CAICT
中国信息通信研究院
China Academy of Information and Communications Technology

MISC in CTF

中国信息通信研究院
邵子扬

MISC in CTF

CAICT 中国信息通信研究院

- MISC ——miscellaneous
  - 包含各种难以归类的题目
  - 内容：PPC / FORENSIC / STEGO / Else
  - 特点：题目范围大，考察内容多
    - 从前沿到老旧
    - 从热门到冷僻
    - 从逻辑到脑洞
  - 难点：取决于选手 *知识面范围，知识熟悉程度，脑洞大小*

CAICT
中国信息通信研究院

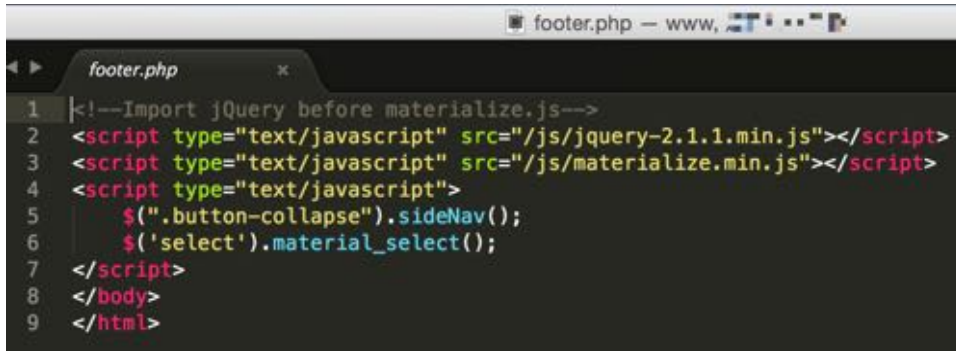
01

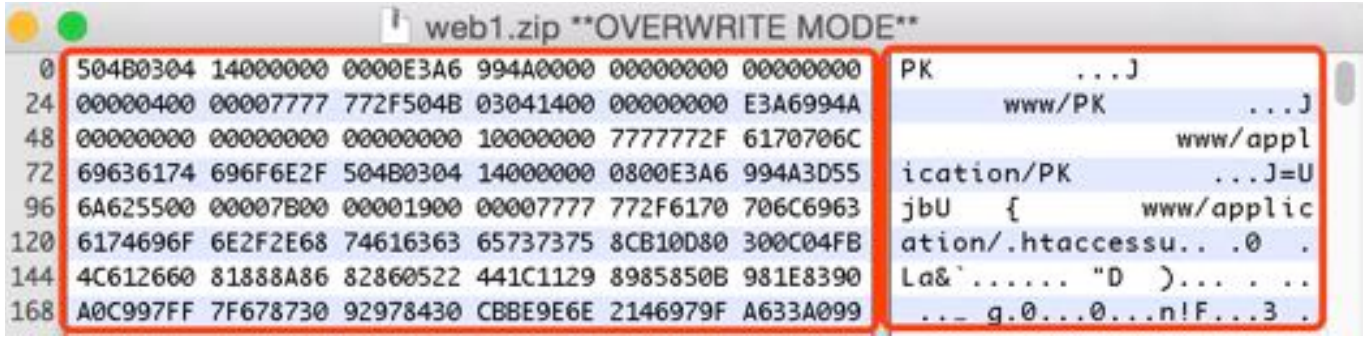
取证技术

文件取证技术
压缩文件取证
Wireshark与抓包
硬盘与内存取证
日志分析技术

文件

- 存储在某种**长期**储存设备或**临时**存储设备中的一段**数据流**
- 分类
  - 文本文件 —— 可见字符组成，用sublime打开可见
  - 二进制文件 —— 任意字符组成，可用WinHex打开





所有文件都是字符 -> 所有文件都是16进制 -> 所有文件都是二进制

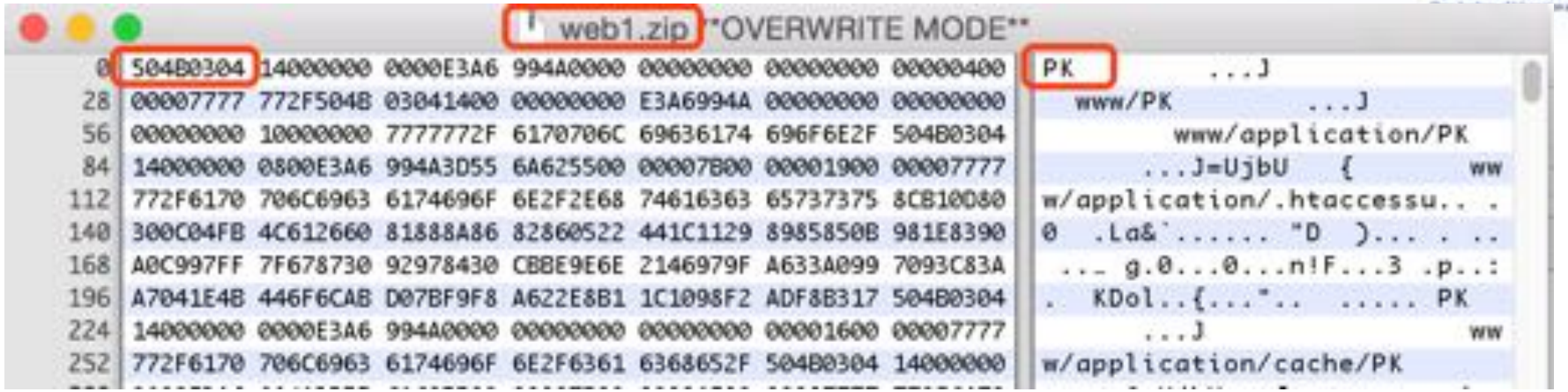
文件魔数

很多类型的文件，其起始的几个字节的内容是固定的（或是有意填充）。根据这几个字节的内容就可以确定文件类型，因此这几个字节的内容被称为魔数 (magic number)。

那么文件后缀是必须的么？

Linux下就有很多文件没有后缀，基本通过魔数进行内容识别；
Mac 和 Windows下则多有后缀，用于用特定程序打开特定文件。
如：docx和zip

docx与zip



将后缀docx改成zip，可以解压缩

常见魔数与文件修复

文件类型	魔数	ASCII 特征
Win PE	4D 5A	MZ
ELF	7F 45 4C 46	.ELF
RAR	52 61 72 21	Rar!....
ZIP	50 4B 03 04	PK..
7Z	37 7A BC AF 27 1C	7z¼¯'
JPEG	FF D8 FF DB	ÿØÿÛ
PNG	89 50 4E 47 0D 0A 1A 0A	.PNG....
GIF	47 49 46 38 39 61	GIF89a
BMP	42 4D	BM
PDF	25 50 44 46	%PDF
Java-class	CA FE BA BE	Êþº¾
VMDK	4B 44 4D	KDM

* JPEG 有“FF D9”作为文件结尾

FILE & BINWALK

CAICT 中国信息通信研究院

- File命令

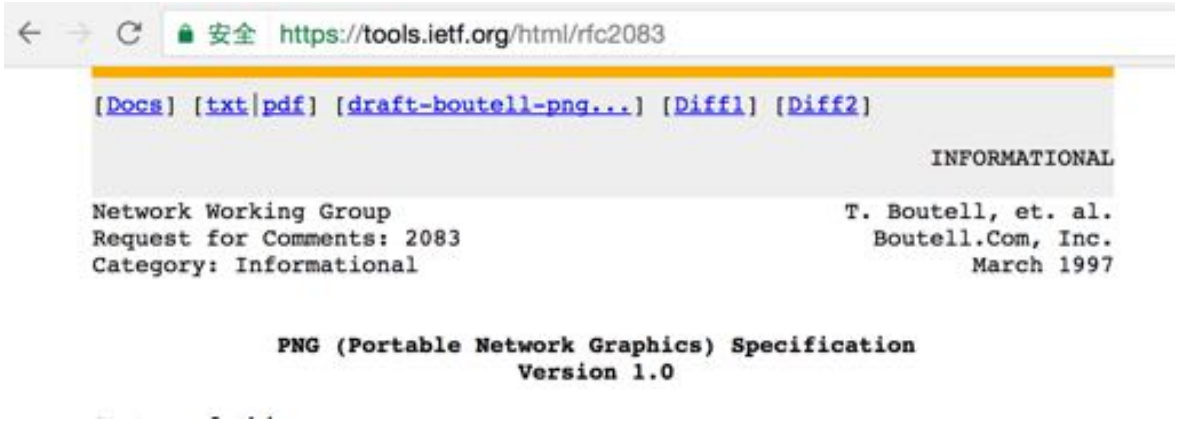
```
➜  burpsuite_pro_v1.7.11 file BurpLoader.jar
BurpLoader.jar: Zip archive data, at least v2.0 to extract
➜  burpsuite_pro_v1.7.11 file hs_err_pid10451.log
hs_err_pid10451.log: ASCII FORTRAN program text
➜  burpsuite_pro_v1.7.11
```

- Binwalk命令

```
➜  ~ binwalk /Users/[redacted]/Web_pentest/PHP-backdoors/Deobfuscated/angel.php

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
4993          0x1381          HTML document header
60552         0xEC88          Copyright string: "Copyright (C) 2004-2011 <a href="http://www.4ngel.net" target="_blank">Security
60692         0xED14          HTML document footer
67813         0x108E5         eCos RTOS string reference: "ecos: '.$array['gecos'].'&#13&#10Dir:  '.$array['dir'].'&#13&#10Shell:
67830         0x108F6         eCos RTOS string reference: "ecos'].'&#13&#10Dir:  '.$array['dir'].'&#13&#10Shell:  '.$array['shell']

➜  ~ binwalk /Users/[redacted]/Screenshot_20170724-111655.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 1080 x 1920, 8-bit/color RGB, non-interlaced
56            0x38            Zlib compressed data, default compression
```

进一步理解文件格式

- RFC document



安全 https://tools.ietf.org/html/rfc2083

[Docs] [txt|pdf] [draft-boutell-png...] [Diff1] [Diff2]

INFORMATIONAL

Network Working Group — T. Boutell, et. al.
Request for Comments: 2083 — Boutell.Com, Inc.
Category: Informational — March 1997

PNG (Portable Network Graphics) Specification
Version 1.0

- wikipedia

Technical details [edit]

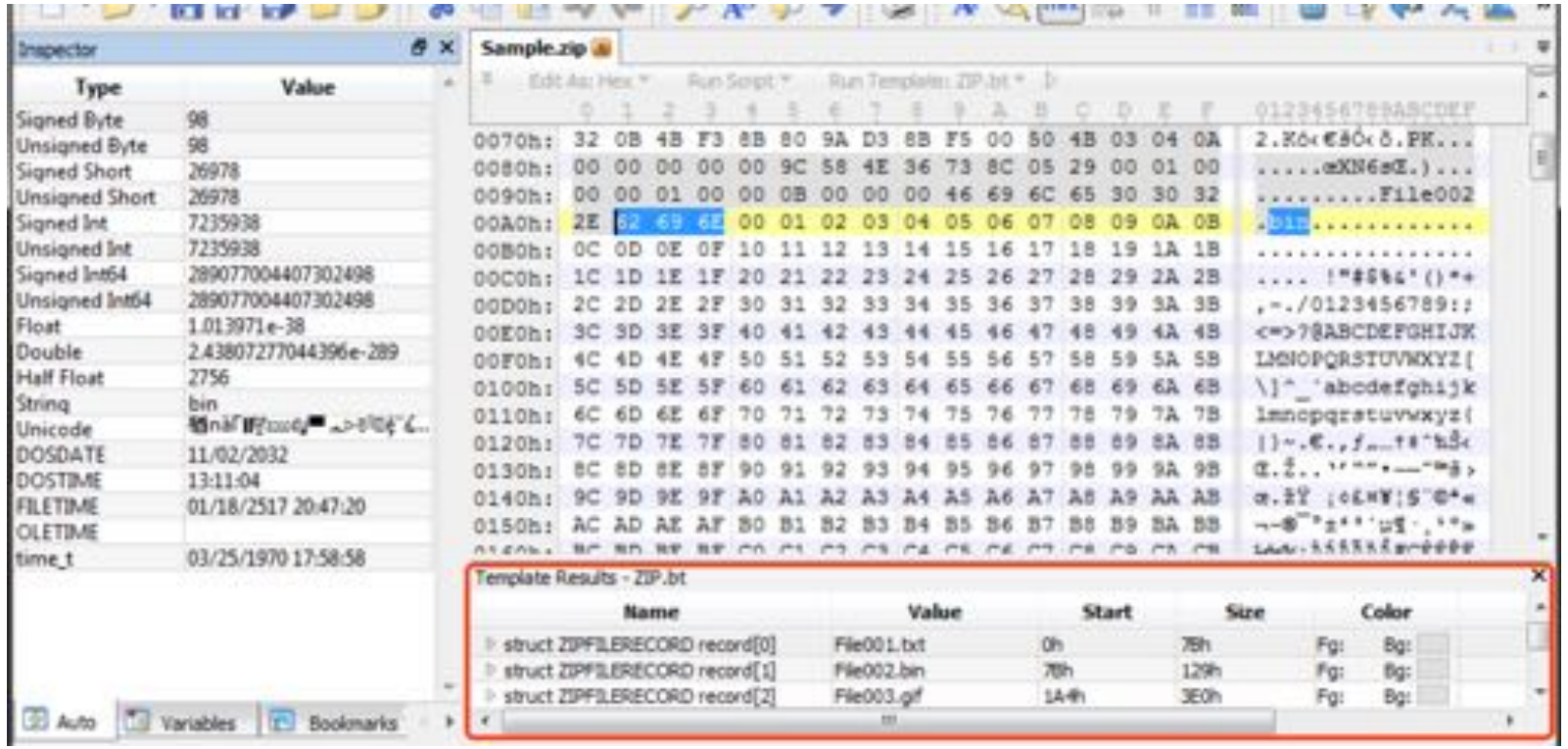
File header [edit]

A PNG file starts with an 8-byte signature:[10] (see hex editor image on the right)

Values	Purpose
89	Has the high bit set to detect transmission systems that do not support 8 bit data and to reduce the chance that a text file is mistakenly interpreted as a PNG, or vice versa.
50 4E 47	In ASCII, the letters *PNG*, allowing a person to identify the format easily if it is viewed in a text editor.
0D 0A	A DOS-style line ending (CRLF) to detect DOS-Unix line ending conversion of the data.
1A	A byte that stops display of the file under DOS when the command type has been used—the end-of-file character.
0A	A Unix-style line ending (LF) to detect Unix-DOS line ending conversion.

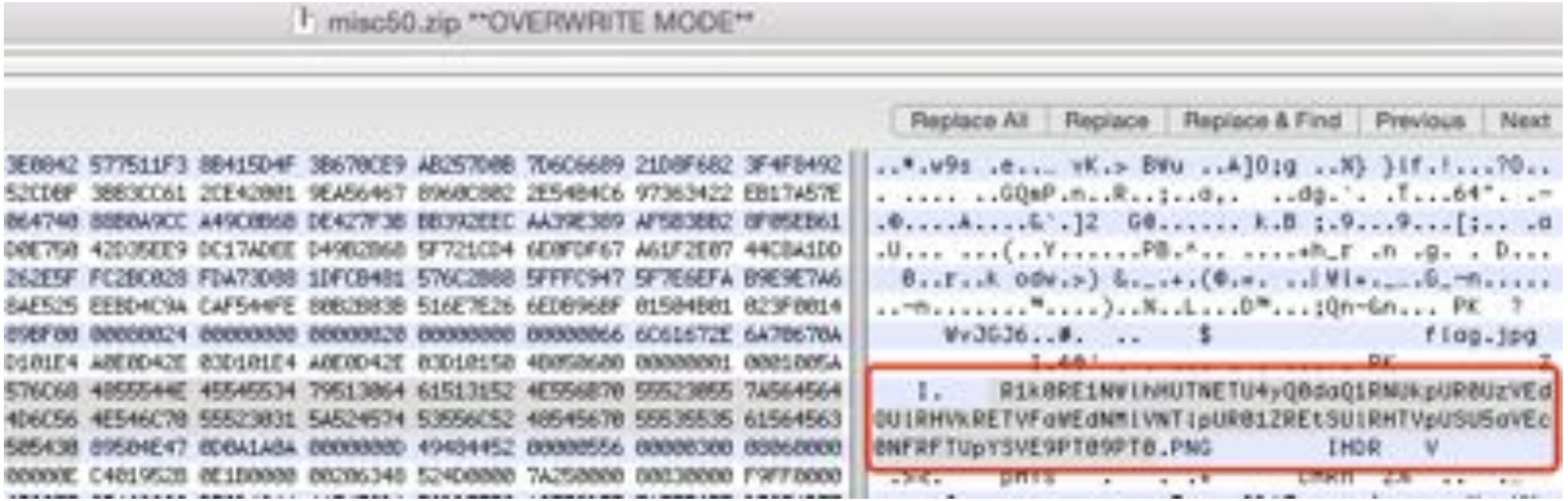
进一步理解文件格式

- 010 Editor + Template



在文件中插入字符

文件中有很多地方可以加入字符，但是不影响文件的正常使用。



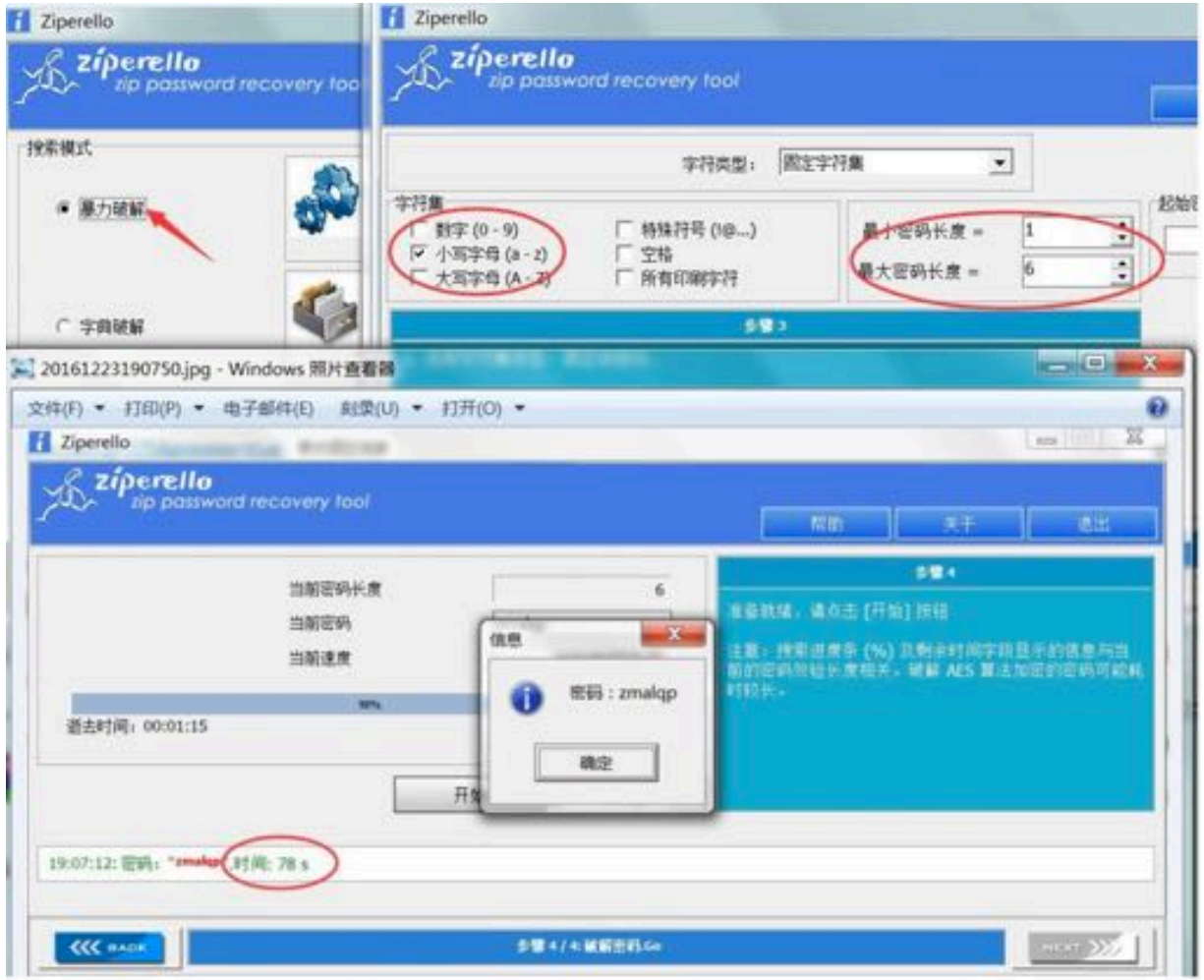
Office取证

- 考点
  - 藏在解压文件中
  - 藏在doc十六进制中
  - 文件爆破
- 工具
  - Advanced Office Password Recovery

压缩文件取证

- 考察点
  - 文件修复
  - 密码爆破
  - 伪加密位
  - 明文攻击
  - CRC32碰撞

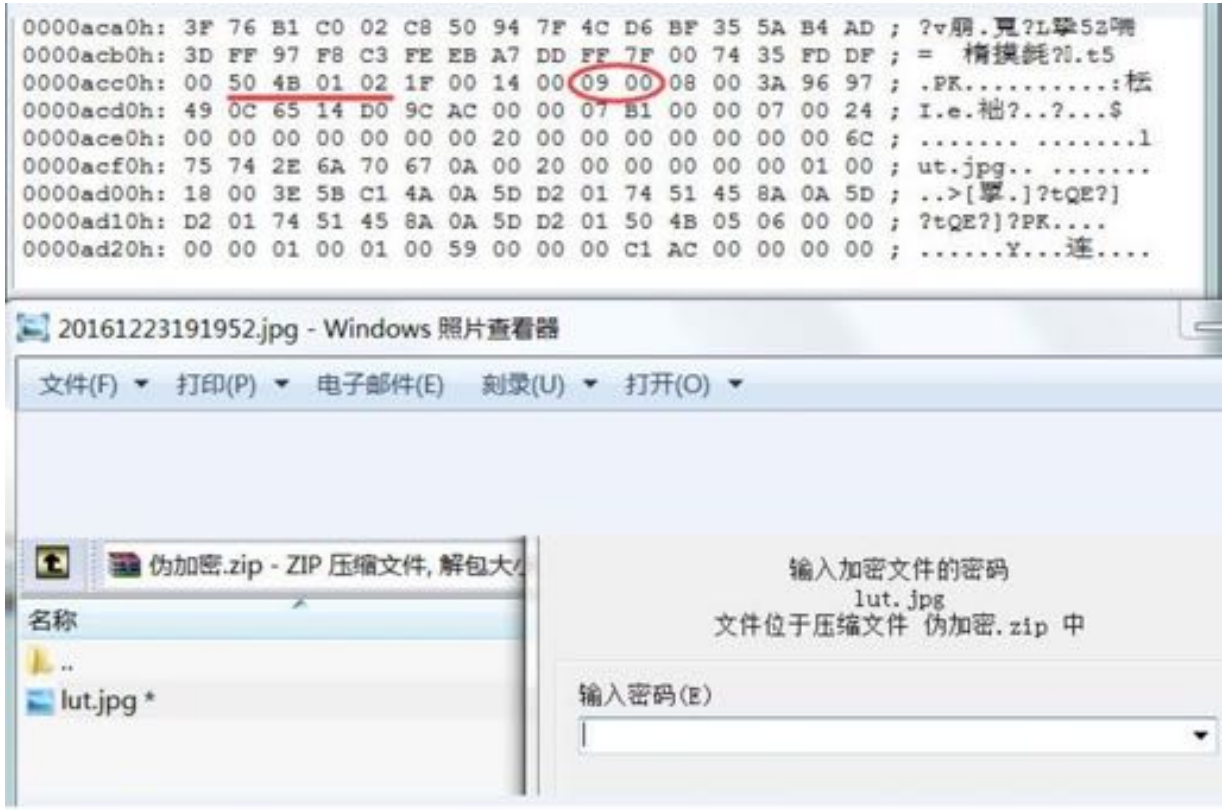
压缩文件爆破



工具：ziperello

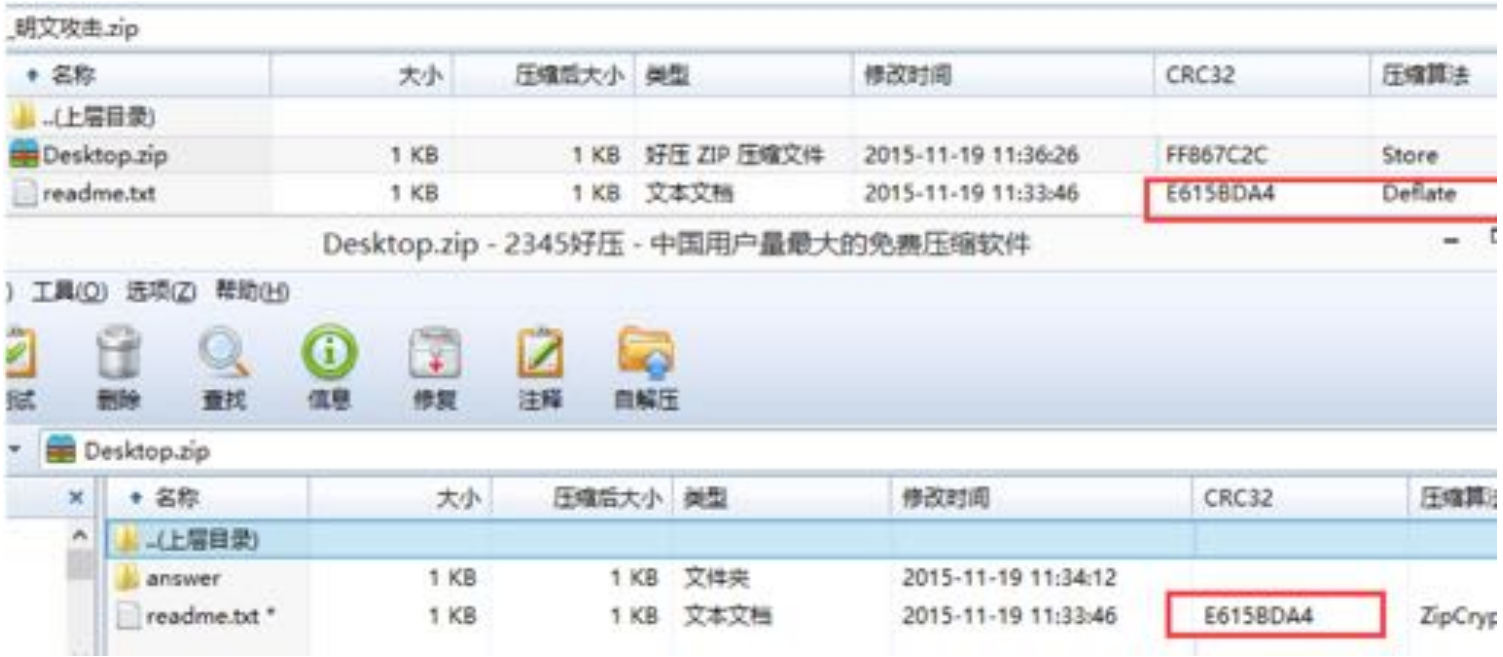
- 字符表爆破
- 字典爆破
- 掩码爆破

压缩包伪加密



- 在Mac OS及部分Linux（如Kali）系统中，可以直接打开伪加密的zip压缩包
- 使用 ZipCenOp.jar 解密
- 50 4B 01 02，在14 00 后修改回00 00即可

明文攻击



工具：

PasswareKit
Advanced zip Password Recovery

有一个文件A没有被加密，还有一个被加密的文件A以及同被加密的FLAG

需要将没有加密的压缩，然后开始进行明文攻击
注意：加密的文件A与未加密的文件A压缩算法需要一致（使用7z）

CRC32碰撞

```
#coding:utf-8
import zipfile
import string
import binascii

def CrackCrc(crc):
    for i in dic:
        for j in dic:
            for p in dic:
                for q in dic:
                    s = i + j + p + q
                    if crc == (binascii.crc32(s) & 0xffffffff):
                        #print s
                        f.write(s)
                        return

def CrackZip():
    for I in range(68):
        file = 'out' + str(I) + '.zip'
        f = zipfile.ZipFile(file, 'r')
        GetCrc = f.getinfo('data.txt')
        crc = GetCrc.CRC
        #以上3行为获取压缩包CRC32值的步骤
        #print hex(crc)
        CrackCrc(crc)

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

f = open('out.txt', 'w')
CrackZip()
f.close()
```

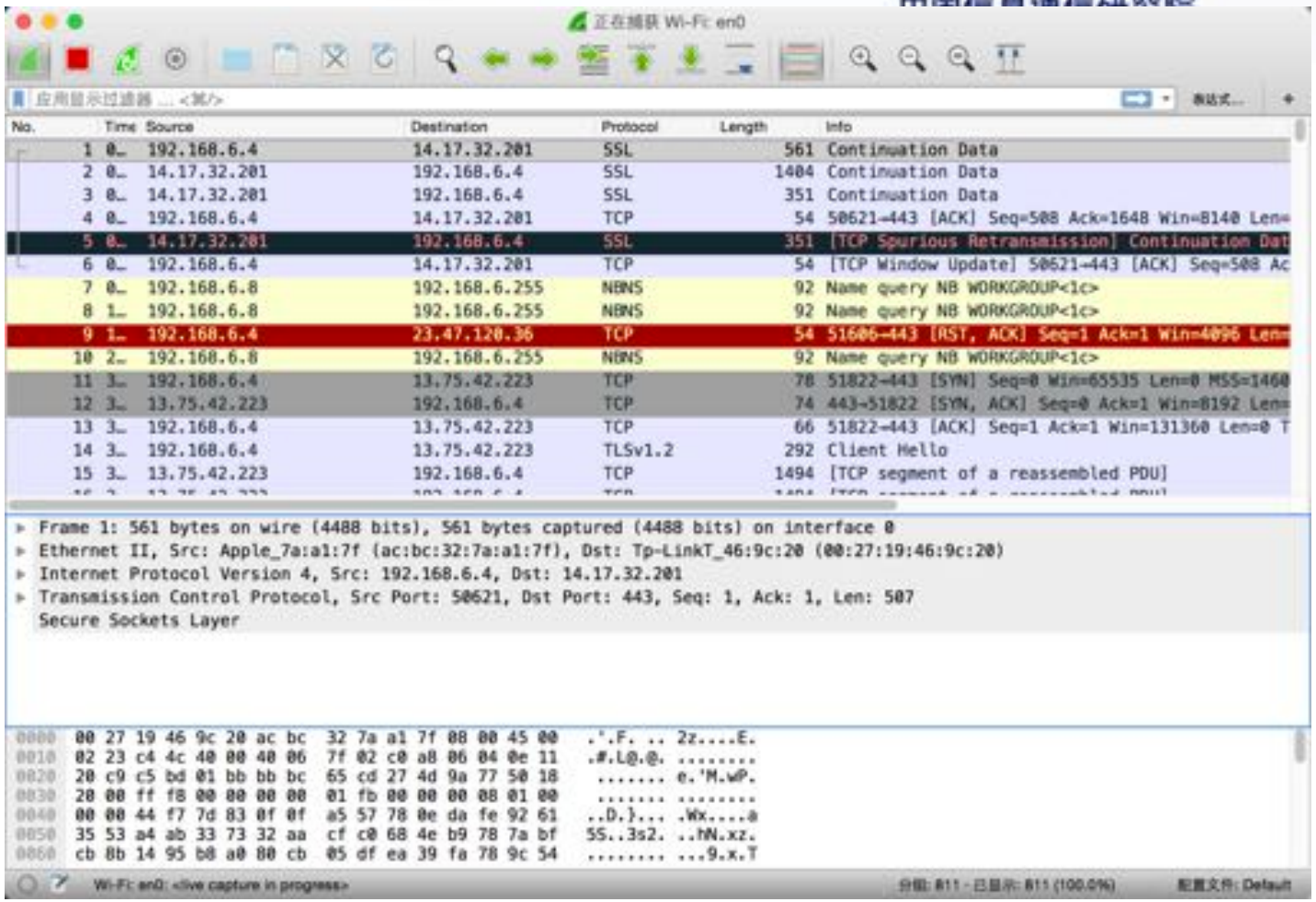
目标文件的大小比较小 / 知道大概构成（如数字组合）

就可以通过穷举组合来计算CRC值

——> 再和压缩包中文件的CRC值进行对比即可

Wireshark

- 考点
  - 从数据流中取字符串
  - 从数据流中取文件
  - 协议相关数据提取



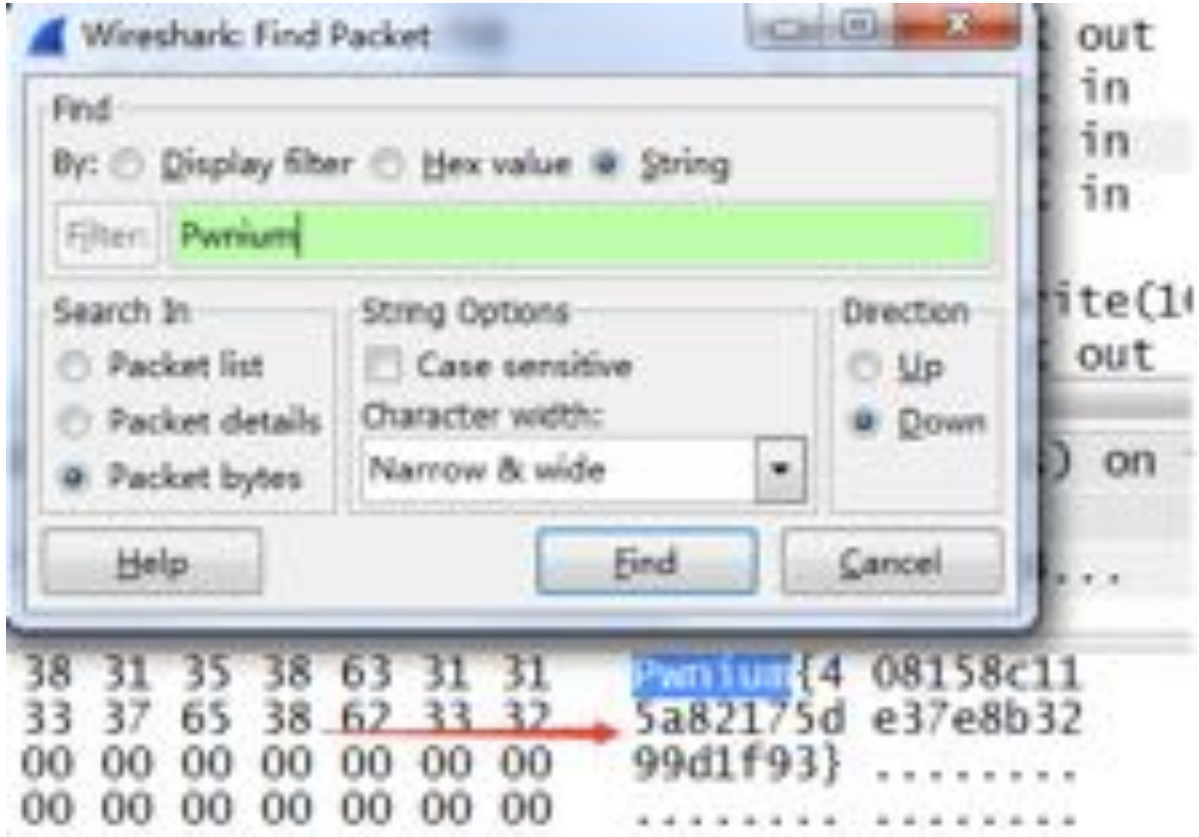
流中直接找字符串

strings 命令：

用于找出一个二进制文件中的字符、字符串

如果流中有完整的FLAG，

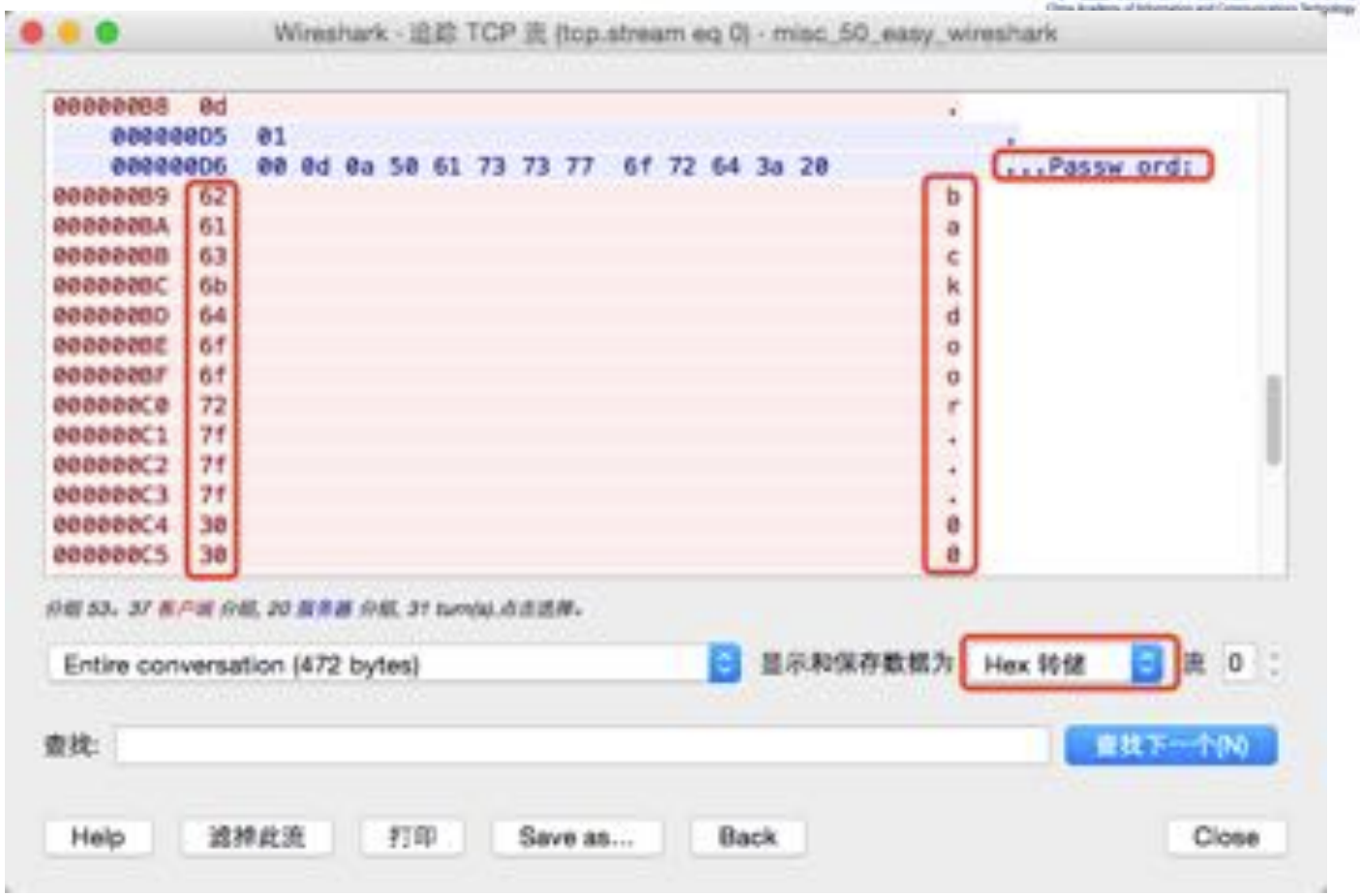
那么就可以直接使用 **strings** 命令获取FLAG



字符串提取

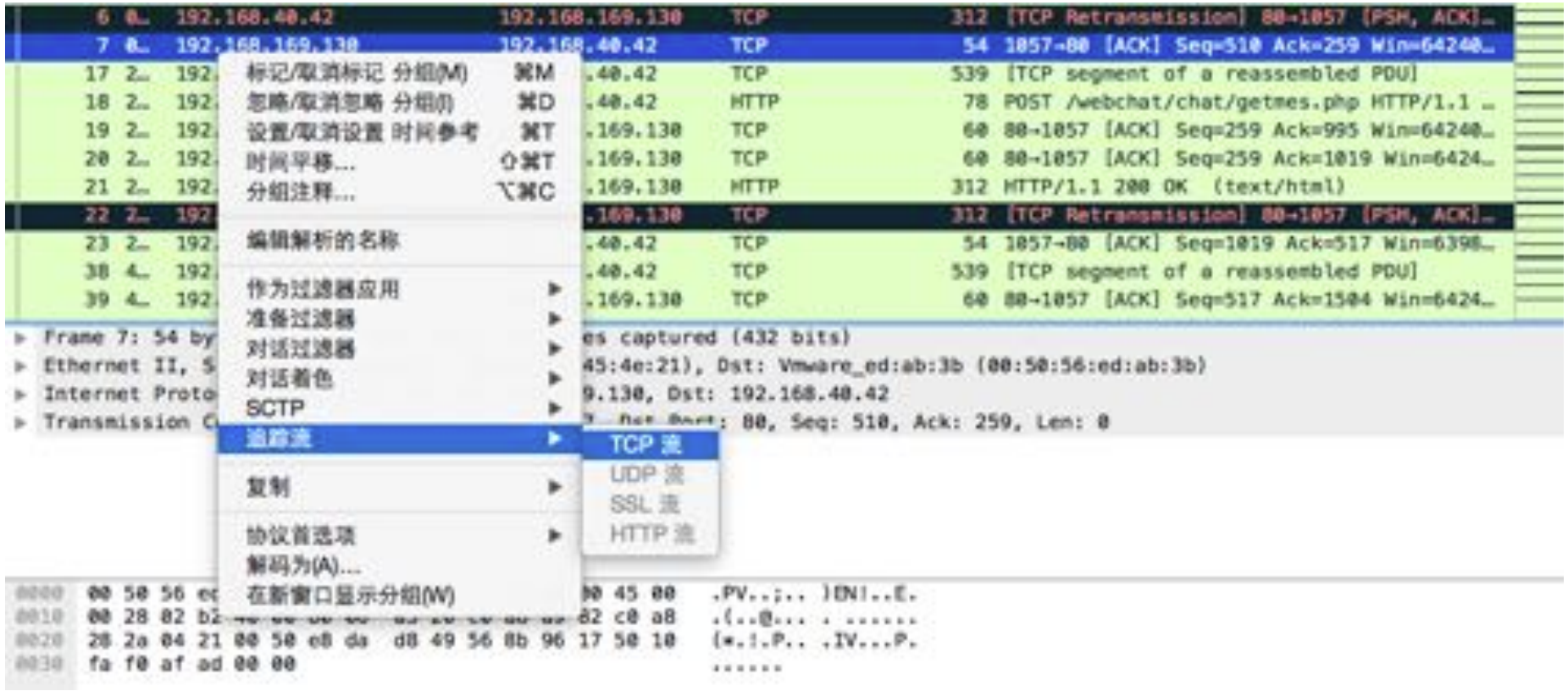
有时候还是得看看字符串的16进制究竟长什么样子

16进制是内容的最真实写照。

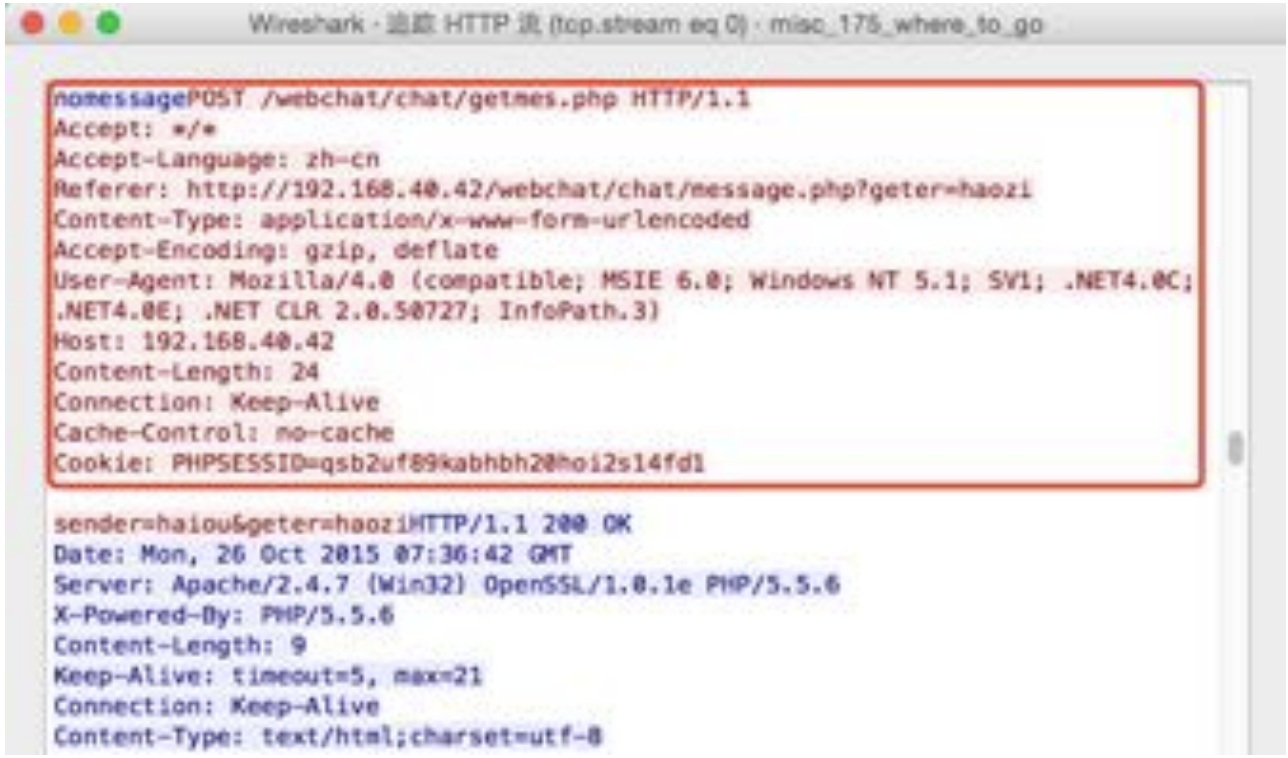


文件数据流

主要在各种用于文件传输的协议：TCP、UDP、FTP、HTTP、SAMBA等



导出文件



导出文件之后就可以继续进行文件/内容的检查。
如：一般都在其中有zip的密码之类的东西，很多东西隐藏在里面。

读懂协议——进阶网络包分析

CAICT 中国信息通信研究院

USB协议

- 键盘打字
- 鼠标绘图
- 文件传输

```
tshark.exe -r usb2.pcap -T fiel
ds -e usb.capdata > usbdata.txt
```

GitHub

- UsbKeyboardDataHacker
- UsbMiceDataHacker

Table 12: Keyboard/Keypad Page

Usage ID (Dec)	Usage ID (Hex)	Usage Name	Ref: Typical AT-101 Position	PC-AT	Mac	UNIX	Boot
0	00	Reserved (no event indicated)[9]	N/A	√	√	√	6/101/104
1	01	Keyboard ErrorRollOver[9]	N/A	√	√	√	6/101/104
2	02	Keyboard POSTFail[9]	N/A	√	√	√	6/101/104
3	03	Keyboard ErrorUndefined[9]	N/A	√	√	√	6/101/104
4	04	Keyboard a and A[4]	31	√	√	√	6/101/104
5	05	Keyboard b and B	50	√	√	√	6/101/104
6	06	Keyboard c and C[4]	48	√	√	√	6/101/104
7	07	Keyboard d and D	33	√	√	√	6/101/104
8	08	Keyboard e and E	19	√	√	√	6/101/104

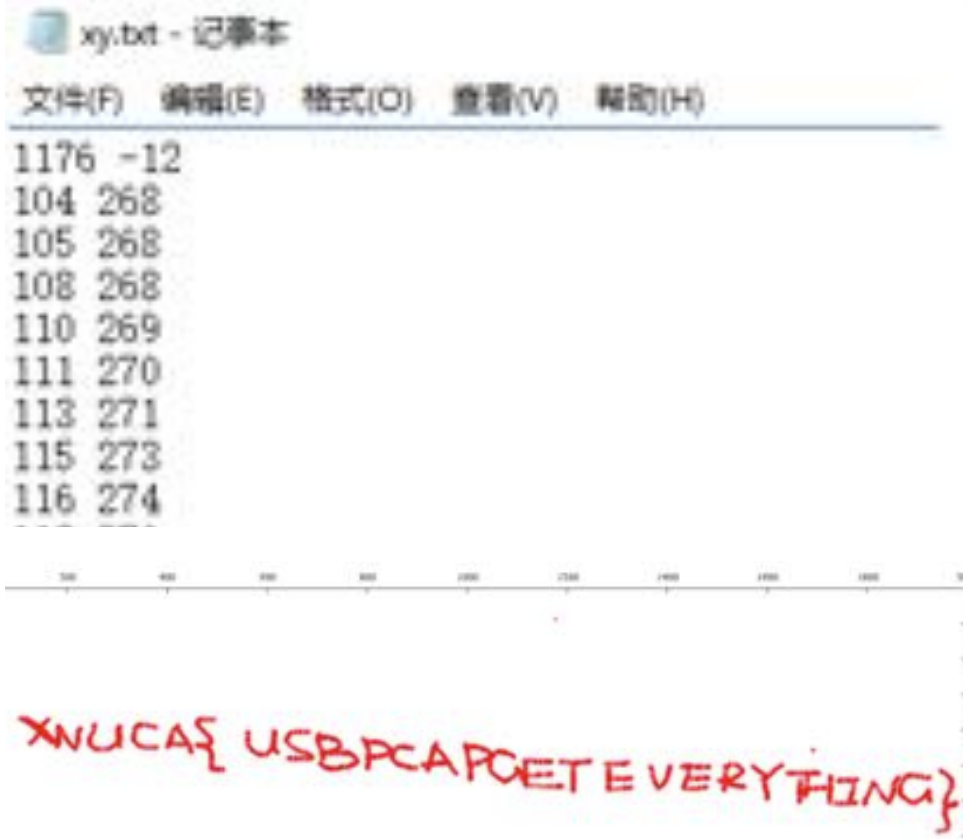
键盘对照

```
02:00:02:00
02:00:02:00
02:00:01:00
02:ff:03:00
02:00:01:00
02:00:03:00
```

USB鼠标

USB鼠标流量

```
nums = []
keys = open('data.txt','r')
posx = 0
posy = 0
for line in keys:
    if len(line) != 12 :
         continue
    x = int(line[3:5],16)
    y = int(line[6:8],16)
    if x > 127 :
        x -= 256
    if y > 127 :
        y -= 256
    posx += x
    posy += y
    btn_flag = int(line[0:2],16)
    if btn_flag == 1 :
        print posx , posy
keys.close()
```

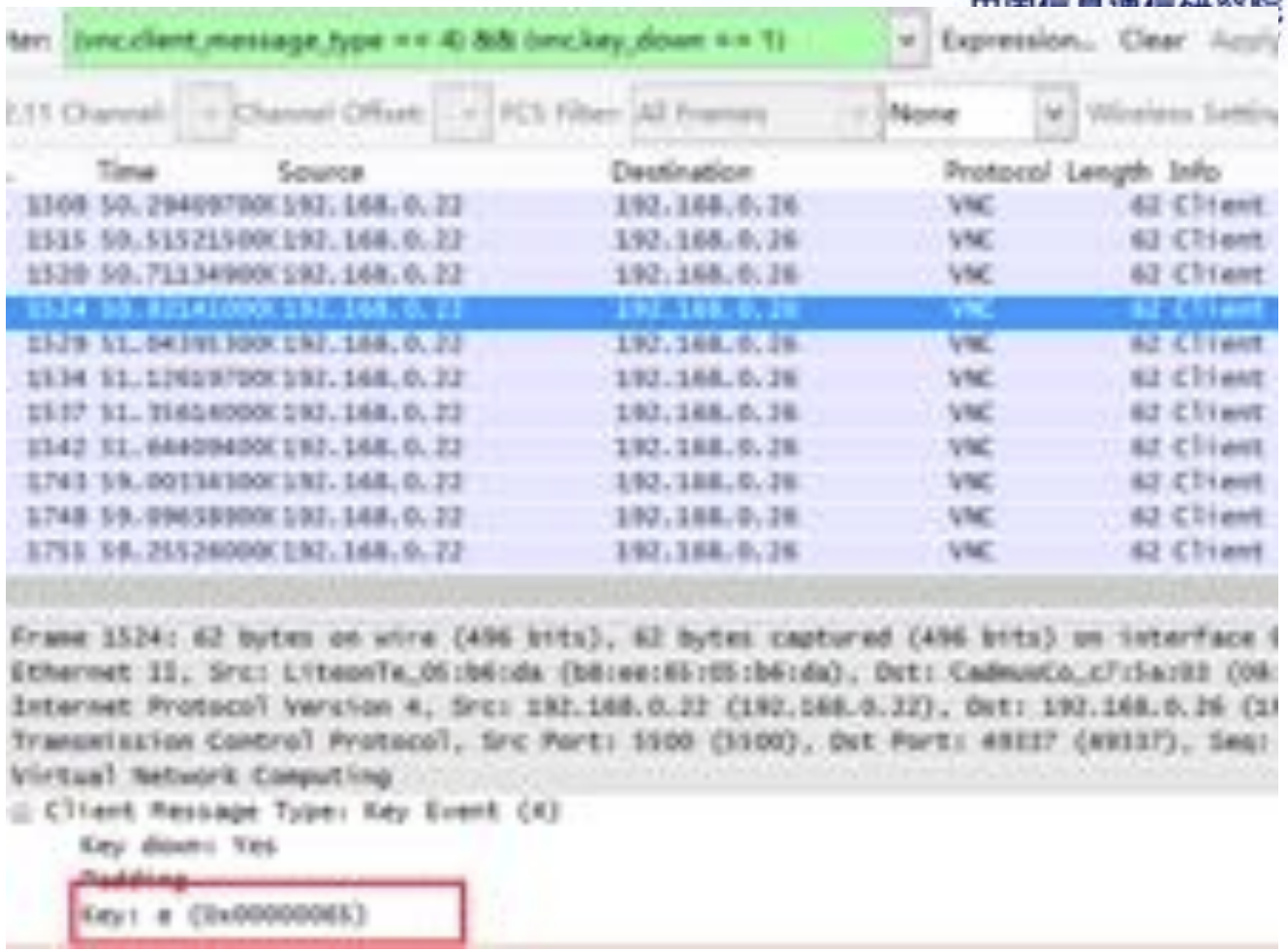


VNC流量

Key_down
-> 按下某个键

client_message_type
-> 就是按键事件

tshark -n -r forensics.pcapng 'vnc.key_down == Yes'|awk '{print $8}'|tr '\n' ' '



流量分析小结

- 流程
  - 先stirngs
  - 确定协议和事件
  - 确定IP，去除杂项
  - 关键词搜索
  - 具体分析
- 遇到陌生协议一定要仔细看
- 关键数据包都要每个字段都仔细确认

硬盘&内存取证

- 考察
  - 文件提取
  - 文件对比
  - 文件恢复
- 文件
  - VMDK 虚拟硬盘镜像
  - IMG 硬盘镜像
- 工具
  - Diskgenius
  - Virtual box
  - R-Studio
  - isoBuster

硬盘与内存取证

- 考察
  - 从内存中获取文件（图片、文档）
  - 从内存中获取秘钥
- 工具
  - Gimp
  - Volatility
  - aeskeyfind

```
root@kali:~/volatility-master# ./vol.py -f /root/dump1.raw pslist --profile=Win10x64 pstree
Volatility Foundation Volatility Framework 2.5
Offset(V)          Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
------------------ -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------------------------------
0xffffe00032553780 System                    4      0    126        0 ------      0 2016-04-04 16:12:33 UTC+0000
0xffffe0003389c040 smss.exe                268      4      2        0 ------      0 2016-04-04 16:12:33 UTC+0000
0xffffe0003381b080 csrss.exe               344    336      8        0      0      0 2016-04-04 16:12:33 UTC+0000
0xffffe000325ba080 wininit.exe             404    336      1        0      0      0 2016-04-04 16:12:34 UTC+0000
0xffffe000325c7080 csrss.exe               412    396      9        0      1      0 2016-04-04 16:12:34 UTC+0000
0xffffe00033ec6080 winlogon.exe            460    396      2        0      1      0 2016-04-04 16:12:34 UTC+0000
0xffffe00033efb440 services.exe            484    404      3        0      0      0 2016-04-04 16:12:34 UTC+0000
0xffffe00033f08080 lsass.exe               492    404      6        0      0      0 2016-04-04 16:12:34 UTC+0000
0xffffe00033ec5780 svchost.exe             580    484     16        0      0      0 2016-04-04 16:12:34 UTC+0000
0xffffe00034202280 svchost.exe             612    484      9        0      0      0 2016-04-04 16:12:34 UTC+0000
0xffffe000341cb640 dwm.exe                 712    460      8        0      1      0 2016-04-04 16:12:34 UTC+0000
0xffffe00034222780 svchost.exe             796    484     45        0      0      0 2016-04-04 16:12:34 UTC+0000
0xffffe000342a7780 VBoxService.ex          828    484     10        0      0      0 2016-04-04 16:12:34 UTC+0000
0xffffe000342ad780 svchost.exe             844    484      8        0      0      0 2016-04-04 16:12:34 UTC+0000
```

Volatility

- 查看进程信息

  volatility pslist -f *target.vmem*

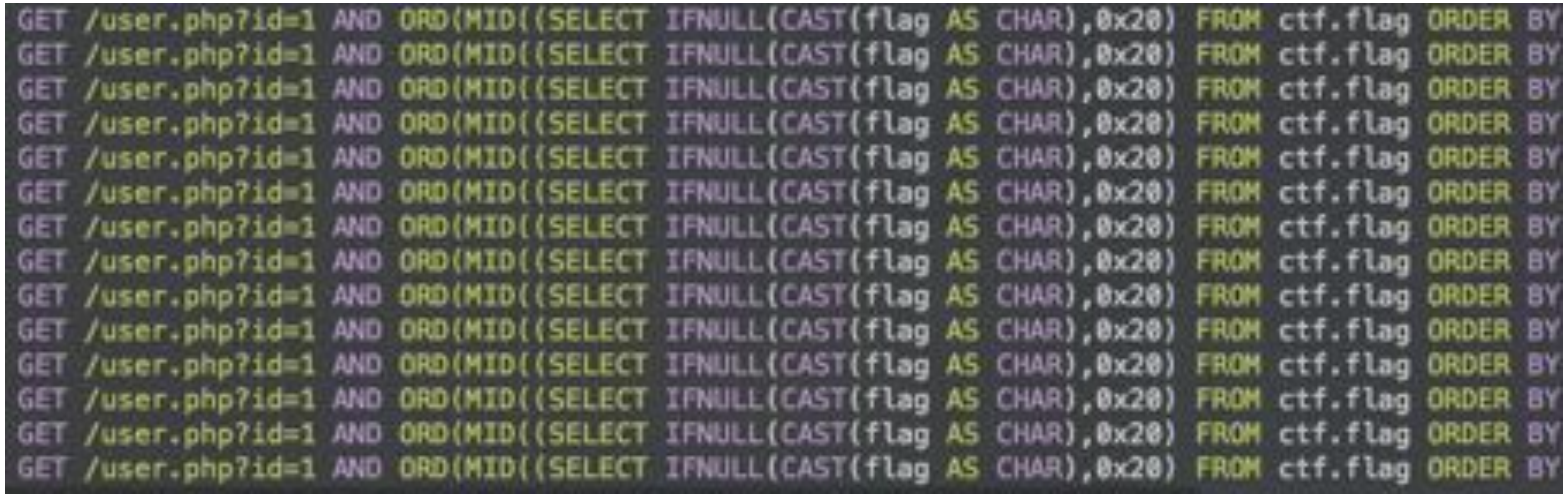
- Dump进程内存

  volatility procdump –u –memory –D ./ -p *PID* -f *target.vmem*

  volatility -f *target.raw* --profile=Win10x64 memdump -p *PID* –D ./

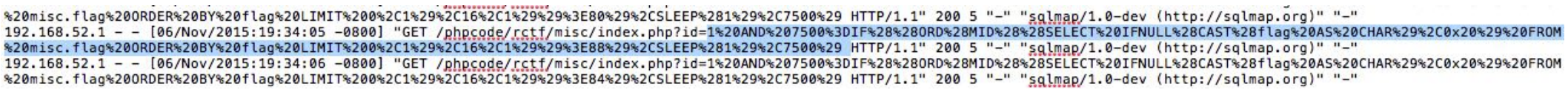
日志取证

- 考点
  - SQL注入日志分析
  - 结合web进行漏洞分析
- 文件
  - Access.log
  - Error.log



```
GET /user.php?id=1 AND ORD(MID((SELECT IFNULL(CAST(flag AS CHAR),0x20) FROM ctf.flag ORDER BY
GET /user.php?id=1 AND ORD(MID((SELECT IFNULL(CAST(flag AS CHAR),0x20) FROM ctf.flag ORDER BY
GET /user.php?id=1 AND ORD(MID((SELECT IFNULL(CAST(flag AS CHAR),0x20) FROM ctf.flag ORDER BY
GET /user.php?id=1 AND ORD(MID((SELECT IFNULL(CAST(flag AS CHAR),0x20) FROM ctf.flag ORDER BY
GET /user.php?id=1 AND ORD(MID((SELECT IFNULL(CAST(flag AS CHAR),0x20) FROM ctf.flag ORDER BY
GET /user.php?id=1 AND ORD(MID((SELECT IFNULL(CAST(flag AS CHAR),0x20) FROM ctf.flag ORDER BY
GET /user.php?id=1 AND ORD(MID((SELECT IFNULL(CAST(flag AS CHAR),0x20) FROM ctf.flag ORDER BY
GET /user.php?id=1 AND ORD(MID((SELECT IFNULL(CAST(flag AS CHAR),0x20) FROM ctf.flag ORDER BY
GET /user.php?id=1 AND ORD(MID((SELECT IFNULL(CAST(flag AS CHAR),0x20) FROM ctf.flag ORDER BY
GET /user.php?id=1 AND ORD(MID((SELECT IFNULL(CAST(flag AS CHAR),0x20) FROM ctf.flag ORDER BY
GET /user.php?id=1 AND ORD(MID((SELECT IFNULL(CAST(flag AS CHAR),0x20) FROM ctf.flag ORDER BY
GET /user.php?id=1 AND ORD(MID((SELECT IFNULL(CAST(flag AS CHAR),0x20) FROM ctf.flag ORDER BY
GET /user.php?id=1 AND ORD(MID((SELECT IFNULL(CAST(flag AS CHAR),0x20) FROM ctf.flag ORDER BY
GET /user.php?id=1 AND ORD(MID((SELECT IFNULL(CAST(flag AS CHAR),0x20) FROM ctf.flag ORDER BY
```

SQL日志分析




```
1 AND 7500=IF((ORD(MID((SELECT IFNULL(CAST(flag AS CHAR),0x20) FROM
misc.flag ORDER BY flag LIMIT 0,1),16,1)))!=83),SLEEP(1),7500)
```

关键：

- 找准关键语句，千万不要做无用功
- 好好理解日志中的内容是怎么样运作的
- 留意一下时间、IP和返回长度
- 正则很好用！！！

CAICT
中国信息通信研究院

02

隐写术

图片隐写

音频隐写

视频隐写

图片隐写

文件拼接

- 通常是使用copy拼合文件制作而成
  - copy /b 2.jpg+1.zip output.jpg
- P.S. 不单单只有jpg的拼接
- 工具： binwalk + dd + WinRAR

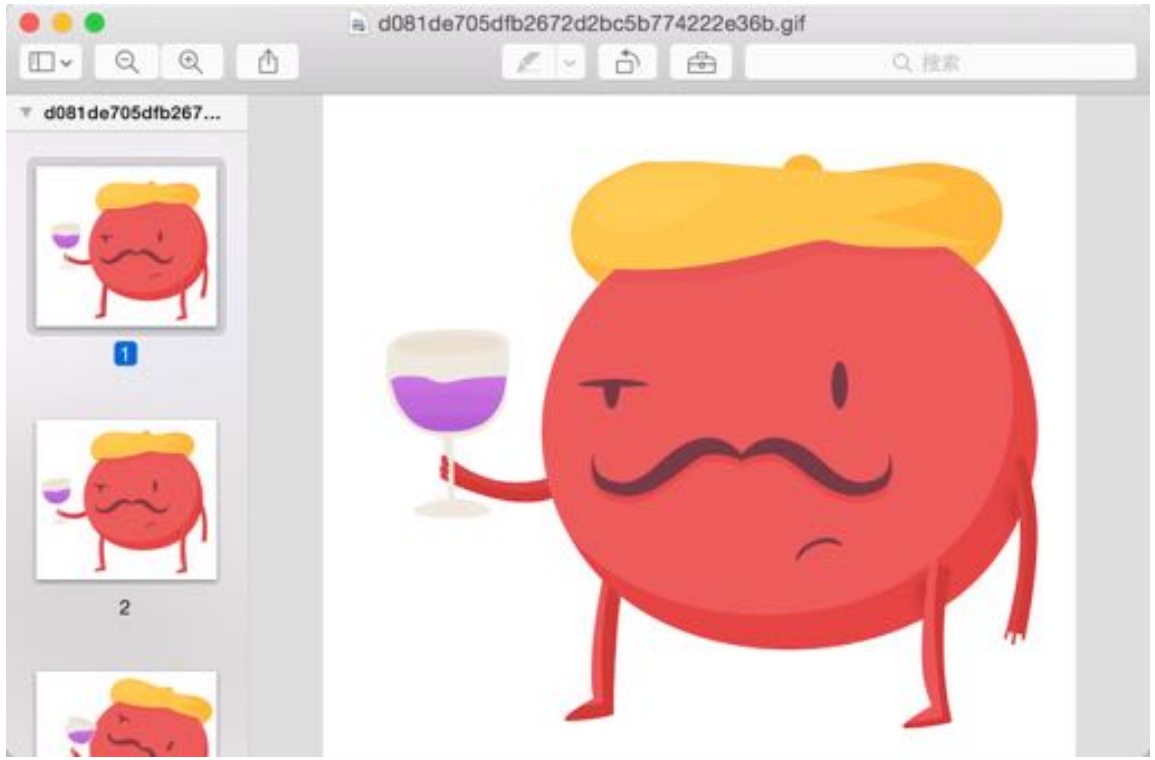
EXIF属性

- 藏在EXIF信息里
  文件属性、工具

- Tools
  Exif Pilot

GIF图

只要把内容按帧释放即可， 工具GifSplitter / MAC preview



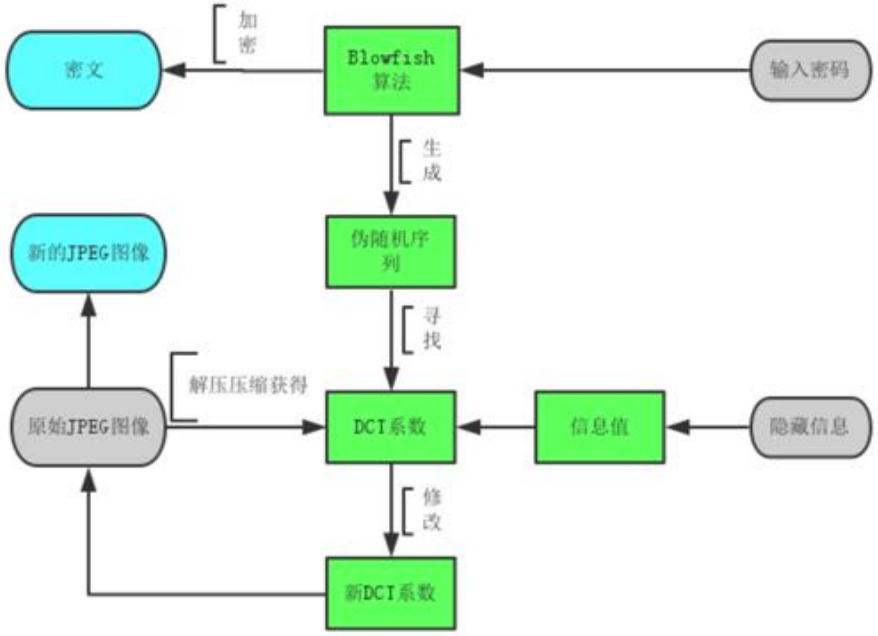
IDAT数据

- 储存图像像数数据
- 在数据流中可包含多个连续顺序的图像数据块
- 采用LZ77算法的派生算法进行压缩
- 可以用zlib解压缩

```
1  #! /usr/bin/env python
2  import zlib
3  import binascii
4  IDAT = "789C5D91011280400802BF04FFFF5C75294B5537738A21A27D1E49CFD17DB3937A92E7
   E603880A6D485100901FB0410153350DE83112EA2D51C54CE2E585B15A2FC78E8872F51C6FC188
   1882F93D372DEF78E665B0C36C529622A0A45588138833A170A2071DDCD18219DB8C0D465D8B69
   89719645ED9C11C36AE3ABDAEFCFC0ACF023E77C17C7897667".decode('hex')
5  result = binascii.hexlify(zlib.decompress(IDAT))
6  print result
```

数据加密类

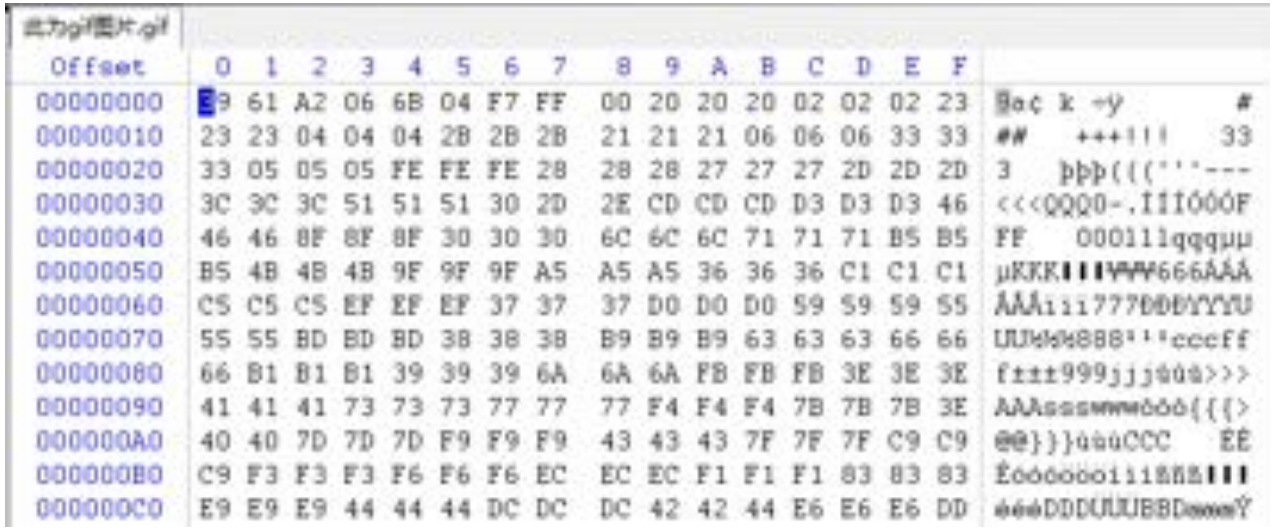
需要用到如stegdetect之类的软件进行解密



文件修复类

- 文件头 —— 文件魔数
- 文件体 —— 各部分的规范（如图片长宽、IDAT、IHDR等）
- 文件尾 —— 主要是JPEG

pngcheck -v corrupt.v2.png

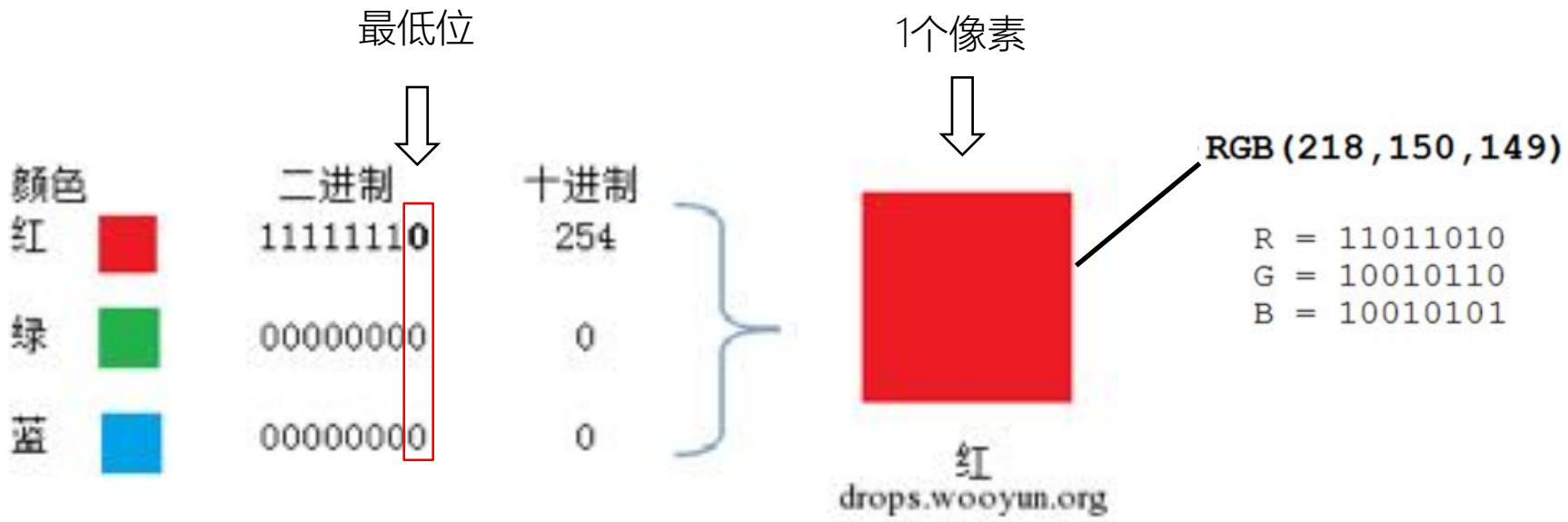


Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	E9	61	A2	06	6B	04	F7	FF	00	20	20	20	02	02	02	23
00000010	23	23	04	04	04	2B	2B	2B	21	21	21	06	06	06	33	33
00000020	33	05	05	05	FE	FE	FE	28	28	28	27	27	27	2D	2D	2D
00000030	3C	3C	3C	51	51	51	30	2D	2E	CD	CD	CD	D3	D3	D3	46
00000040	46	46	8F	8F	8F	30	30	30	6C	6C	6C	71	71	71	B5	B5
00000050	B5	4B	4B	4B	9F	9F	9F	A5	A5	A5	36	36	36	C1	C1	C1
00000060	C5	C5	C5	EF	EF	EF	37	37	37	D0	D0	D0	59	59	59	55
00000070	55	55	BD	BD	BD	38	38	38	B9	B9	B9	63	63	63	66	66
00000080	66	B1	B1	B1	39	39	39	6A	6A	6A	FB	FB	FB	3E	3E	3E
00000090	41	41	41	73	73	73	77	77	77	F4	F4	F4	7B	7B	7B	3E
000000A0	40	40	7D	7D	7D	F9	F9	F9	43	43	43	7F	7F	7F	C9	C9
000000B0	C9	F3	F3	F3	F6	F6	F6	EC	EC	EC	F1	F1	F1	83	83	83
000000C0	E9	E9	E9	44	44	44	DC	DC	DC	42	42	44	E6	E6	E6	DD

LSB原理

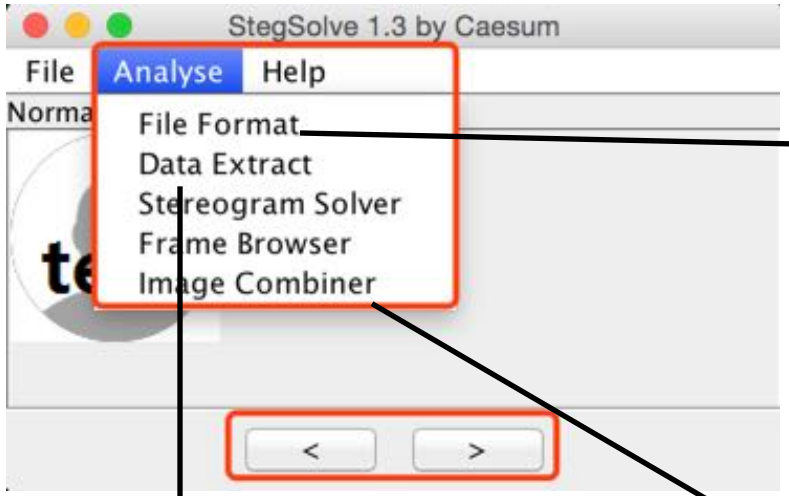
- 通过修改图片灰度、阈值、曝光度、颜色曲线等方式找到人眼识别不到的信息
- 推荐的工具：**Stegsolve.jar**
- JS库：**http://hughsk.io/lsb/**

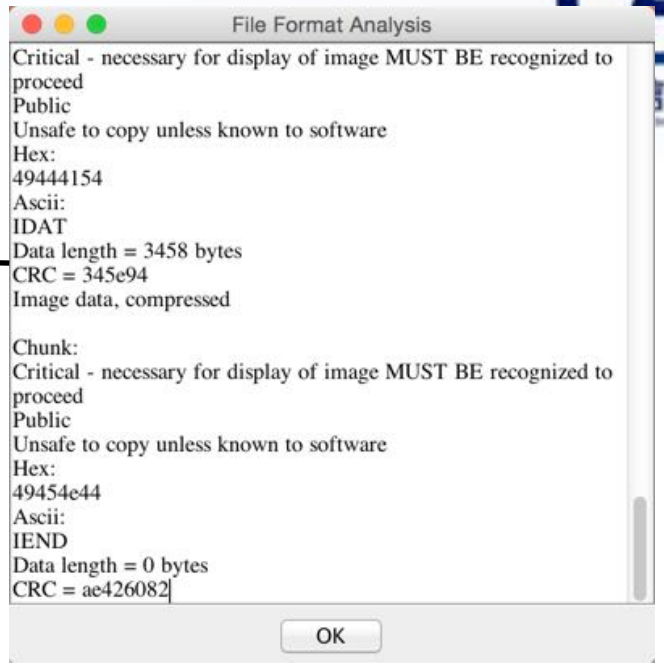
理论上可以放下**height(p)*width(p)*3**

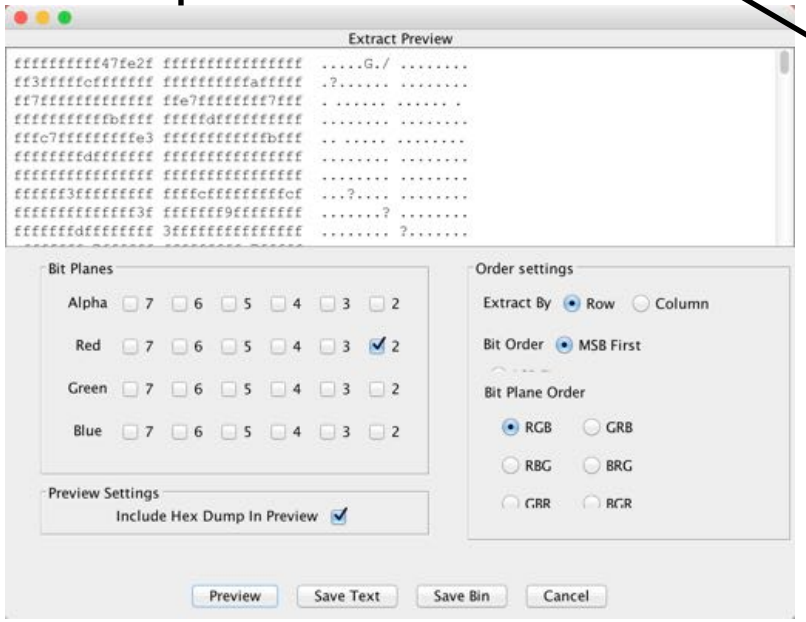


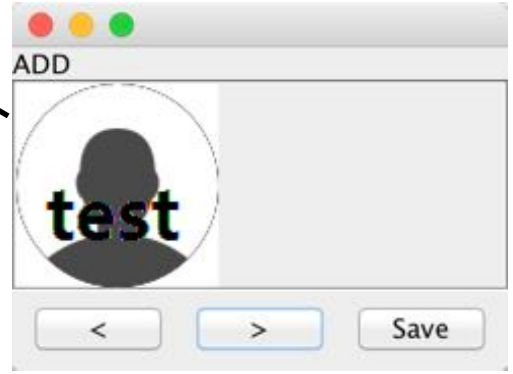
对于RGB模式

Stegsolve


StegSolve 1.3 by Caesum
File
Analyse
Help
File Format
Data Extract
Stereogram Solver
Frame Browser
Image Combiner
<
>

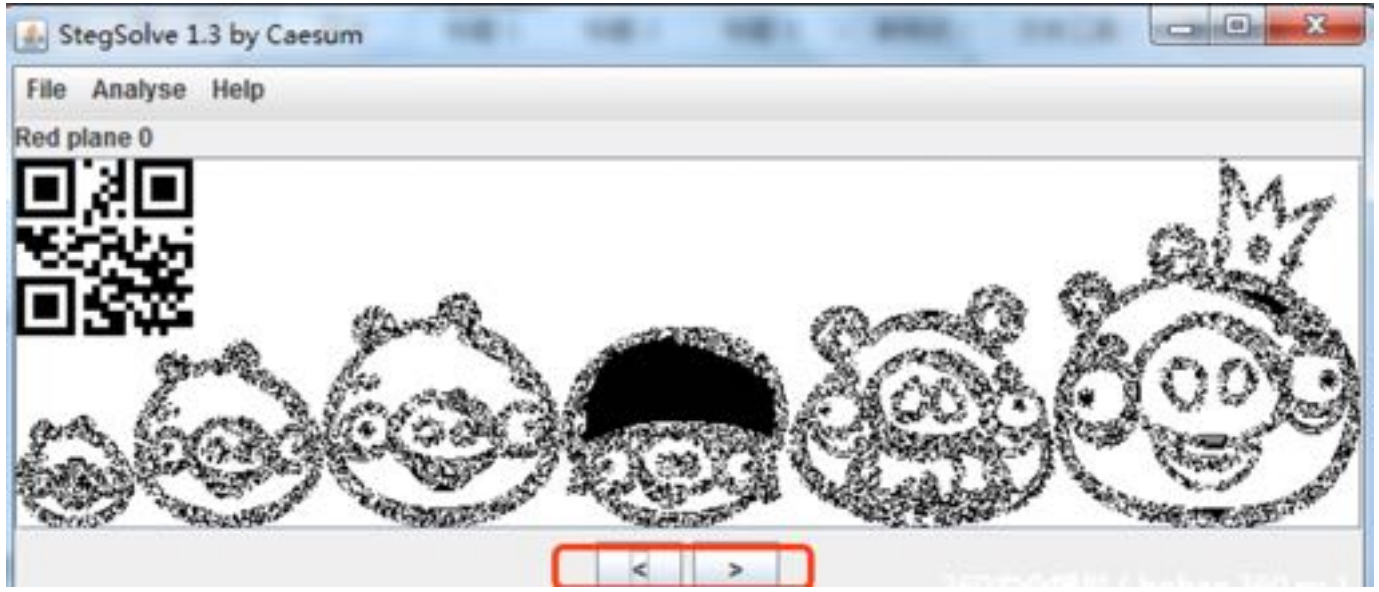

File Format Analysis
Critical - necessary for display of image MUST BE recognized to proceed
Public
Unsafe to copy unless known to software
Hex:
49444154
Ascii:
IDAT
Data length = 3458 bytes
CRC = 345e94
Image data, compressed
Chunk:
Critical - necessary for display of image MUST BE recognized to proceed
Public
Unsafe to copy unless known to software
Hex:
49454e44
Ascii:
IEND
Data length = 0 bytes
CRC = ae426082
OK


Extract Preview
Bit Planes
Alpha
Red
Green
Blue
Order settings
Extract By
Row
Column
Bit Order
MSB First
Bit Plane Order
RGB
GRB
RBG
BRG
GRR
BGR
Preview Settings
Include Hex Dump In Preview
Preview
Save Text
Save Bin
Cancel

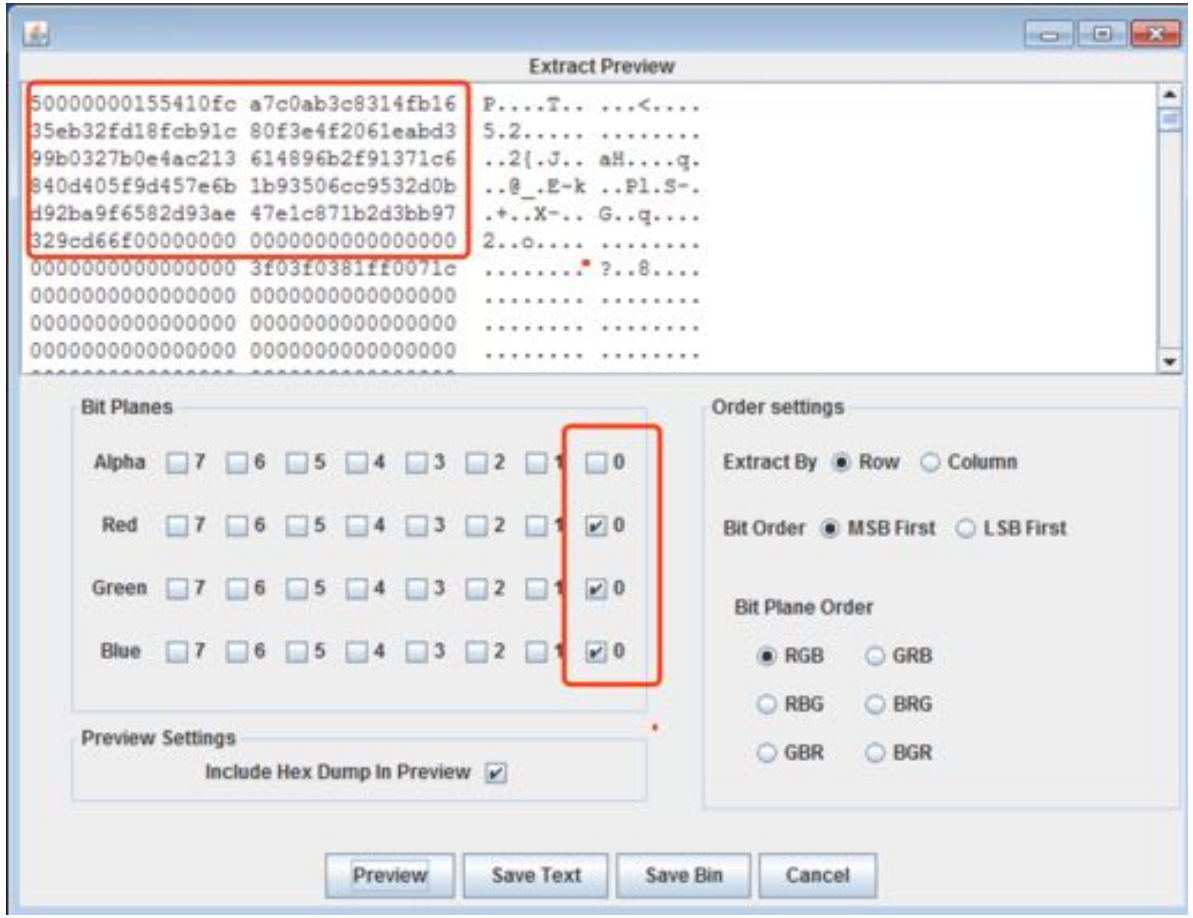

ADD
test
<
>

LSB题目


Welcome to hackgame
Enjoy it~


StegSolve 1.3 by Caesum
File Analyse Help
Red plane 0
< >

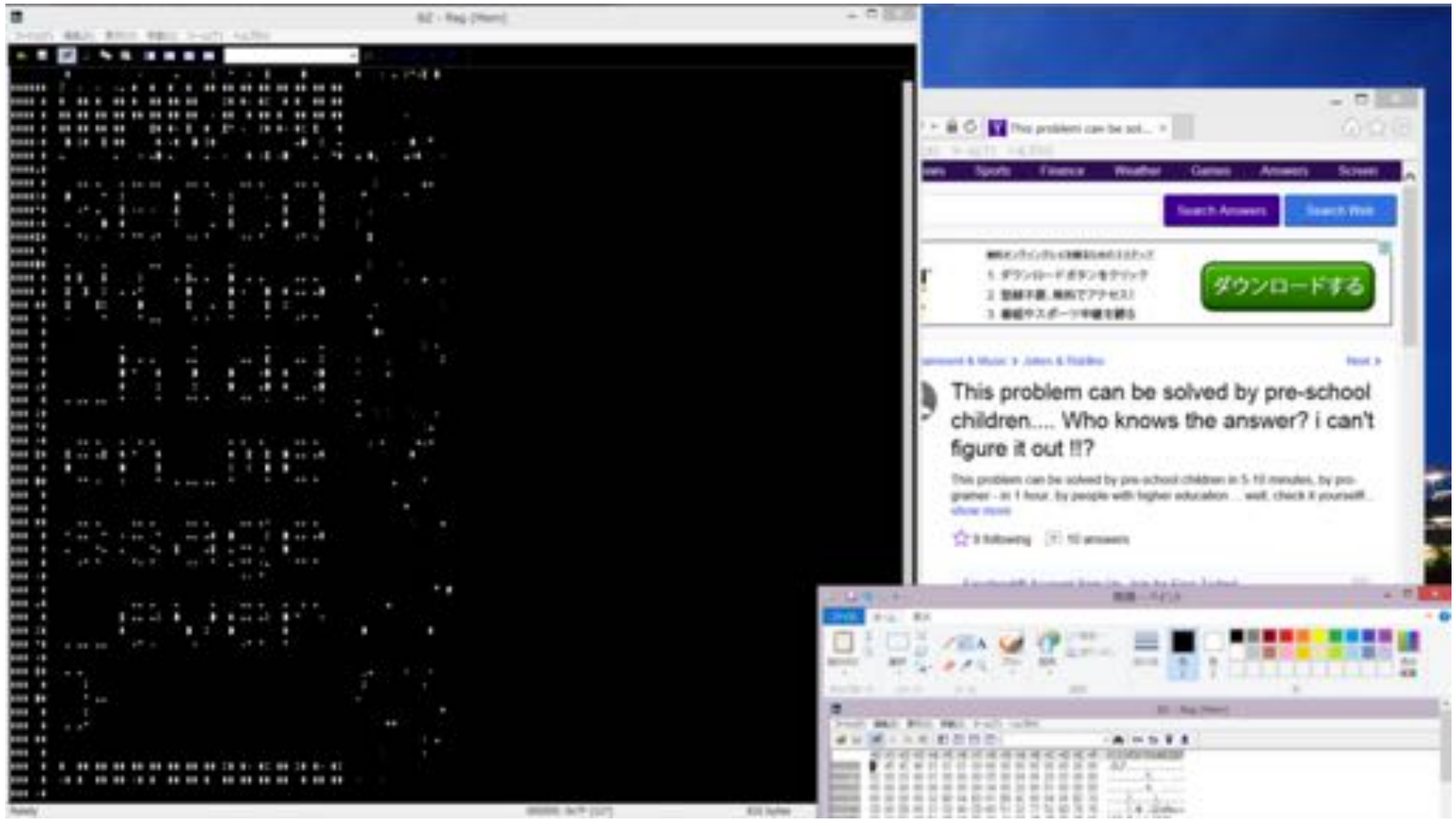
LSB题目



有可能隐写了图片，

也有可能隐写了文本

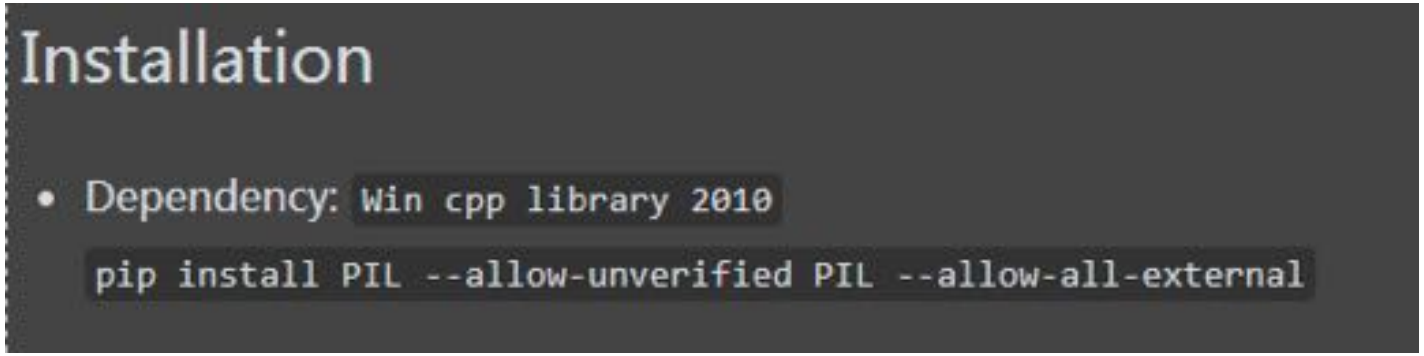
用画图做题


ダウンロードする
This problem can be solved by pre-school children.... Who knows the answer? i can't figure it out !!?

PIL图片处理

包括降噪、筛选颜色、重新排序等

- **工具：Python Imaging Library (PIL)**
  - Python 图像处理库，能够获取图像信息，进行图像编程。很好用，几乎可以替代Mathlab。



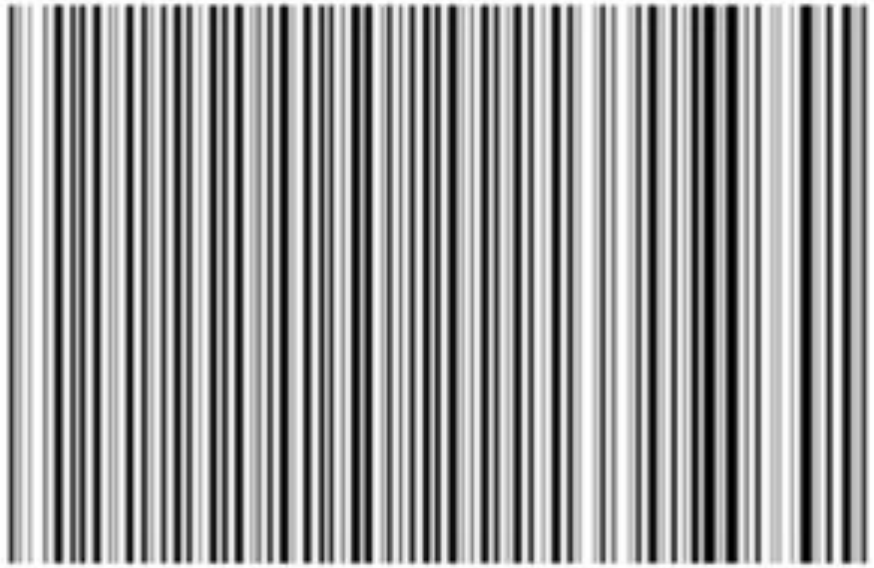
像素到图片

CAICT 中国信息通信研究院

```
255,255,255
255,255,255
255,255,255
255,255,255
255,255,255
255,255,255
255,255,255
255,255,255
255,255,255
255,255,255
255,255,255
255,255,255
255,255,255
```

```python
#-*- coding:utf-8 -*-
from PIL import Image
import re

x = 503 #x坐标  通过对txt里的行数进行整数分解
y = 122 #y坐标  x*y = 行数

im = Image.new("RGB",(x,y))#创建图片
file = open('misc100.txt') #打开rbg值文件

#通过一个个rgb点生成图片
for i in range(0,x):
    for j in range(0,y):
        line = file.readline()#获取一行
        rgb = line.split(",")#分离rgb
        im.putpixel((i,j),(int(rgb[0]),int(rgb[1]),int(rgb[2])))#rgb转化为像素
im.show()
```

图片处理



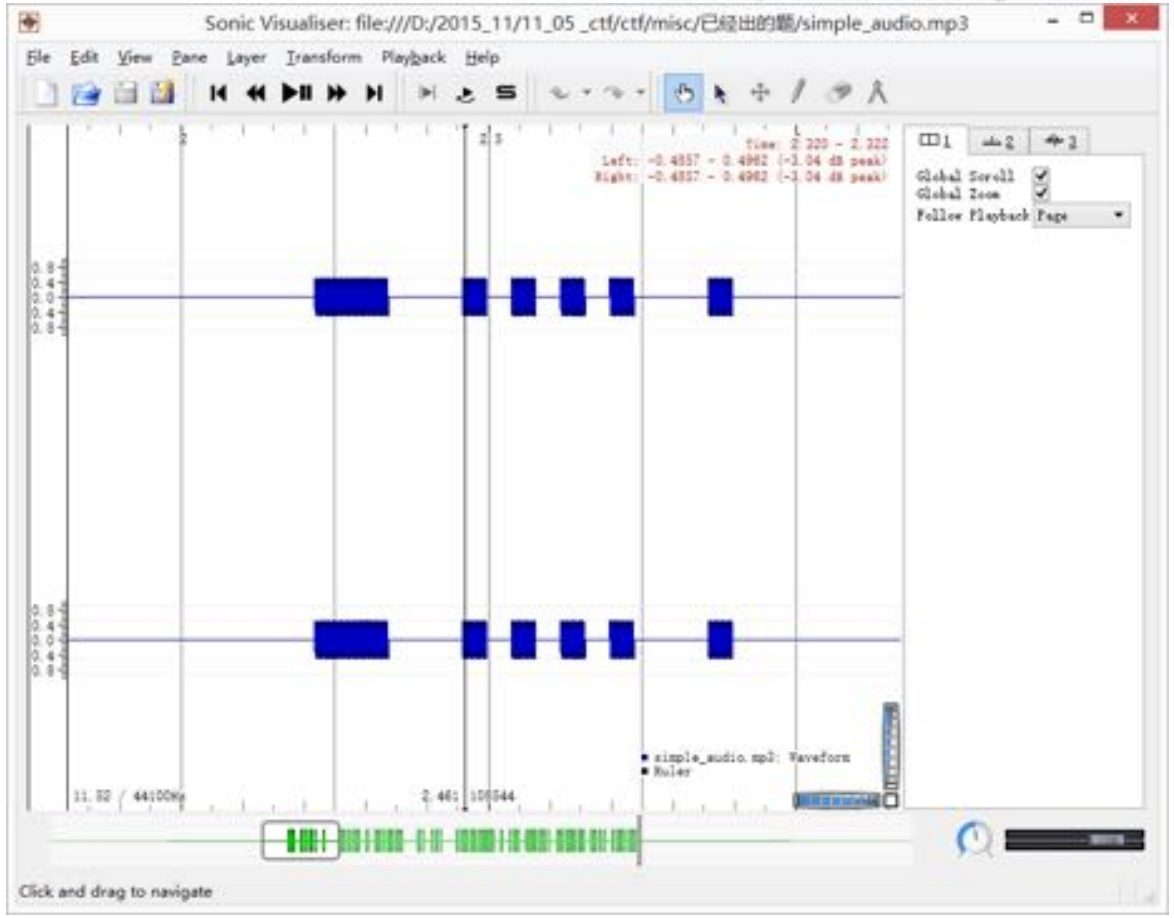
思路？

PIL总结

- 简单异或 or 加密（关键是异或 / 加密的密码）
- 根据像素进行填充补全（关键是提取像素点和位置）
- 留意像素的色差（可能构成其他图片）
- 留意像素中值相同的通道（通常都去掉）
- 降噪等高级应用

音频隐写

- 常见思路：
  - 隐藏在声音里（顺序、逆序）
  - 隐藏在数据里
  - 在声波和频谱里
- 工具：
  - Audacity
  - Matlab
  - silienteye



频谱图


2,0
- 1,0
0,0
1,0
2,0
sound
1,0
Name...
Move Track Up
Move Track Down
Waveform
Waveform (dB)
Spectrogram
Spectrogram log(f)
Pitch (EAC)
Mono
Left Channel


sound
Stereo, 44100Hz
32-bit float
Mute
Solo
L
R
22k
15k
10k
5k
0k
flag: e5353bb7b57578bd4da1c898a8e2d767
flag: e5353bb7b57578bd4da1c898a8e2d767

MP3 STEG

MP3StegoDecode.exe -P *password* -X target.mp3

视频隐写

- 主要是在metadata，或者在帧里面，帧转化成图片隐写
- 主要工具：strings/010Editor ffmpeg

视频隐写 → 图片隐写

```
ffmpeg -i 10.mp4 -an -f image2 'output_%05d.jpg'
```

CAICT
中国信息通信研究院

03

打开脑洞

数据分析

社会工程

来玩游戏

脑洞！脑洞！脑洞！

数据计算

[Misc]IPv4

[题]截止到2014.2.23，亚太互联网络信息中心分配给中国大陆的IPv4地址是多少个？

Apnic	CN	ipv4	1.2.2.0	256	20110331	assigned
等级机构	获得该IP段的国家/组织	资源类型	起始IP	IP段长度	分配日期	分配状态

```
cat delegated-apnic-20140223|grep "|CN|"|grep "ipv4" > ipv4.1.txt
awk -F "|" '{print $5}' ipv4.1.txt>ipv4.2.txt
awk '{sum+=$1;i++} END{print sum}' ipv4.2.txt
```

数据计算

BigData-100

> 一名员工通过内网web服务入侵了某台服务器，植入了webshell。
> 现在需根据该台服务器的web访问日志确认webshell事件id行数并取和。
> 例如日志中有问题的事件id行数分别为 1、3、19，那么flag即为 23 。
> 题目下载地址：点击下载

因为是webshell,一般是post请求,但是经过检查所有post并无异常。

这样,在get请求的情况下,可能会有关键字。
尝试eval,shell,hack,attack, evil,login,pass等关 键字,发现可疑文件:

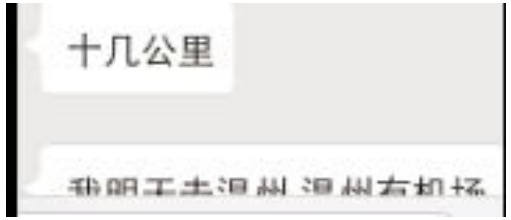
config1.php?act=attack

统计id可得答案。

社会工程学

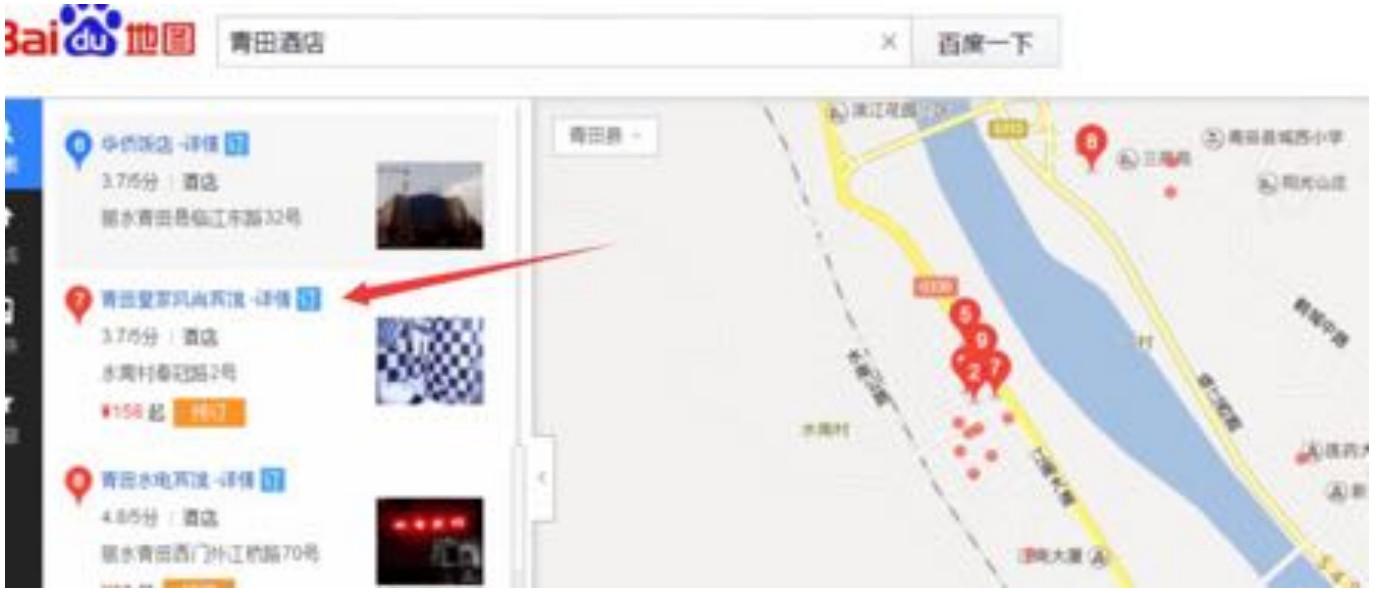
- 很多时候需要进行密码猜测，甚至是通过已有信息生成字典进行爆破

社会工程学



+

+

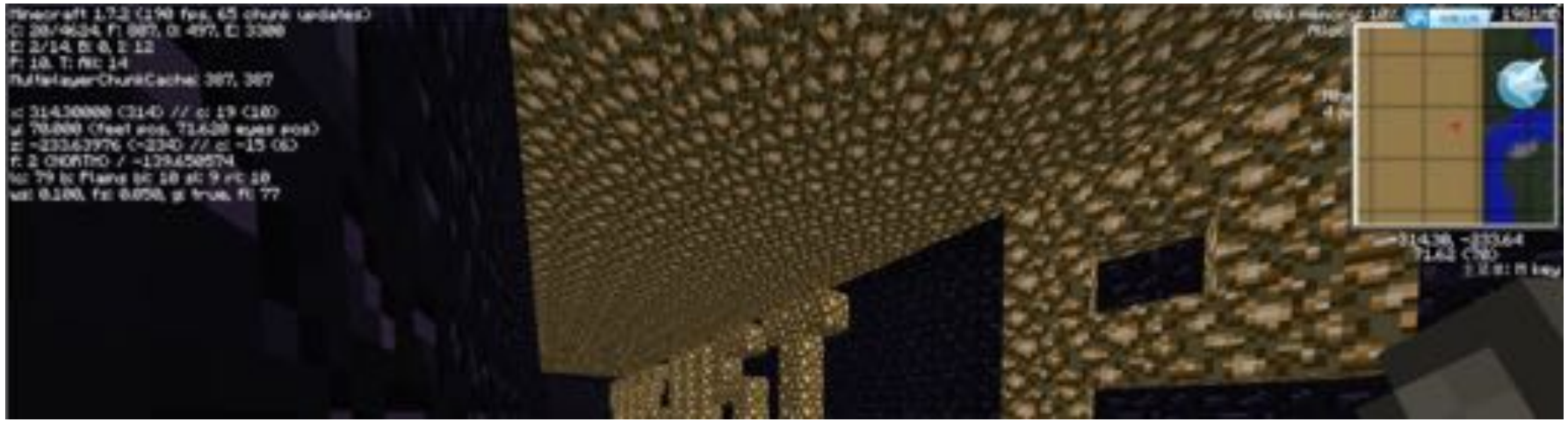


Let’s Play games

CAICT
中国信息通信研究院
China Academy of Information and Communications Technology

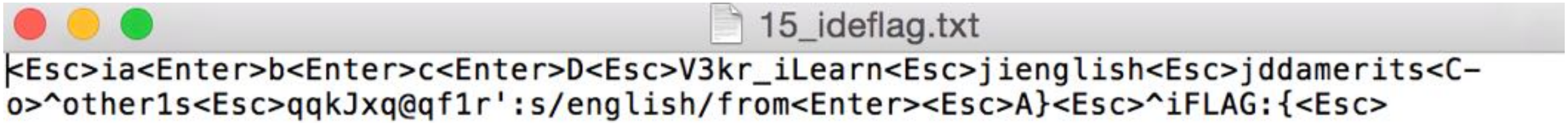
untitled • Curl.php

```
1  218.2.197.248:25565
2  the coordinate of flag(304,66,-302)
```



Let's Play games

IDE输入

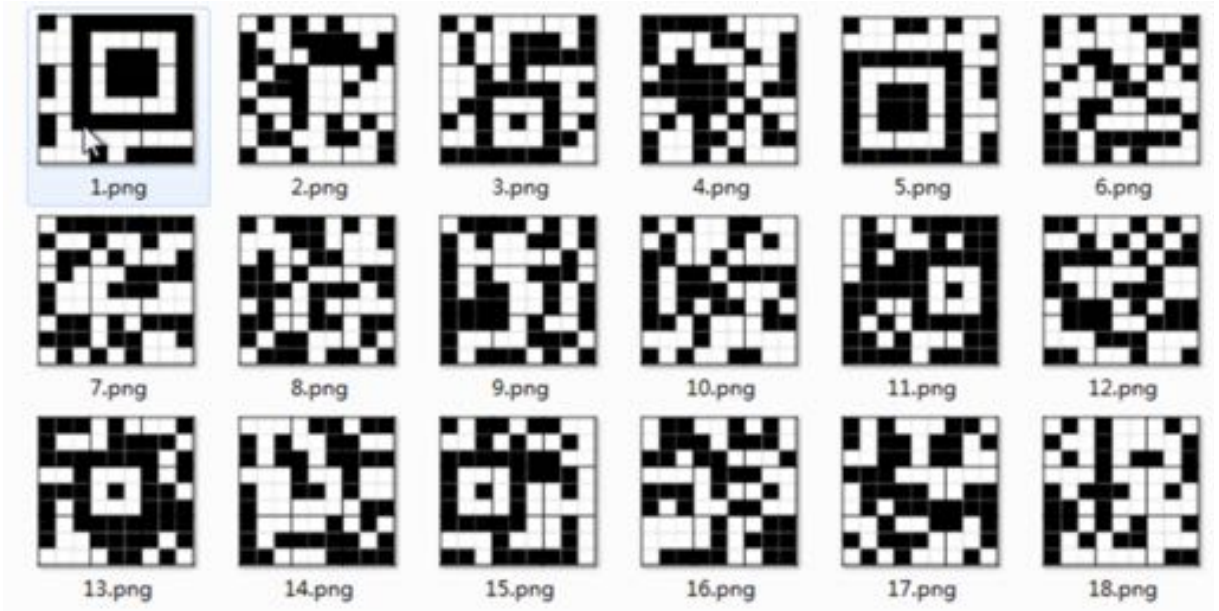

15_ideflag.txt
<Esc>ia<Enter>b<Enter>c<Enter>D<Esc>V3kr_iLearn<Esc>jienglish<Esc>jddamerits<C-
o>^other1s<Esc>qqkJxq@qf1r':s/english/from<Enter><Esc>A}<Esc>^iFLAG:{<Esc>

你的脑洞有多大？

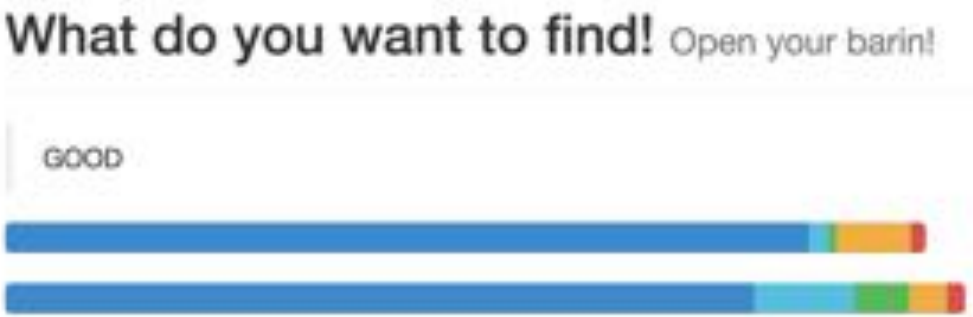
CAICT 中国信息通信研究院

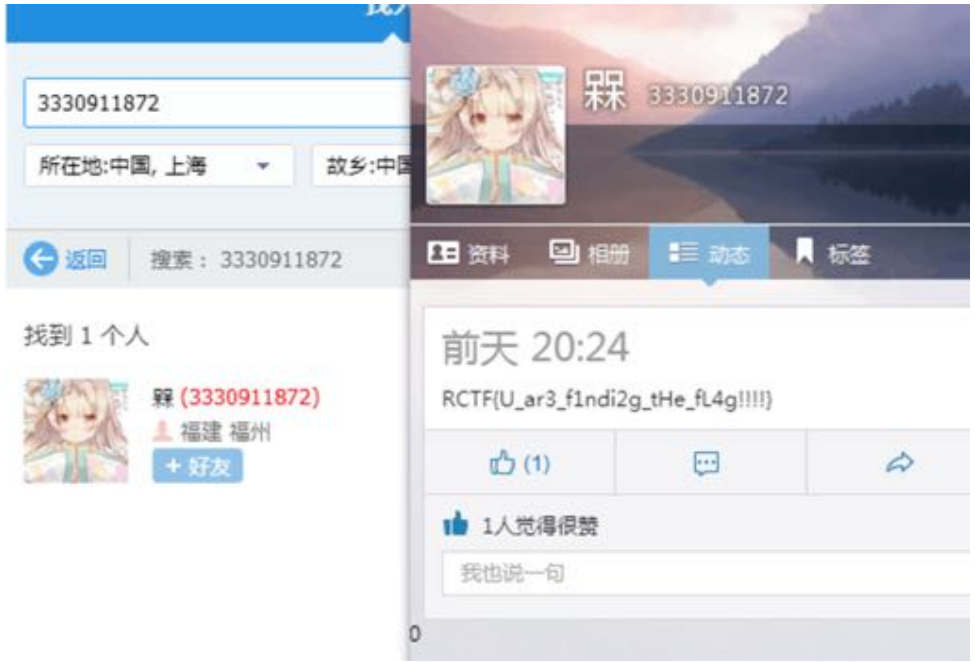
3		2	7	5	6	4	1	8
		8						6
		7		4	1	2		3
5		6		3	9	7		2
1		9		7	8	6		5
		4						1
2		3	5	8	4	1	6	9
9								
			9		7	3	5	4

PS



你的脑洞有多大？



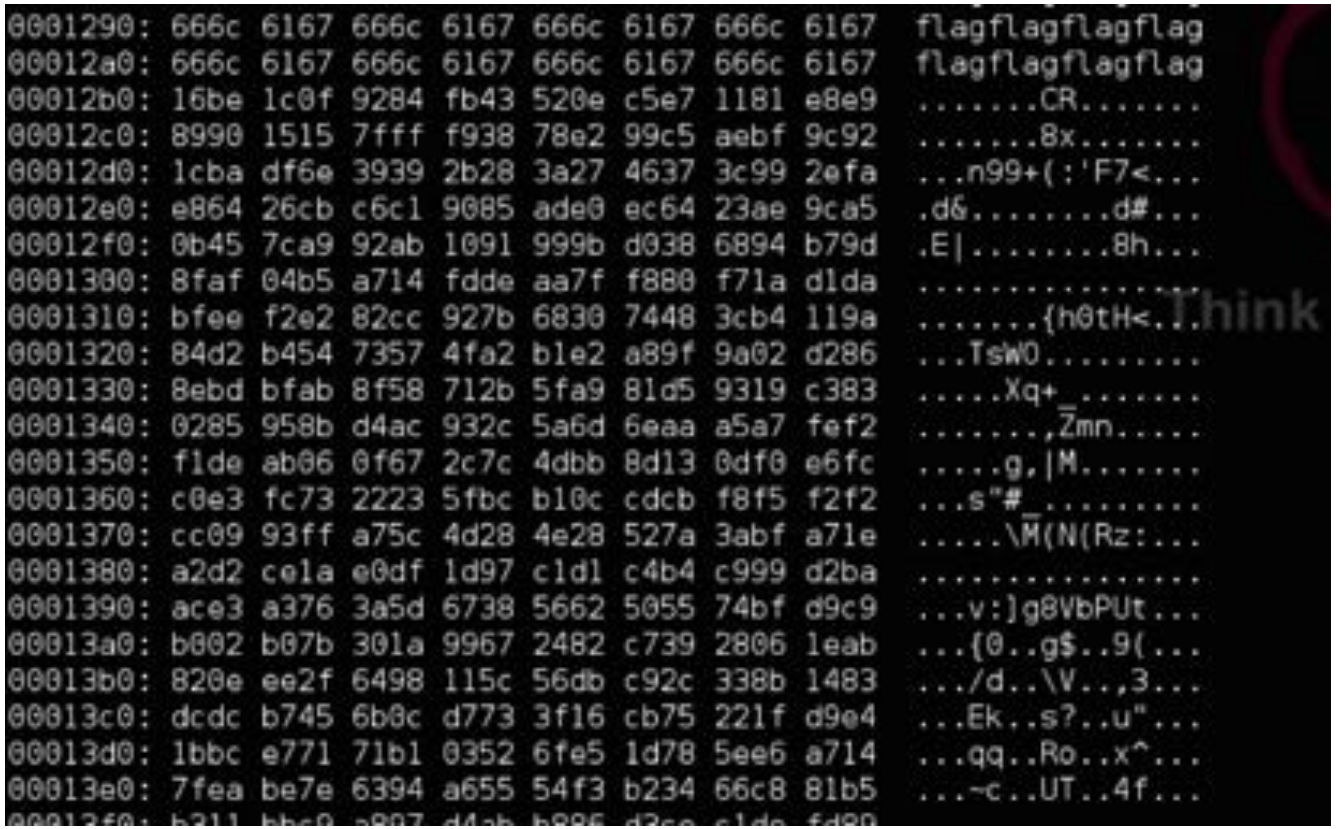


颜色值 * 宽度 求和

然后就得到 QQ

看看资料就有 flag。

16进制排序


0001290: 666c 6167 666c 6167 666c 6167 666c 6167 flagflagflagflag
00012a0: 666c 6167 666c 6167 666c 6167 666c 6167 flagflagflagflag
00012b0: 16be 1c0f 9284 fb43 520e c5e7 1181 e8e9 ....CR.......
00012c0: 8990 1515 7fff f938 78e2 99c5 aebf 9c92 ....8x.......
00012d0: 1cba df6e 3939 2b28 3a27 4637 3c99 2efa ...n99+(:'F7<...
00012e0: e864 26cb c6c1 9085 ade0 ec64 23ae 9ca5 .d&........d#...
00012f0: 0b45 7ca9 92ab 1091 999b d038 6894 b79d .E|........8h...
0001300: 8faf 04b5 a714 fdde aa7f f880 f71a d1da ....
0001310: bfee f2e2 82cc 927b 6830 7448 3cb4 119a ....{h0tH<...
0001320: 84d2 b454 7357 4fa2 b1e2 a89f 9a02 d286 ...TsWO.........
0001330: 8ebd bfab 8f58 712b 5fa9 81d5 9319 c383 .....Xq+_.......
0001340: 0285 958b d4ac 932c 5a6d 6eaa a5a7 fef2 ....,Zmn.....
0001350: f1de ab06 0f67 2c7c 4dbb 8d13 0df0 e6fc .....g,|M.......
0001360: c0e3 fc73 2223 5fbc b10c cdcb f8f5 f2f2 ...s"#_.........
0001370: cc09 93ff a75c 4d28 4e28 527a 3abf a71e .....\M(N(Rz:...
0001380: a2d2 c01a e0df 1d97 c1d1 c4b4 c999 d2ba ....
0001390: ace3 a376 3a5d 6738 5662 5055 74bf d9c9 ...v:]g8VbPUt...
00013a0: b002 b07b 301a 9967 2482 c739 2806 1eab ...{0..g$..9(...
00013b0: 820e ee2f 6498 115c 56db c92c 338b 1483 .../d..\V..,3...
00013c0: dcdc b745 6b8c d773 3f16 cb75 221f d9e4 ...Ek..s?..u"...
00013d0: 1bbc e771 71b1 0352 6fe5 1d78 5ee6 a714 ...qq..Ro..x^...
00013e0: 7fea be7e 6394 a655 54f3 b234 66c8 81b5 ...~c..UT..4f...

从入门到进(fang)阶(qi)

- 保持对世界的好奇
- 广泛的知识面
- 谷歌和度娘
- 打比赛、做题、打比赛、做题
- 脑洞、脑洞、脑洞、脑洞

CAICT
中国信息通信研究院
China Academy of Information and Communications Technology

谢谢！