.gif")
binstr = ""
flag = ""

for i in range(2,1168,2):
    im.seek(i)
    tmp1 = im.getpixel((115,55))
    tmp2 = im.getpixel((115,150))
    if(tmp1 == 251):
        binstr += '1'
    elif(tmp2 == 186):
        binstr += '0'
    else:
        binstr += ' '
#print(binstr)

for i in range(0,len(binstr) - 1,9):
    flag += chr(int(('0b' + binstr[i:i+7]),2))

print(flag)
```

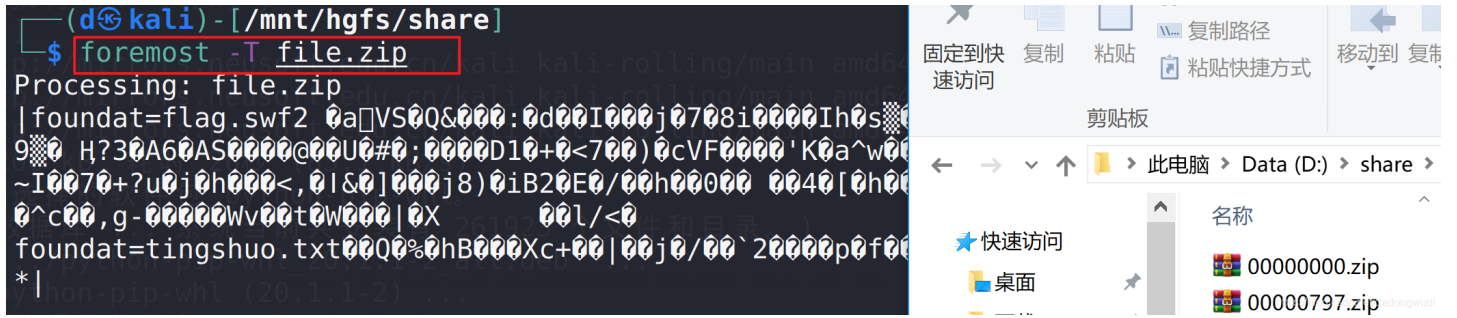
flag{Pi34s3_p4y_4tt3nt10n_t0_tr4ff1c_s4f3ty_wh3n_y0u_4r3_0uts1d3}

MISC 不简单的压缩

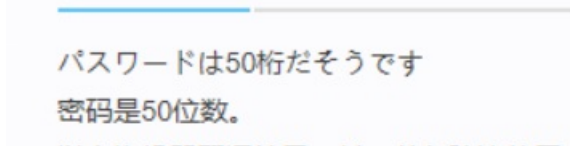
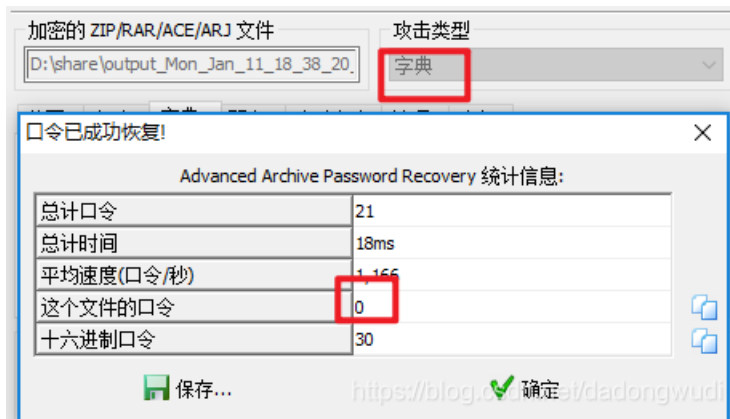
关键字： Flash Decompiler winhex foremost 暴力破解 字典爆破

步骤:

1.winhex观察 发现有两个PK头, foremost分离

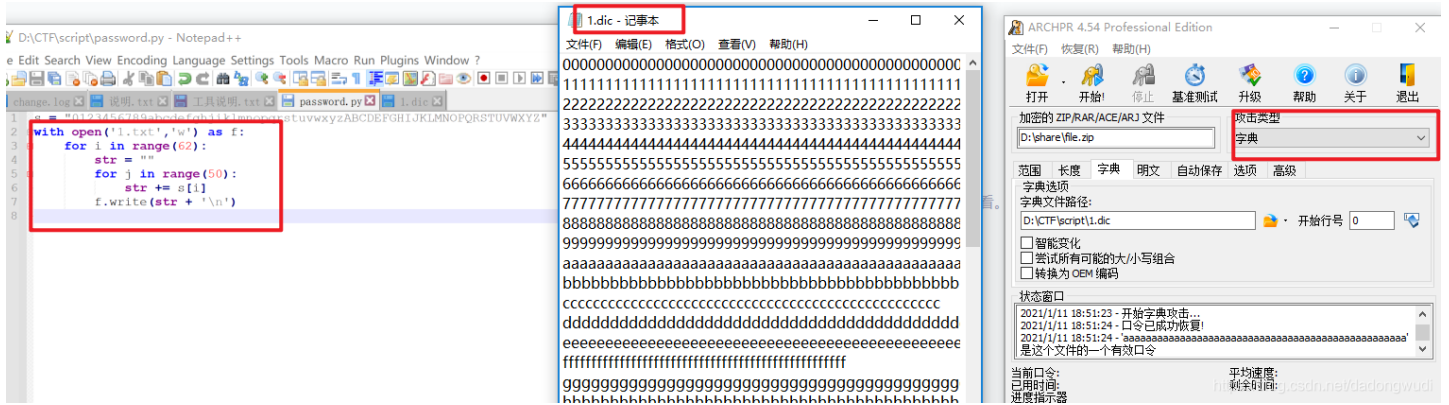


archpr自带字典破解小压缩包, 得到txt文档, 翻译

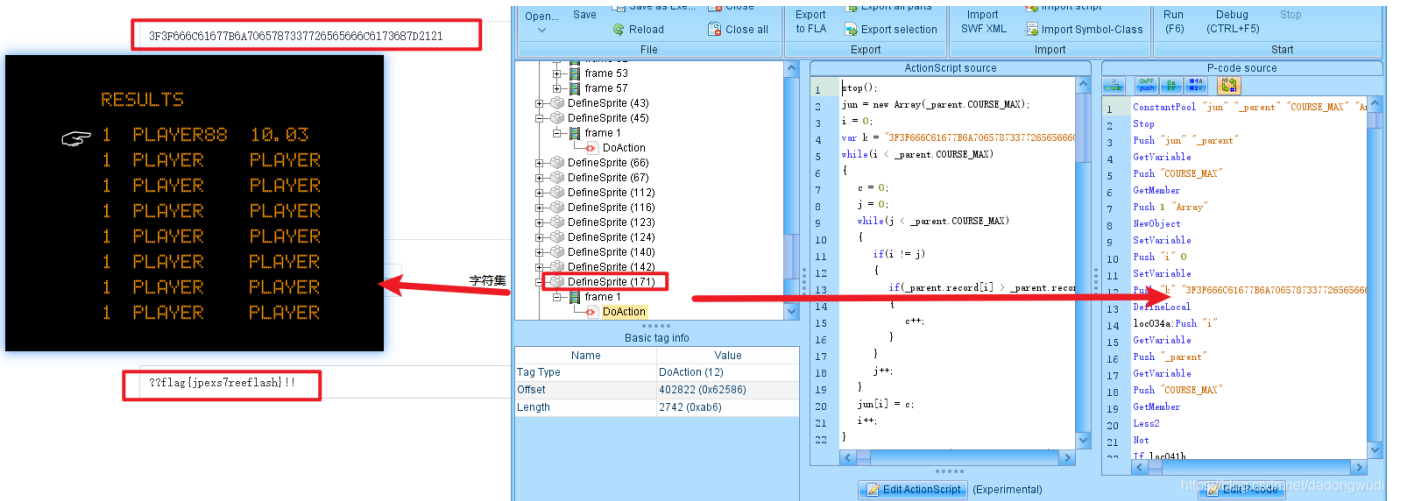


2.密码是50位, 应该就是同样的字符了, 50个而已, 不然的话基本上是无解的, 先生成了一个字母数字的字典, txt文档改后缀dic, 字典破解

```
s = "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
with open('1.txt', 'w') as f:
    for i in range(62):
        str = ""
        for j in range(50):
            str += s[i]
        f.write(str + '\n')
```



3.JPEXS Free Flash Decompiler下载地址: 链接: <http://pan.baidu.com/s/1jHOxB2Q> 密码: nda5
 JPEXS Free Flash Decompiler反编译flash文件 在171中找到base64, 解码



flag{jpexs7reeflash}

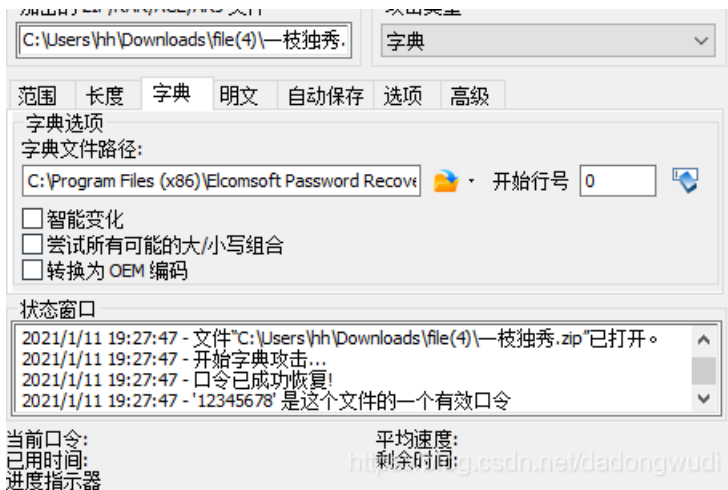
MISC 一枝独秀

关键字: stegSolve winhex ARCHPR jphs

知识点:

步骤:

- 1.winhex查看png 发现PK头 改后缀zip
- 2.需要密码, ARCHPR字典爆破 解压后发现81张大小不一 查看属性





flower (81).jpg 属性

常规

安全

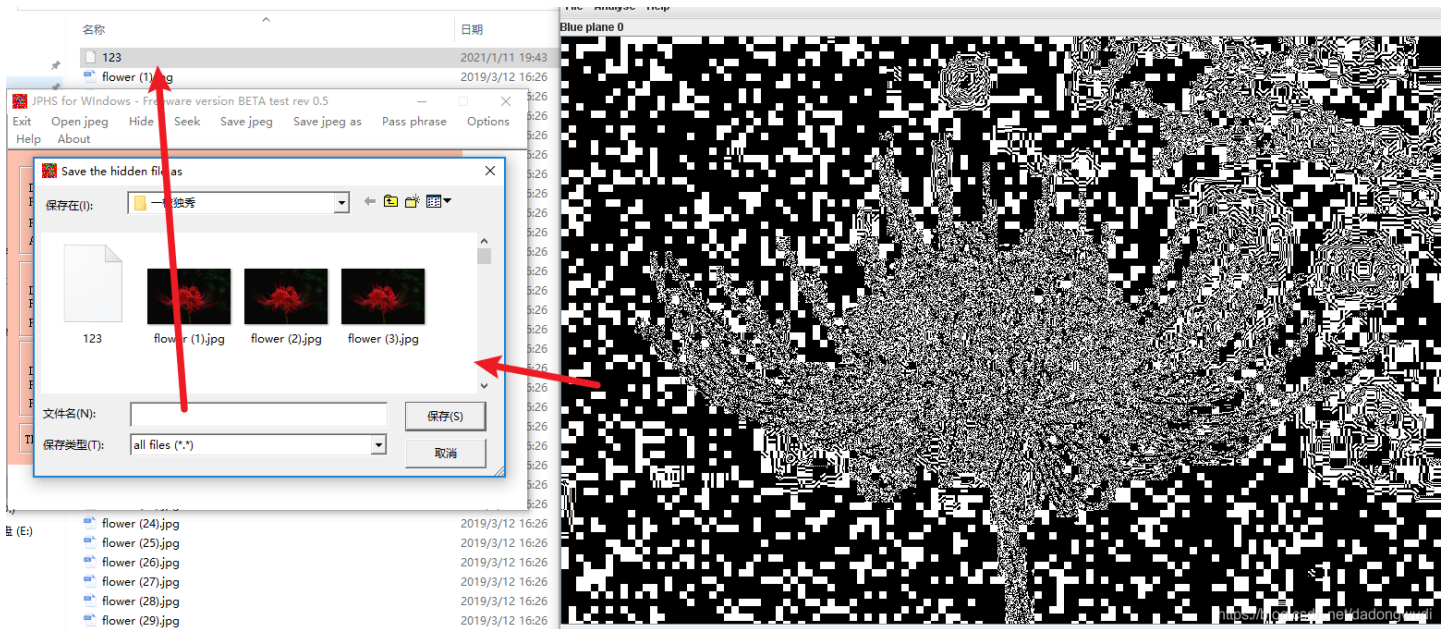
详细信息

以前的版本

属性	值
说明	
标题	
主题	flowers
分级	☆☆☆☆☆
标记	

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

3.stegSolve查看 有东西 jphs05提取 填入flowers作为密码，提取后winhex打开，发现PK头，改后缀，解压txt文档，佛说，根据hint进行栅栏解密，base64解密



<http://www.keyfc.net/bbs/tools/tudoucode.aspx>

与佛论禅

H-hDs100ZL31hIZZbeRSbbbVRZMm32W2X33mGm3Tx+t999RdV9lx0

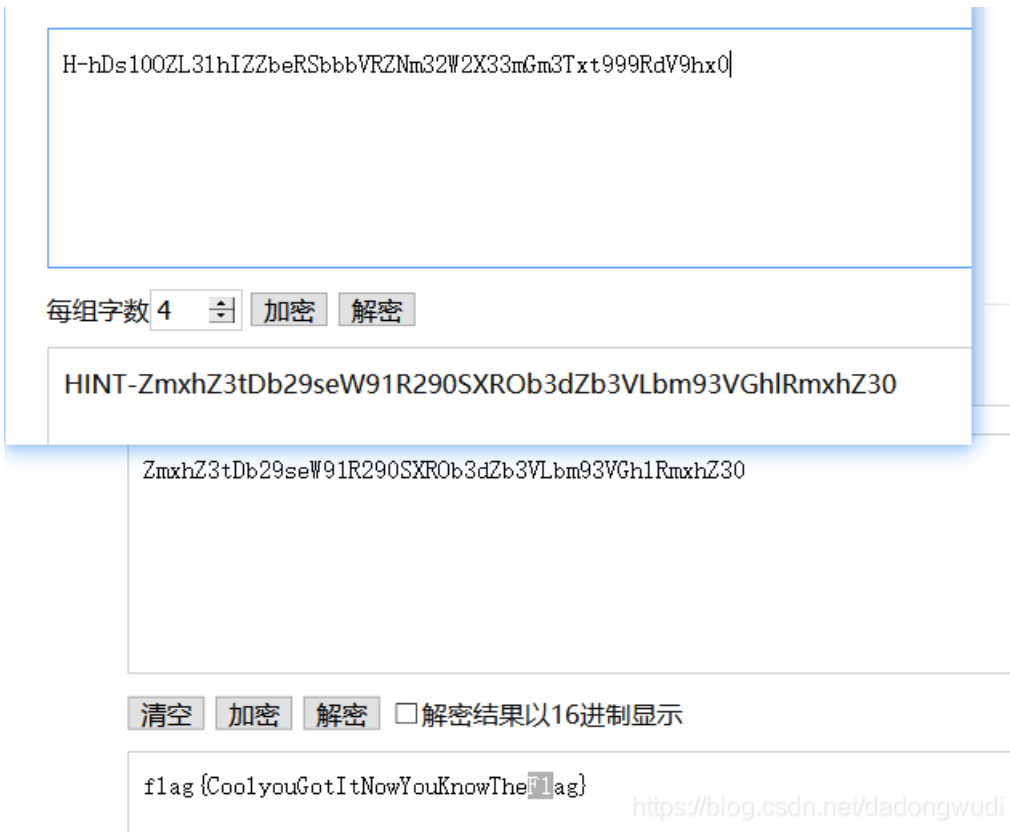
听佛说宇宙的真谛 **参悟佛所言的真意** 普度众生

刹那便是永恒

佛曰：阿霸豆赫婆提诸真诸迦亦任乘任大梵尼朋梵澜吟翻余旧奢般是诸燥悉吟燥冥参特参性皇瞻吉赫阿基耶暗诸
任真融诸诸尼赫曰。梵究呐稠虚他怪明曼究呐得吟获集能冥晝滅知俱朋性室神奢囉姪豆霸帝造赫明梵苦奢密任曰
赫者特吟呼赫赫不冥死等那阿冥恋奢薩巨皇赫波割。霸摩任故霸夢赫恐瞻寫諸閣舍吟得波苦奢即霸恐冥道一吟究
梵呼冥閣吟上霸南詞諸寫真依瞻者吟諸故死吟夷普任曰呐逝至瞻佛諸耶

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





MISC 妹子的陌陌

关键字: forewalk

步骤:

1.foremost分解出一个压缩包，但是解压需要密码，图片上写到：喜欢我吗。



2.文本内容 莫斯密码解密，输入网址，二维码（黑白相反或者反色处理）微信扫码 orQR_Research有自动纠错功能(本人无法使用，打开后点击任务栏在切换大小写)

摩斯电码解密结果: <http://encode.chahuo.com/>

是一个解密网站。

下面的是AES解密，U2FsdGVkX18tl8Yi7FaGiv6jK1SBxKD30eYb52onYe0=是密文，@#@@#¥%.....¥¥%%.....&¥是密钥，拿去解密得到momoj2j.png，访问<http://c.bugku.com/momoj2j.png>，得到一个二维码：


```
import zipfile
import string
import binascii

def CrackCrc(crc):
    for i in dic: #迭代的不是值而是键 (key)
        for j in dic:
            for k in dic:
                for h in dic:
                    s = i + j + k + h
                    if crc == (binascii.crc32(s.encode())):
                        f.write(s)
                        return

def CrackZip():
    for i in range(0,68):
        file = 'out'+str(i)+'.zip'
        crc = zipfile.ZipFile(file, 'r').getinfo('data.txt').CRC
        CrackCrc(crc)

dic = string.ascii_letters + string.digits + '+/='

f = open('out.txt', 'w')
CrackZip()
print("CRC32碰撞完成")
f.close
```

2.得到的是base64编码之后的字符，使用base64解码，将解码结果复制到记事本，使用全局替换\

```
6s
>z#IT4kcI3CMT "AOSHÖEQAF B |m+il(,3(H8,Fv+gMrJQt
;?"Jt -baScI0 flag.txt4ifix the file and get the flagL{
```

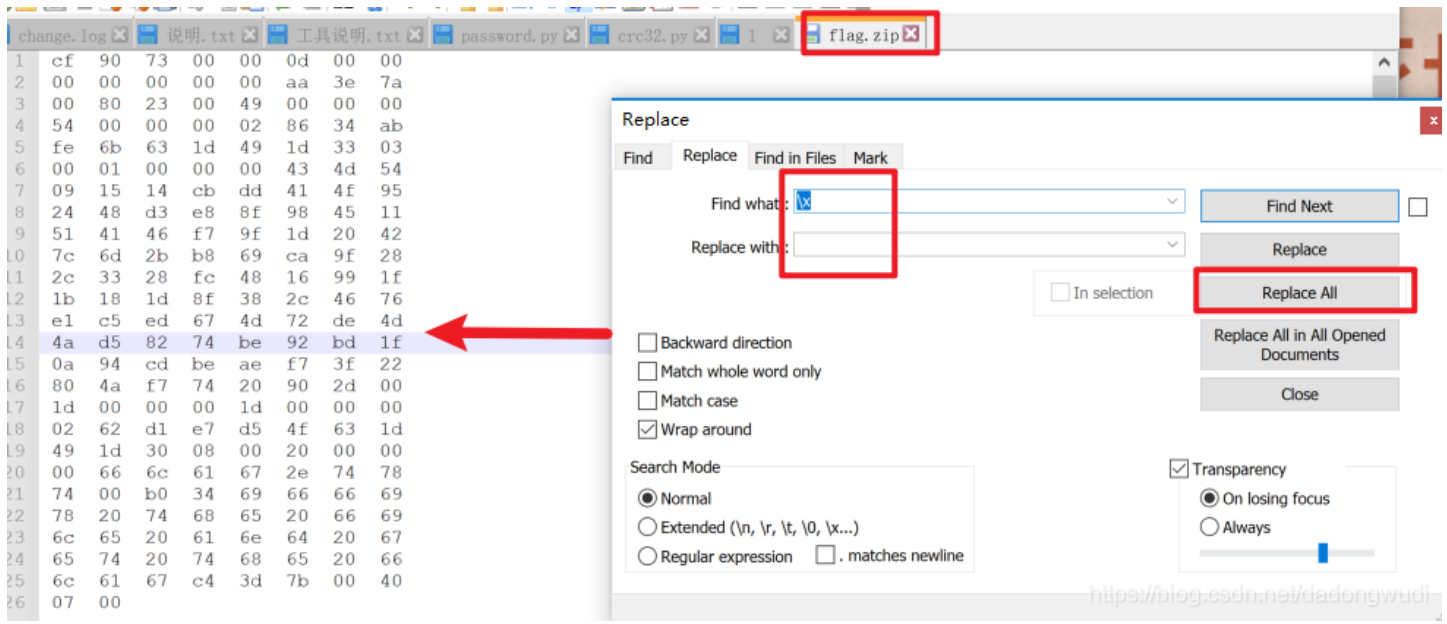
```
z5BzAAAAAAAAAAAKo+egCAIwBJAAAAVAAAAKGNkv+a2MdSR0zAwABAAAQ01UCRUUy91BT5UkSNPoj5hFEVFBFRvefHSBCfG0ruGnKnygsMyj8SBaZHxs
YHY84LEZ24cXtZ01y3k1K1YJ0vpK9HwqUzb6u9z8igEr3dCCQLQAdAAAAHQAAAAJi0efVT2MdSR0wCAAgAAAAZmxhZy50eHQAsDRpZmZpeCB0aGUgZmlsZS
BhbmQgZ2V0IHRoZSBmbGFndD17AEHAA==
```

清空 加密 解密 解密结果以16进制显示

```
\xcf \x90 \x73 \x00 \x00 \x0d \x00 \x00
\x00 \x00 \x00 \x00 \x00 \xaa \x3e \x7a
\x00 \x80 \x23 \x00 \x49 \x00 \x00 \x00
\x54 \x00 \x00 \x00 \x02 \x86 \x34 \xab
\xfe \x6b \x63 \x1d \x49 \x1d \x33 \x03
\x00 \x01 \x00 \x00 \x00 \x43 \x4d \x54
```

复制

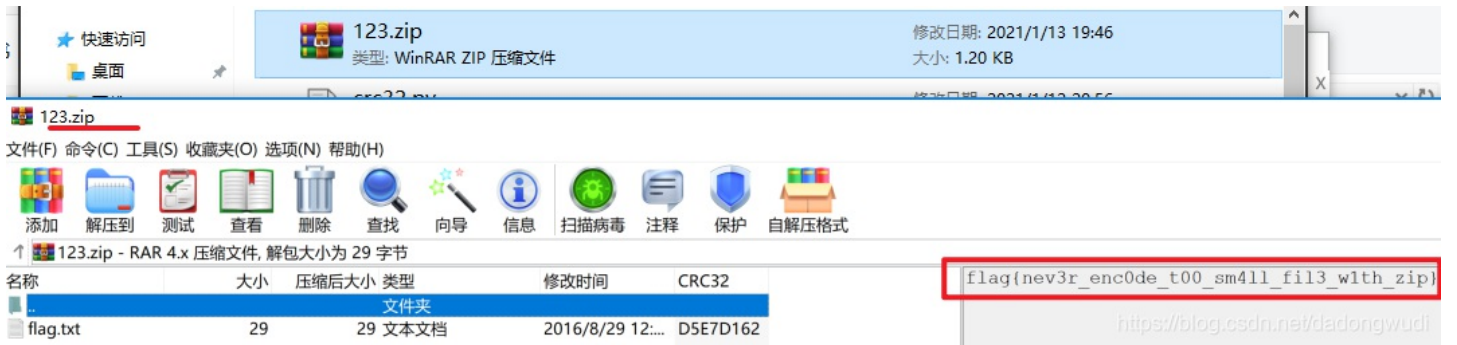
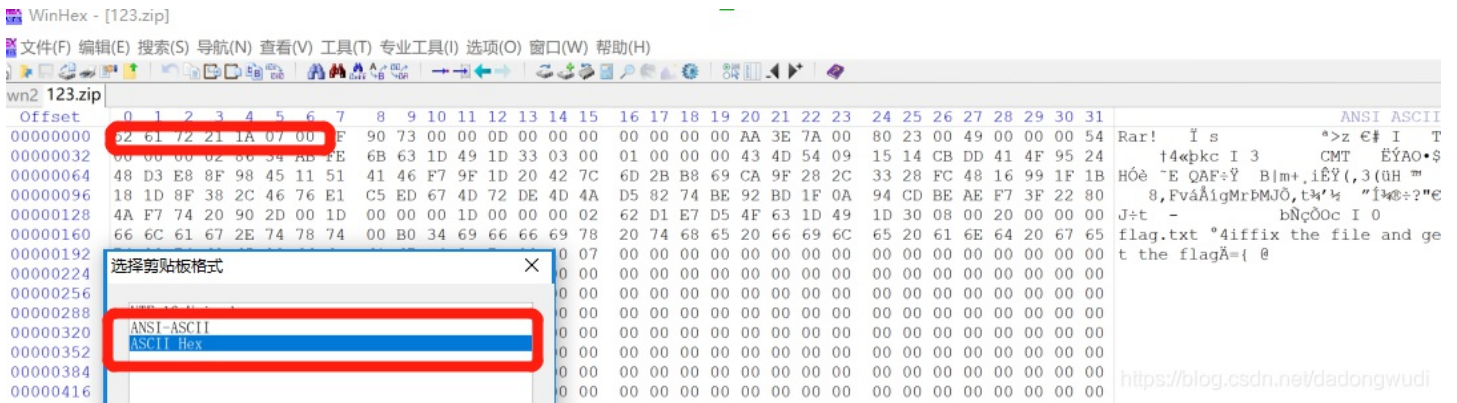
<https://blog.csdn.net/dadongwudi>



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

3.根据flag.txt可以知道这是个压缩包，而且需要我们修复文件才能得到flag，将base64解码之后的文件复制到winhex中，发现有rar文件的文件尾C4 3D 7B 00 40 07 00，还存在一个名为CMT的文件，即"注释"

先保存为rar文件，然后使用UE打开，插入十六进制，补上rar的文件头52 61 72 21 1A 07 00,然后保存，打开压缩包得到flag



flag{nev3r_enc0de_t00_sm4ll_fil3_w1th_zip}

MISC

关键字:

知识点:

步骤:

杂七杂八:

1. Burpsuite Fiddler Wireshark简单对比

1. Burpsuite基于java语言，具有跨平台的优势；
2. Fiddler基于.Net，入门简单，功能强大，但是只适用win平台，目前借助mono也无法很好的在mac、linux平台运行；
3. Wireshark各平台都有相应版本，不过入门门槛相对较高。
Burpsuite和Fiddler在抓包上侧重请求和响应的数据，Wireshark侧重于数据帧

2.grep的基本用法

https://blog.csdn.net/successdm/article/details/90145432?utm_medium=distribute.pc_relevant.none-task-blog-baidujs_title-2&spm=1001.2101.3001.4242

3.CTF密码学常见加解密总结

https://blog.csdn.net/qq_40836553/article/details/79383488

https://blog.csdn.net/qq_40837276/article/details/83080460?utm_medium=distribute.pc_relevant.none-task-blog-baidujs_title-2&spm=1001.2101.3001.4242

4.破解ZIP加密文件

<https://www.cnblogs.com/ECJTUACM-873284962/p/9387711.html>

5.常见的文件头尾

https://blog.csdn.net/qq_29277155/article/details/98060616

```
7z
文件头标识: 37 7A BC AF 27 1C
JPEG/JPG
文件头标识: ff, d8 (SOI) (JPEG 文件标识)
文件结束标识: ff, d9 (EOI)
PNG
文件头标识: 89 50 4E 47 0D 0A 1A 0A
GIF
文件头标识: 47 49 46 38 39(37) 61 GIF89(7)a
BMP
文件头标识: 42 4D--- BM
HTML (html)
文件头标识: 68746D6C3E
ZIP Archive (zip)
文件头标识: 504B0304 PK
RAR Archive (rar)
文件头标识: 52617221
```

```
JPEG (jpg), 文件头: FFD8FFE0

PNG (png), 文件头: 89504E47

GIF (gif), 文件头: 474946383961

ZIP Archive (zip), 文件头: 504B0304

RAR Archive (rar), 文件头: 52617221

Wave (wav), 文件头: 57415645

AVI (avi), 文件头: 41564920

Real Audio (ram), 文件头: 2E7261FD

Real Media (rm), 文件头: 2E524D46

MPEG (mpg), 文件头: 000001BA

MPEG (mpg), 文件头: 000001B3

7z文件头: 37 7A BC AF 27 1C
```

6.jpeg格式详解

https://blog.csdn.net/yun_hen/article/details/78135122

参考链接: <https://www.cnblogs.com/cat47/p/11432475.html>

参考链接: https://blog.csdn.net/qq_39629343/article/details/80611614



[创作打卡挑战赛](#) >

[赢取流量/现金/CSDN周边激励大奖](#)