misc 大杂烩考点集合

原创

匡小萌 于 2020-07-17 19:53:59 发布 1380 收藏 2

分类专栏： CTF

本文链接：https://blog.csdn.net/khy123khy/article/details/107416530

http://www.bugku.com/forum-43-1.html

https://bbs.pediy.com/thread-220021.html

文章目录

信息搜集

狭义的Recon

搜索 引擎，搜索语法

广义的Recon

编码分析（常见编码、转换技巧）

bin二进制

二进制与其他进制之间的转换，需要转换成十进制时，在Python中，必须遵循

```
>>> int(1001,2)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: int() can't convert non-string with explicit base
>>> int("1001",2)
9
```

如下命令格式，需要转换的需要以字符串的形式，并在后面说明类型。

ascii 7或8位

二维码

*二维码修复 与定位点比较，与常见的二维码比较
*gif 二维码叠加
二维码编码形式

*此题则是由二进制与二维码的转换，共有625位，组成二维码的25*25,1代表黑，0代表白，

字符化

此题则是提取出二维码，发现根据1和0 的排列不同从而显示出特定的字符

Morse

Morse电码一般在音频中，且一般为时域的波形。往往这类题会有相关的用词，比如仔细听音乐这类似的，想起第三届xman的选拔赛的“my little pony"。当听到正常音乐中突然有嘈杂的噪声时一般就是Morse可能的地方不过时长一般很短，需把波形进行处理，比如放大之类，然后才会有明显的特征。
不一定是01，只要是两种元素的组合都可以当成是二出题码的转换，所以扩展的空间挺大。

base64

1）末尾一般“=”，大写加小写
2）3x8转换成4x6
3）建议在Linux 里 python 中进行解码

```
>>> import base64
>>> s="sdffsdd"
>>> a=base64.b64encode(s)
>>> print a
c2RmZnNkZA==
>>> b=base64.b64decode(a)
>>> print b
sdffsdd
>>>
```

这个适用于python 2 中

在base64 的隐写是这样的，因为64的规则是将3个8比特的字符用4个6比特的字符来替代，虽然ascii码是八位，但是用不到全部，6位能满足需求。所以当把等号去掉时，之前末尾补得零也去掉了，因此这几个零对解码就没有影响，我们的信息就可以隐写在其中。
推荐一个base64的解密网站，能直接把文本拖进去进行解密，然后把输出的文档下载就是结果
https://gchq.github.io/CyberChef/

b85 base85 在Python3 里

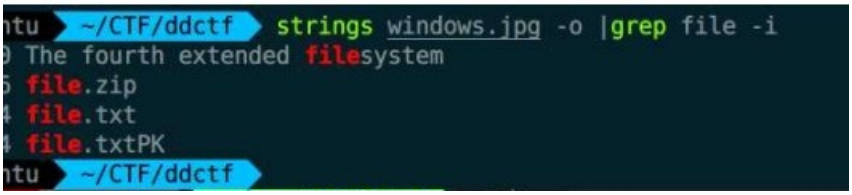
Python内置的base64模块可以实现base64、base32、base16、base85、urlsafe_base64的编码解码，python 3.x通常输入输出都是二进制形式，2.x可以是字符串形式。

其他编码

基本上编码的类型都有囊括
https://hackfun.org/2017/02/22/CTF%E4%B8%AD%E9%82%A3%E4%BA%9B%E8%84%91%E6%B4%9E%E5%A4%A7%E5%BC%80%E7%9A%84%E7%BC%96%E7%A0%81%E5%92%8C%E5%8A%A0%E5%AF%86/

工具

http://tool.ph0en1x.com/hashtool/tools.html#conv/ 一个在线强大的工具
JPK
Shell
Python grep strings & awk &sed &cat
NTFS隐写 http://sus.njnet6.edu.cn:20000/HKT7c/chapter_1.rar
#数字取证&隐写分析
1.发现文件中包含的隐藏字符串
2.文件修复
3.字符串与加密
4.特殊编码
技能需求：编码方式，Python（字符串处理、二进制文件处理、文件处理zip png pcap）、c，文件格式，工具（file鉴定文件类型，strings binwalk winhex Grep awk）
##工具
Strings(能显示文件中的可打印的字符)



http://man.linuxde.net/strings
binwalk(能根据文件头自动识别包含文件，并自动化提取，搜索嵌入的二进制镜像文件的代码及文件）
https://github.com/ReFirmLabs/binwalk
010editor，winhex
文件的16进制分析

文件载体

文件格式

png

PNG文件署名域+标准数据块(+辅助数据块)

注意：png是以大端模式进行数据的存储，即数据的高字节存放在内存的低地址
文件署名 89 50 4e 47 0d 0a 1a 0a 地址从左往右逐渐增加，所以实际数据为0x89504e47…
和实际阅读习惯一致。
每个数据块中包括了以下四个部分，CRC是由chunk TYpe Code 和Chunk Data计算得到的

名称	字节数	含义
Length	4字节	数据域长度
Chunk Type Code	4字节	A-Za-z组成
Chunk Data	可变	数据
CRC	4字节	循环冗余码

标准数据块：

除了数据块：

*IHDR文件头数据块：位置第一个，图像数据的基本信息

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)

*PLTE调色数据块：索引彩色图像有关，在IDAT数据块前

颜色	字节	意义
Red	1 byte	0 = 黑色, 255 = 红
Green	1 byte	0 = 黑色, 255 = 绿色
Blue	1 byte	0 = 黑色, 255 = 蓝色

*IDAT图像数据块：实际的图像数据，可多个连续

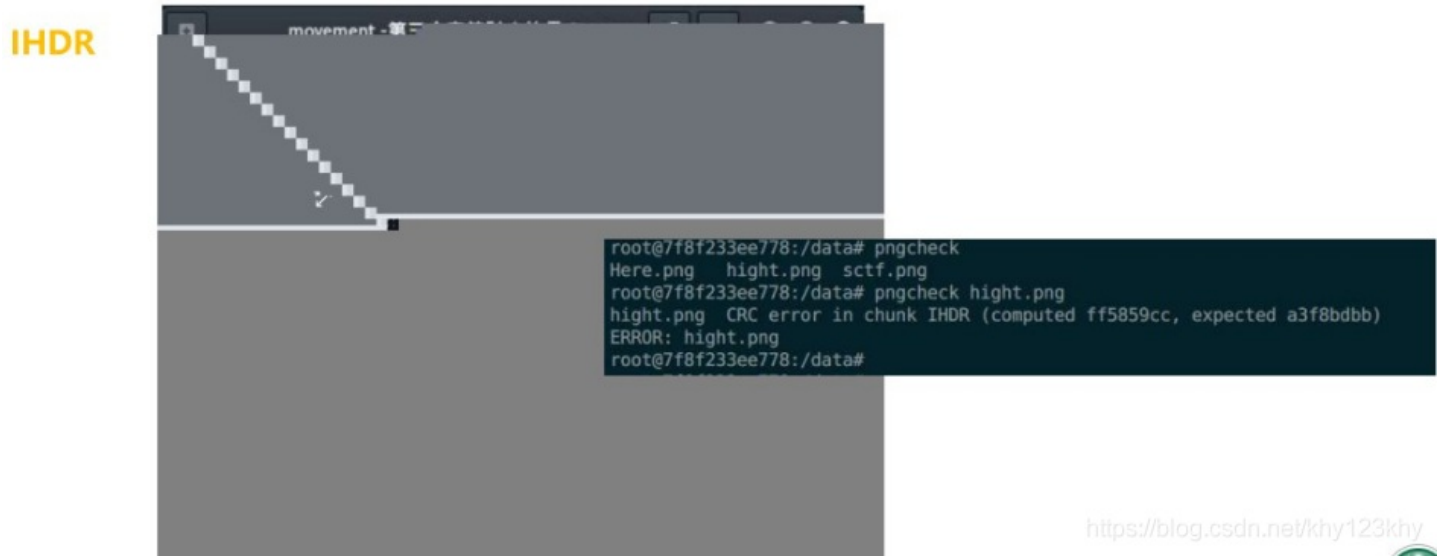
*IEND结束数据块：00 00 00 00| 49 45 4E 44 | AE 42 60 82，结束标志

不难明白，由于数据块结构的定义，IEND数据块的长度总是0（00 00 00 00，除非人为加入信息），数据标识总是IEND（49 45 4E 44），因此，CRC码也总是AE 42 60 82。

除了头和尾中间的可随意放置

数据块符号	数据块名称	多数据块	可选否	位置限制
IHDR	文件头数据块	否	否	第一块
cHRM	基色和白色点数据块	否	是	在PLTE和IDAT之前
gAMA	图像γ数据块	否	是	在PLTE和IDAT之前
sBIT	样本有效位数据块	否	是	在PLTE和IDAT之前
PLTE	调色板数据块	否	是	在IDAT之前
bKGD	背景颜色数据块	否	是	在PLTE之后IDAT之前
hIST	图像直方图数据块	否	是	在PLTE之后IDAT之前
tRNS	图像透明数据块	否	是	在PLTE之后IDAT之前
oFFs	(专用公共数据块)	否	是	在IDAT之前
pHYs	物理像素尺寸数据块	否	是	在IDAT之前
sCAL	(专用公共数据块)	否	是	在IDAT之前

IDAT	图像数据块	是	否	与其他IDAT连续
tIME	图像最后修改时间数据块	否	是	无限制
tEXt	文本信息数据块	是	是	无限制
zTXt	压缩文本数据块	是	是	无限制
fRAc	(专用公共数据块)	是	是	无限制
gIFg	(专用公共数据块)	是	是	无限制
gIFt	(专用公共数据块)	是	是	无限制
gIFx	(专用公共数据块)	是	是	无限制
IEND	图像结束数据	否	是	最后一个数据块

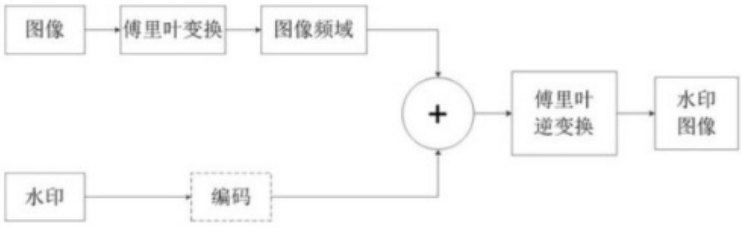


示crc error 高度发生改变，pngcheck 工具检查，IHDR 高度修改，需i根据CRC值进行修改。
PLTE:色彩图像相关
IDAT:可以多个连续的数据块
IEND：文件结束
binwalk 工具
LSB:最低有效位
stegslove 工具 图片通道，只适用于无损压缩像png这种格式
频域手段加手印：
tweakpng工具

频域手段添加数字盲水印

https://www.zhihu.com/question/50735753


```
IM = imread('shimakaze.bmp');
IM_FFT = fft(IM);
imshow(real(uint8(IM_FFT)));
```



jpg

文件格式：https://blog.csdn.net/STN_LCD/article/details/78629029
隐写介绍：https://3gstudent.github.io/3gstudent.github.io/%E9%9A%90%E5%86%99%E6%8A%80%E5%B7%A7-%E5%88%A9%E7%94%A8JPEG%E6%96%87%E4%BB%B6%E6%A0%BC%E5%BC%8F%E9%9A%90%E8%97%8Fpayload/
每个数据段的格式：

段类型:

名称	标记码	说明
SOI	D8	文件头
EOI	D9	文件尾
SOF0	C0	帧开始（标准 JPEG）
SOF1	C1	同上
DHT	C4	定义 Huffman 表（霍夫曼表）
SOS	DA	扫描行开始
DQT	DB	定义量化表
DRI	DD	定义重新开始间隔
APP0	E0	定义交换格式和图像识别信息
COM	FE	注释

说明：有的文章也将DNL段（标记码=DC，定义扫描行数）列为必须段。

隐写：
1.直接在尾部添加数据。ffd9 后面加东西
2.插入自定义COM注释。
jepgsnoop http://www.impulseadventure.com/photo/jpeg-snoop.html
stegdetect工具，用于分析
https://github.com/abeluck/stegdetect
http://www.bugku.com/thread-56-1-1.html

q – 仅显示可能包含隐藏内容的图像n – 启用检查JPEG文件头功能，以降低误报率。如果启用，所有带有批注区域的文件将被视为没有被嵌入信息。如果JPEG文件的JFIF标识符中的版本号不是1.1，则禁用OutGuess检测。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。d – 打印带行号的调试信息。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。

identity 命令
convert命令

GIF

文件格式

在线编辑GIF的 https://ezgif.com/split
https://blog.csdn.net/riba2534/article/details/70544076?locationNum=14&fps=1
imagemagic套件中，提供图片分析的功能，
1.identity 命令
https://www.ibm.com/developerworks/cn/linux/l-graf/index.html
比如说 `identify -format "%wx%h" xx.png` 用-format能指定图片输出想知道的图片信息，具体参数查看命令
2.convert命令
https://www.ibm.com/developerworks/cn/linux/l-graf/index.html

流量包分析

文件修复（pcapfix) 在线网站https://f00l.de/hacking/pcapfix.php
协议分析（wireshark） http/https/dns/ftp
数据提取（tshark)

- 手工提取：
  - Python：pcapy
  - Tshark：
    - -r **.pcap –Y ** -T fields –e ** |

```
tshark -r fore2.pcap -Y 'usb.capdata and usb.device_address==3' -T fields -e usb.capdata > raw
```

-e 后面的参数为选定显示的数据段，
鼠标的流量提取，通过鼠标画出flag，然后提取出流量分析
键盘的流量提取

https://delcoding.github.io/2018/05/ciscn-writeup/ 18年国赛 ciscn

压缩包分析

zip文件格式 https://blog.csdn.net/ETF6996/article/details/51946250
是小端模式。???
压缩源文件数据区：
文件头标记50 4B 03 04 数据就是 0x04034b50
接着后2个字节为解压文件所需的PKware版本，再后两个字节为加密标记位00 00
压缩源文件目录区：
核心目录文件头标识50 4B 01 02 数据就是 0x02014B50
压缩源文件目录结束标志：
目录结束标识结构50 4B 05 06 数据就是 0x06054b50
zip隐写：
1.通过进制转换隐藏信息
比如给一串16进制或2进制的信息，解密后能看见文件头例如PK等，然后将16进制数据导进16进制编辑器，保存为相应的文件类型
2.在图片中隐藏压缩包（图种）
基本上就是在一张图片中隐藏一个压缩包，比如说jpg格式的文件，其文件由FF D8开始，FF D9结束，图片浏览器会自动忽略D9后面的的内容，所以一般可以在后面加上隐藏的内容。
这种类型就是通过binwalk分析，然后foremost提取。

加密

1.直接爆破。（ARCHPR window下，Linux下fcackzip）
2.伪加密。 zip中有一位是标记文件是否加密的，如果更改一个未加密zip包的加密标记位，那么在打开压缩包时就会提示该文件是加密的。
可以用16进制编辑器修改加密标志位 用WinRAR的修复功能。
在Mac或Linux貌似可以直接打开伪加密的zip
3.CRC32碰撞
CRC 本身是「冗余校验码」的意思，CRC32 则表示会产生一个 32 bit（8 位十六进制数）的校验值。由于 CRC32 产生校验值时源数据块的每一个 bit（位）都参与了计算，所以数据块中即使只有一位发生了变化，也会得到不同的 CRC32 值。

CRC32 校验码出现在很多文件中比如 png 文件，同样 zip 中也有 CRC32 校验码。值得注意的是 zip 中的 CRC32 是未加密文件的校验值。

主要用于对于内容比较短（通常比赛中为四个字节），加密密码长，然后不爆破压缩包密码，而是爆破内容，得出CRC32与之匹配。

主要用到一个itertools模块，调用 `itertools.product(,repeat=)` 用来生成一个迭代器，即一个排列组合

```
>>> y=itertools.product('abcd',repeat=3)
>>> for x in y
  File "<stdin>", line 1
    for x in y
             ^
SyntaxError: invalid syntax
>>> for x in y:
... print x
  File "<stdin>", line 2
    print x
        ^
IndentationError: expected an indented block
>>> for x in y:
...     print x
...
('a', 'a', 'a')
('a', 'a', 'b')
('a', 'a', 'c')
('a', 'a', 'd')
('a', 'b', 'a')
```

前面是大集合，后面的repeat表示重复的次数。
Python2中，crc32计算出来的值带有符号需要&0xffffffff, 在Python3中不用
4.明文攻击
当你不知道一个zip的密码，但已知里面的一个加密文件（大于12byte），因为压缩包里的所有文件都是使用同一个加密密钥来加密的，可以用已知文件来寻找加密密钥，从而利用密钥来解锁其他加密文件。
在Linux平台中 可以用`zipinfo -v`查看一个压缩包的信息
pkcrack攻击 http://www.cnblogs.com/ECJTUACM-873284962/p/9387711.html

音频隐写

*频谱 时域转频域，高频部分
*波形 集中区域很少，往往最开始时候，一般听尖锐声音，
*工具 用mp3stego

内存分析

*进程 确定内存结构，查看进程列表，根据题目寻找相关进程 volatility工具

其他

pyc文件

Magic [4]	TIMESTAMP [4]	PyCodeObject
表示pyc版本信息	创建时间	

co_code 指令序列 指令opcode+参数oparg

沙箱 Python

Word 字体隐藏

PDF wbstego4工具

NTFS数据流

ntfsstreamseditor 工具

视频

AVI MP4 Ourstrect工具

HTML隐写

http://fog.misty.com/perry/ccs/s

工控

misc+

- 验证码破解
  - 机器学习 + 图像处理

SUCTF-2018

- HateIT
  - 文件泄露 + githack

+Crypto

古典密码学

- 栅栏啊
- 凯撒啊
- 维吉尼亚啊
- 字频分析啊
- 培根啊

中低分Misc题

现代密码学

- Aes啊
- Des啊
- Balabala啊
-

高分Misc题

+Pwn

Python SandBox

- Ciscn 2018 Run

```
>>>(lambda r,w:r.seek(0x08de2b8) or w.seek(0x08de8c8) or w.write(r.read(8)) or
().__class__.__bases__[0].__subclasses__()[40]('l'+'s'))
((().__class__.__bases__[0].__subclasses__()[40]('/proc/self/mem','r'),
().__class__.__bases__[0].__subclasses__()[40]('/proc/self/mem', 'w', 0))
```


+Re

- 固件分析
  - 32C3 CTF 2015 config-bin-150
- 病毒分析

Binwalk,winhex, Firmware Modification Kit
Squashfs-tools

赛题分析

http://sus.njnet6.edu.cn:20000/HKT7c/chapter_1.rar
https://github.com/XuCcc/InitUbuntu