CTF-MISC文件隐写总结(图片,音频,视频,压缩包等文件)

原创

OceanSec 于 2021-09-24 16:45:02 发布 1733 收藏 30

分类专栏： # CTF 文章标签： linux 运维 ctf misc 隐写

前置知识

file 命令根据文件头，识别文件类型，如果文件头前边有数据就识别不出来了

strings 输出文件中的可打印字符

可以发现一些提示信息或者特殊编码的信息

可以配合-o参数获取所有的ASCII字符偏移（字符串的位置）

binwalk命令

根据文件头识别文件

-e提取文件

foremost命令

提取文件，建议两个分离命令都试一遍

各种文件头

本地搜索：文件头

压缩包

基本思路

1. 尽量用winrar避免异常
2. 看属性
3. 伪加密
4. 暴力破解
5. 明文攻击
6. crc32碰撞
7. 多个压缩文件合并 cat 文件名(按需) > 保存文件名

文件结构

[记录文件头 + 文件结构 + 数据描述符] {此处可重复多次} + 核心目录 + 目录结束标识

暴力破解密码

仅适用于密码长度小于等于6位的，大于6位时间会非常的长

直接用apchpr爆破

zip伪加密

原理：

```
记录文件头以及核心目录区存在一个通用位标记（General purpose bit flag）的两个字节，不同的比特位有着不同的涵义

无加密：
压缩源文件数据区(ushort frFlags)的全局加密应当为 00 00,
且压缩源文件目录区的全局方式标记(ushort deFlags)应当为00 00

伪加密
压缩源文件数据区的全局加密应当为 00 00
且压缩文件目录区的全局方式标记应当为 09 00

真加密
压缩源文件数据区的全局加密应当为 09 00
且压缩源文件目录区的全局方式应当为 09 00
```

名称	值	开始
struct ZIPDIRENTRY dirEntry[4] 核心目录区	6/6/2.txt 文件名	242h
char deSignature[4]	PK	242h
ushort deVersionMadeBy	20	246h
ushort deVersionToExtract	20	248h
ushort deFlags 加密标识	9 奇数为伪加密	24Ah
enum COMPTYPE deCompression	COMP_STORED (0)	24Ch
DOSTIME deFileTime	13:33:00	24Eh
DOSDATE deFileDate	07/31/2019	250h
uint deCrc	C2FF4639h	252h

修复工具：

winrar修复(也可以修复其他的修改内容) 工具->修复

binwalk foremost无视伪加密

ZipCenOp.jar(win)

找到所在文件夹，在地址栏输入cmd

```
java -jar ZipCenOp.jar r 文件名
```

CRC32碰撞

原理：

```
压缩包的crc32的值是未加密状态计算来的
```

如果文件内容很少（4字节左右）加密的密码却很长，可以使用crc32爆破获取原文件的内容

脚本F:\CTF\CTF工具合集\脚本\CRC32碰撞\crc32-linux.py

明文攻击

基本条件

- 一个加密的压缩包文件
- 已知压缩文件的压缩工具，如winrar，7z
- 已知压缩工具的版本
- 已知压缩包文件中部分连续的内容，至少12字节，任意文件

方法

1. 将明文文件压缩成压缩包
2. 确认两者的压缩方式相同，即对比压缩之后的crc32值
3. 使用apchpr进行明文攻击

tips：

在攻击过程中密钥框如果有字符了就可以停止攻击，点击右侧使用已知密钥解密按钮

docx

包含xml文件的zip压缩包

可能隐藏文件，信息在压缩包里，在word中看不到

可以将docx后缀改为zip

图片

图片内容、图片分析、图片拼接、图片修复、EXIF、LSB

基本思路

看属性详细信息可以使用exiftool（linux）或者 identify 查看

010editor或winhex或notepad++打开看有无特殊信息，然后搜索ctf、CTF、flag、key等关键字

string、file命令(kali)

```
strings test | grep -i flag
file 1.txt
```

检查图像的开头标志和结束标志是否正确，若不正确修改图像标志恢复图像，打开查看是否有flag或ctf信息，（往往gif属于动图，需要分帧查看各帧图像组合所得数据 若不是直接的ctf或flag信息 需要考虑将其解码）

stegslove或者binwalk/foremost分离文件

注意：foremost、binwalk会根据IEND块提取png文件，IEND块后的内容会被忽略，提取不出来，可以隐藏信息

修改高度png改IHDR、jpg改ffc2（16进制搜索）三个字节后的数据

根据对应格式使用响应隐写检测工具

看图片有无异常 盲水印 、f5、Lsb、guess、stegpy、steg、jphide、stegdetect

JPG

- ### 特征

  文件头标识(2 bytes)：FF D8

  文件结束标识(2 bytes)：FF D9

- ### Lsb

- ### IDAT隐写

1. 使用pngcheck分析 pngcheck.exe -v file
2. 判断异常IDAT串，使用winhex等工具创建新文件
3. 根据创建后的新文件继续分析

- ### 修改高度

  jpg改ffc2（16进制搜索）三个字节后的数据

  stegdetect (win)

  先用这个工具检测隐写，然后再用下边的工具提取

  （**检查**jpg图片隐写方法,Stegdetect可以检测到通过JSteg、JPHide、OutGuess、Invisible Secrets、F5、appendX和Camouflage等这些隐写工具隐藏的信息）将图片复制到 **stegdetect.exe** 所在文件夹，打开 cmd 输入：

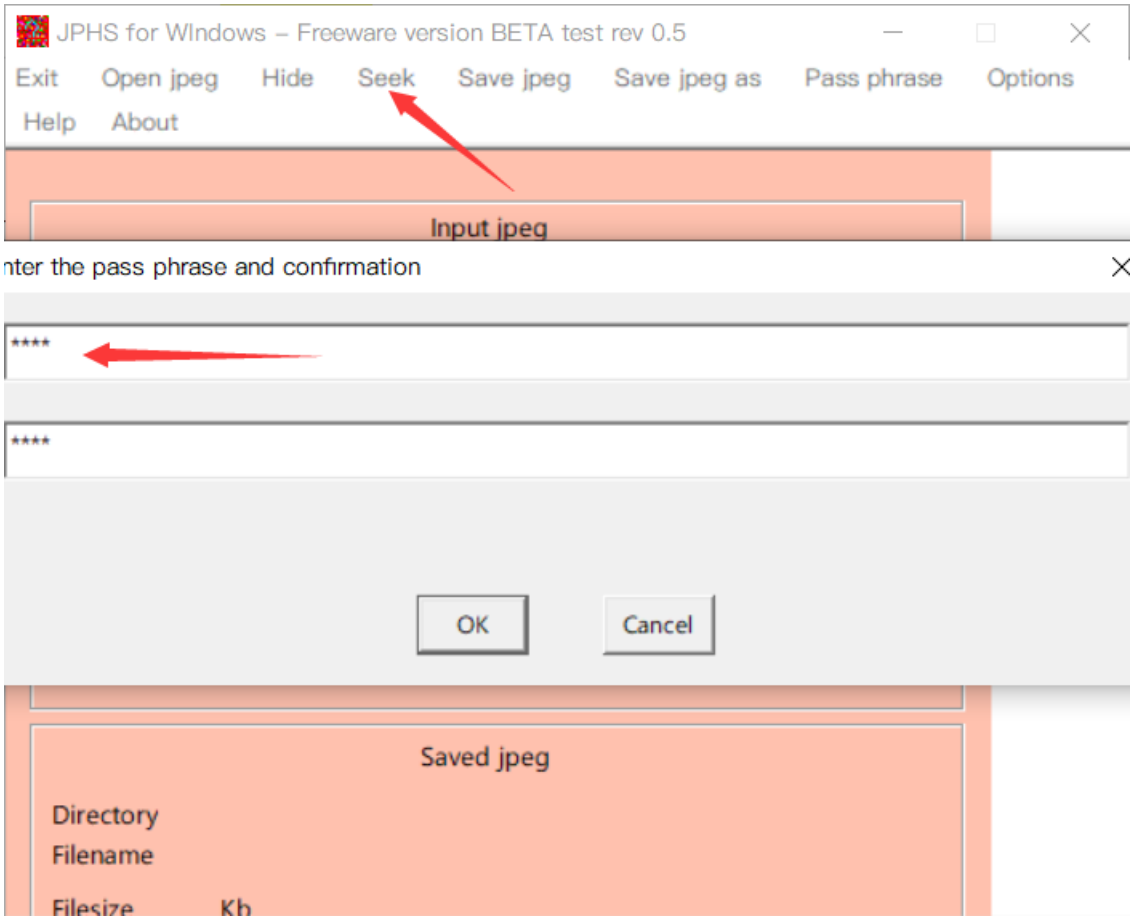
```
stegdetect.exe -tjopi -s 10.0 [stego_file]

-s 修改检测算法的敏感度，该值的默认值为1，检测结果的匹配度与检测算法的敏感度成正比，算法敏感度的值越大，检测出的可疑文件包
```

```
含敏感信息的可能性越大。
-t 设置要检测哪些隐写工具（默认检测jopi），可设置的选项如下：
 j检测图像中的信息是否是用jsteg嵌入的。
    o 检测图像中的信息是否是用outguess嵌入的。
    p 检测图像中的信息是否是用jphide嵌入的。
    i 检测图像中的信息是否是用invisible secrets嵌入的
```

jphide

用于提取 jpg 隐写的信息

图形化操作，使用工具打开文件，点击工具栏的 seek 按钮，输入密码，点击 ok 保存为 txt 文件



steghide(win)

查看图片中嵌入的文件信息：

```
steghide info out.jpg
```

提取**含有密码**的隐藏内容：

```
steghide extract -sf out.jpg -p 123456
```

提取**不含有密码**的隐藏内容：

```
steghide extract -sf out.jpg
```

steghide爆破密码

有些题目用steghide加密文件但是不给密码，此时就需要爆破，steghide本身并不支持爆破，需要一些其他的方法：

https://github.com/Va5c0/Steghide-Brute-Force-Toolpython

```
steg_brute.py -b -d [字典] -f [jpg_file]
```

需要安装的库：**progressbar**

```
pip install progressbar2
```

F5（矩阵编码）

(F5隐写，需要passwd）

在**kail**下切换到F5-steganography，在java Extract运行命令：

```
java Extract 123456.jpg图片的绝对地址 -p 123456
```

outguess（基于频率变换）

(**kali**下**图片隐写**+可需要可不要**passwd**）

```
outguess -r /root/angrybird.jpg(绝对路径) 123.txt(信息存放的文本)
outguess -k 12345 -r 2.jpg out.txt -k后接密码 -r后接解密图片 输出文件
```

win下

F:\CTF\CTF工具合集\隐写\图像隐写\F5\f5-steganography\tests

需要密码：java -jar f5.jar e -e msg.txt -p mypasswd -q 70 in.jpg out.jpg

不需要密码：java -jar f5.jar x -e out.txt pic.jpg

PNG

- ### 特征

文件头标识(8 bytes)：89 50 4E 47 0D 0A 1A 0A

文件结束：00 00 00 00 49 45 4E 44 AE 42 60 82

格式

IHDR:Header Chunk

png 数据流中第一个数据块

一个 png 数据流中只能有一个文件头数据块

png 格式内有很多 IDAT 数据块，每个数据块都是 zlib 格式压缩的，第一块 IDAT 数据块，有一个 zlib 格式的标志如：789c

idat 块只有当上一个块充满时，才会继续下一个新的块，一个图片只有一个 789c 标志

修改高度

010 editor

tweakpng.exe打开图片提示IDHRcyc错误，表示文件尺寸被修改，且未修改crc值

LSB隐写

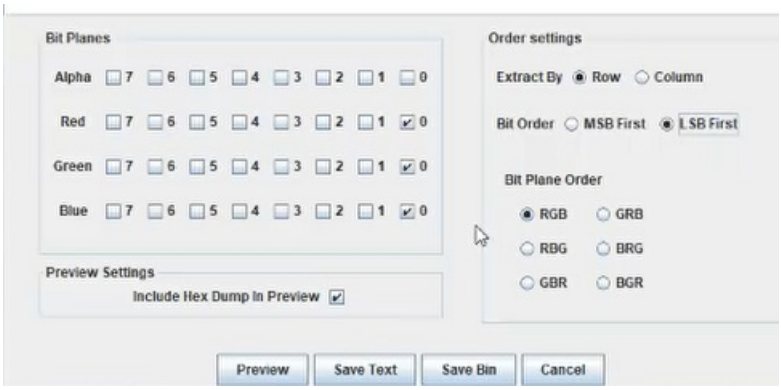
在通道的最低位隐藏数据

适用于 png 文件（无损压缩）和 pmb 文件（无压缩）

工具：stegsolve 和 zsteg

stegsolve 提取 lsb 数据

选择 extract preview，bit plane order 可以多试试，其他固定即可



- **XOR**

1.binwalk分析出**两张图片**

2.用stegslove打开选择image combiner 选择XOR

3.根据XOR后的结果继续分析

zsteg（kali）

zsteg可以检测**PNG**和BMP图片里的隐写数据（**lsb隐写、zlib、openstego等**），一般来讲用 zsteg 解密的文件都为 bmp 文件

```
zsteg 图片名
```

发现隐藏的数据，位置处于`extradata:0`

将数据提取出来：`zsteg -E "extradata:0" /home/volcano/桌面/misc17.png > 1.txt`

然后再binwalk -e把1.txt中的数据分离出来，拿到flag

BlindWaterMark（盲水印,kali）

第一种 正常的bwm

- 打开 **bwm.py**（项目地址）所在文件夹，在文件夹中打开终端

```
# 1.png 为无水印原图
# 2.png 为有盲水印的图
# flag.png 为解出来的图片
> python bwm.py decode 1.png 2.png flag.png
```

第二种 频域盲水印

```
import cv2import numpy as npimport randomimport osfrom argparse import ArgumentParserALPHA = 5def build_pars
er():parser = ArgumentParser()parser.add_argument('--original', dest='ori', required=True)parser.add_argumen
t('--image', dest='img', required=True)parser.add_argument('--result', dest='res', required=True)parser.add_
argument('--alpha', dest='alpha', default=ALPHA)return parserdef main():parser = build_parser()options = par
ser.parse_args()ori = options.oriimg = options.imgres = options.resalpha = options.alphaif not os.path.isfil
e(ori):parser.error("original image %s does not exist." % ori)if not os.path.isfile(img):parser.error("image
 %s does not exist." % img)decode(ori, img, res, alpha)def decode(ori_path, img_path, res_path, alpha):ori =
 cv2.imread(ori_path)img = cv2.imread(img_path)ori_f = np.fft.fft2(ori)img_f = np.fft.fft2(img)height, width
 = ori.shape[0], ori.shape[1]watermark = (ori_f - img_f) / alphawatermark = np.real(watermark)res = np.zeros
(watermark.shape)random.seed(height + width)x = range(height / 2)y = range(width)random.shuffle(x)random.shu
ffle(y)for i in range(height / 2):for j in range(width):res[x[i]][y[j]] = watermark[i][j]cv2.imwrite(res_pat
h, res, [int(cv2.IMWRITE_JPEG_QUALITY), 100])if __name__ == '__main__': main()
```

使用

```
python pinyubwm.py --original 1.png --image 2.png --result out.png
```

查看 **out.png** 即可，如果无法得到正常图片，可将 **1.png** 和 **2.png** 调换位置再次尝试

lsb的py脚本解密（lsb隐写+需要passwd）

F:\CTF\CTF工具合集\脚本\cloacked-pixel-master

使用

```
python lsb.py extract [stego_file] [out_file] [password]
```

pngcheck（检查IDAT块_win）

在 **pngcheck.exe** 所在文件夹打开cmd

```
pngcheck.exe -v 123.png
```

可检查 **png** 的 **IDAT** 块是否有问题相关题目可参考： https://blog.csdn.net/u010391191/article/details/80818785

有关解题脚本可参考 **FzWjScJ** 师傅的blog： http://www.fzwjscj.xyz/index.php/archives/17/

java 盲水印

使用工具提取 F:\CTF\CTF工具合集\隐写\图像隐写

```
 java -jar .\BlindWatermark.jar decode -c .\flag.png output.png
```

apng图片

实际是动图，使用apngdis.exe进行分离

如果用tweakpng打开发现很多报错，就用pngdebuger看报错是不是藏了啥

GIF

特征

文件头标识(6 bytes)：47 49 46 38 39(37) 61 即GIF89a

动图，多张图片组成，有多个元数据（属性信息）

时间轴隐写，利用多帧之间的时间间隔不同，来隐藏信息（大多为二进制 01 ）

BMP

特征

无损，无压缩格式

LSB隐写

频域盲水印隐写

使用 matlab

WebP

安装（kali中）apt install webp需要的时候按 Y 即可

使用

cwebp - 编码器工具：可将png转为webp

```
cwebp 1.png -o 2.webp
```

dwebp - 解码器工具：可将webp转为png

```
dwebp 1.webp -o 2.png
```

vwebp - 查看器工具：可直接查看webp格式图片

```
vwebp 1.webp
```

webpinfo - 格式查看工具：可打印出WebP文件的块级结构以及基本完整性检查

```
webpinfo 1.webp
```

其余（gif2webp、img2webp等可见 官方文档）

exiftool（查看图片exif信息）

```
exiftool 1.jpg # 显示图片所有信息
exiftool 1.jpg | grep flag # 查看图片有关‘flag’字符的信息
exiftool * #查看此文件夹所有图片信息
exiftool -b -ThumbnailImage attachment.jpg >flag.jpg # 提取缩略图*
```

BPG

BPG (新的图像格式)

编辑 讨论

本词条缺少概述图，补充相关内容使词条更完整，还能快速升级，赶紧来编辑吧！

BPG（Better Portable Graphics，更好的可移植图形）是一种新的图像格式。它的目的是在质量或文件大小有问题时替换jpeg图像格式。

中文名	更好的可移植图形	缩　写	BPG
外文名	Better Portable Graphics	同类项	JPEG等图片格式

BPG（更好的可移植图形）是一种新的图像格式。它的目的是在质量或文件大小有问题时替换jpeg图像格式。其主要优点是：

高压缩比。对于类似的质量，文件比jpeg小得多。

文件头：425047FB

在线网站查看即可：https://webencoder.libbpg.org/show.html

工具脚本使用

Pillow

pil 升级版，下边是简单使用

```
from PIL import Image
# 打开一个png图像文件，注意是当前路径:
im = Image.open("file.png")
# 获得图像尺寸:
w, h = im.size
im.show()
# 把图像用png格式保存:
im.save("another.png")
# 图像处理-改变尺寸
im.resize((32,32))
# 图像处理-旋转
im.rotate(90)

# 像素处理-将图像像素数组方式读取(整体)
im.getdata()
# 像素处理-数组生成图像(整体)
im.putdata()
# 像素处理-读取单个像素的rgb值(单个像素)
im.getpixel((x,y))
# 像素处理-读取单个像素的rgb值(单个像素)
im.putpixel((x,y),(0,0,0))

im.copy()
im = Image.new("RGB",(1,1),data)
```

二进制数据生成二维码

根据二进制字符串的长度，开方确定图片尺寸

```
from PIL import Pillow
s = "二进制字符串"
im = Image.new("RGB",(25,25))
im.putdata([(255,255,255) if i=="1" else (0,0,0) for i in s])
im.resize((100,100)).show()
```

identify

功能和exiftool相同，显示信息更多，用法如下

```
identify -verbose pic.png
```

获取指定信息

```
identify -format "%[EXIF:copyright] %c\n" pic.png
# copyright可以替换
```

音频

WAV

无损压缩格式

基本结构 chunk

- RIFF chunk
- Format chunk
- Data chunk（小端）

PCM数据类型	采样	采样
8Bit 单声道	声道0	声道0
8Bit 双声道	声道0	声道1
16Bit 单声道	声道0低位，声道0高位	声道0低位，声道0高位
16Bit 双声道	声道0低位，声道0高位	声道1低位，声道1高位

基本思路

时域隐写

将数据隐藏在时域波形中

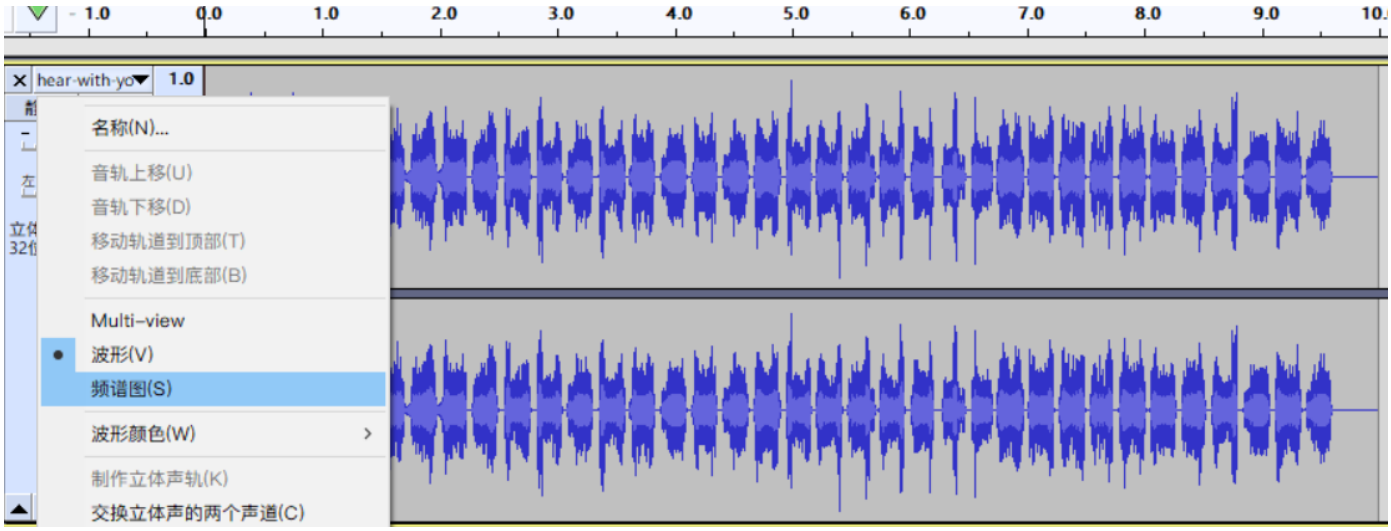
解题方法：总体观察 + 提取数据

工具：audacity 打开音频查看波形

先确定数据在那个时间段，放大进行观察波形异常

频域隐写

使用工具 audacity 打开音频，切换至频谱图



LSB隐写

wav 格式优先考虑lsb隐写，使用silenteye工具

steghide（wav隐藏信息）

使用方法在本文jpg的介绍中

MP3

压缩格式

mp3steg mp3加解密工具

有key的话用mp3steg

在MP3stego文件夹中打开cmd，然后将 **Decode.exe** 拖到命令行里，将要解密的文件放在文件夹中

```
encode -E hidden_text.txt -P pass svega.wavsvega_stego.mp3

Decode.exe -X -P pass(密码) svega_stego.mp3(要拷贝到目录下) //解码
```

m4a文件头

00 00 00 20 66 74 79 70 4D 34 41 20 00 00 00 00

DTMF

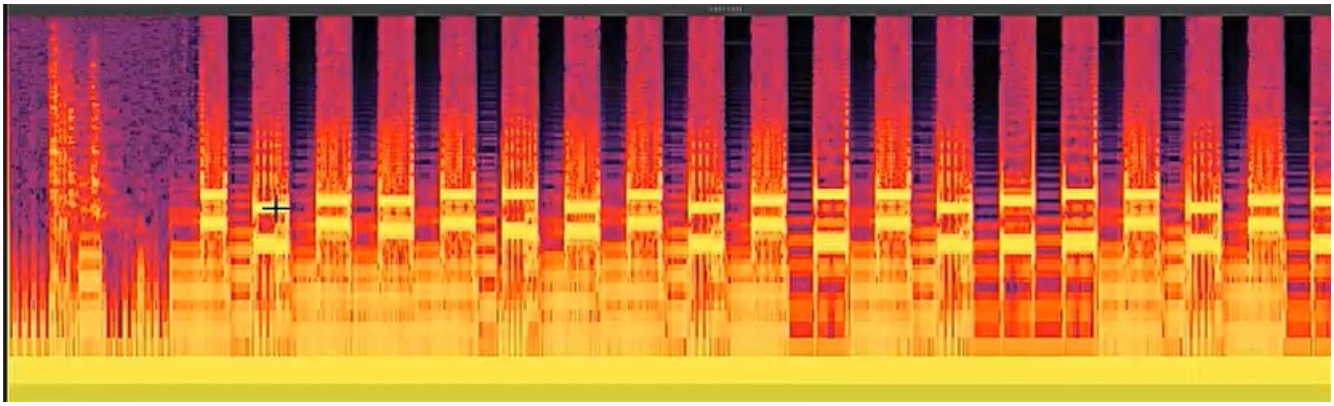
DTMF 双音多频，现在电话拨号机制，每个电话按键都有不同的声音

特点就是电话按键声音

双音多频键盘

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

音频图中查看



- 在线工具http://dialabc.com/sound/detect/
- 本地工具 **tdmf ton decoder**
- https://pas-products.com/download.html免费版有限制

脚本工具

wav

python wav 库

```
import wave
w = wave.open(file)
w.getframerate() # 返回采样频率
w.getnframes() # 返回总帧数
w.readframes(n) # 读取前n帧（不一定是n字节）
```

python列表生成式

快速将01二进制转换成英文字符

```
''.join(chr(int(s[i:i+7],2)) for i in range(0,len(s),7))
```

视频

- 逐帧分割

video to pic.exe或者ffmpeg.exe

- 其他

视频中的音频、视频放到010中查看

PYC隐写

> 导入 python 脚本时在目录下会生成个一个相应的 pyc 文件，是 pythoncodeobj 的持久化储存形式, 加速下一次的装载

- uncompyle6（pyc文件反编译）

```
uncompyle6 test.pyc > test.py
```

Stegosaurus（pyc隐写_win）

版本：Python 3.6 or later

使用在 **stegosaurus.py** 所在文件夹打开cmd，输入：

```
python stegosaurus.py -x [pyc_file]
```

基本用法

```
$ python3 -m stegosaurus -h
usage: stegosaurus.py [-h] [-p PAYLOAD] [-r] [-s] [-v] [-x] carrier

positional arguments:
  carrier               Carrier py, pyc or pyo file

optional arguments:
  -h, --help            show this help message and exit
  -p PAYLOAD, --payload PAYLOAD
                        Embed payload in carrier file
  -r, --report          Report max available payload size carrier supports
  -s, --side-by-side    Do not overwrite carrier file, install side by side
                        instead.
  -v, --verbose         Increase verbosity once per use
  -x, --extract         Extract payload from carrier file
```

乐符解密

https://www.qqxiuzi.cn/bianma/wenbenjiami.php?s=yinyue

PDF

```
pdfinfo 文件名 #查看pdf属性
pdftotext 文件名 #导出文本
```

word/xlsx

- 将文件后缀改为zip，查找里边的flag
- flag为白色字符串，ctrl+a将颜色改为白色

pcap文件修复

- winpcapfix工具• 在线修复• https://f00l.de/hacking/pcapfix.php

linux光盘文件(ext3)

linux挂载光盘，使用notepad或者strings、file命令来搜索关键词

```
strings test | grep -i flag
find | grep 'flag' 或 find -name 'flag*'
```

mount命令挂载文件