ctf-stego汇总

原创

n4nch3ng 于 2017-03-16 17:23:51 发布
分类专栏： misc 文章标签： stego ctf

本文链接：https://blog.csdn.net/fuzz_nancheng/article/details/62428776


misc 专栏收录该内容
4 篇文章 0 订阅
订阅专栏

相关命令，工具：

***binwalk工具使用（提取文件隐藏的文件，原理就是检索匹配文件头）
binwalk test.jpg //显示文件名
dd if=test.png of='test.rar' bs=1 skip=100097 //根据偏移量提取文件
binwalk -e test.jpg //识别并自动化提取test.jpg中的文件
（http://www.freebuf.com/sectool/15266.html）

***formost恢复文件工具（extundelete是另一种）
基于文件头和尾部信息以及文件的内建数据结构恢复文件的命令行工具
支持恢复如下格式：avi, bmp, dll, doc, exe, gif, htm, jar, jpg, mbd, mov,
mpg, pdf, png, ppt, rar, rif, sdw, sx, sxc, sxi, sxw, vis, wav, wmv, xls, zip。
安装#apt-get install foremost
删除USB（/dev/sdba1）存储器中一个 png 文件然后使用 formost 恢复：
rm -f /dev/sdb1/photo1.png
foremost -t png -i /dev/sdb1
（http://os.51cto.com/art/201401/426316.htm）

***strings，file，cat

*判断文件类型TrIDNet(http://www.pconline.com.cn/pcedu/teach/tool/0707/1065748.html)
下载后即可直接使用

图片隐写：
*对图片微处理
根据题目信息来做，可能跟cookie, 编码结合
可能通过firebug查看html，js注释的隐藏信息
通过exif属性的信息隐写：右键图片-属性-详细信息
一些文件如png无法直接打开，也许是缺少文件头，在16进制下加上文件头后即可
尝试修改图片分辨率，查看flag是否被覆盖：http://bobao.360.cn/ctf/learning/137.html 强网杯
base64编码后的文件：Data:image/png;base64…….. Base64编码后的png图片，放到浏览器上回车得到图片
改后缀.txt，搜索flag，key，ctf，{，flage等关键词
有些图片直接用记事本打开看不到内容，但在linux下用cat filename就会得到意想不到的结果。

***stegsolve（http://www.caesum.com/handbook/Stegsolve.jar）
Stegsolve——Analyse——Frame Browser。浏览每个颜色通道的每一位，看是否隐藏了二维码
（二维码在线解码：http://tool.chinaz.com/qrcode/）
Stegsolve——Analyse——Data Extract。这个是使用ASCII隐写的时候可以查看的。
Stegsolve——Analyse——file format。查看文件格式，结合IDAT结构是否异常，结合编程。
Stegsolve——Analyse——frame browser。可以查看GIF文件每一帧的图片。
Stegsolve——Analyse——image combiner双图分析
双图分析见末尾（http://bobao.360.cn/learning/detail/243.html）

***IDAT隐写-方式一
结合stegsolve——Analyse——file format，如果倒数第二个IDAT块未满65524，
则可初步判定最后一个块为添加的块，可使用zlib解压缩（python）
import zlib import binascii IDAT
=”789C5D91011280400802BF04FFFF5C75294B5537738A21A27D1E49CFD17DB3937A92E7E603880A6D485100901FB0
410153350DE83112EA2D51C54CE2E585B15A2FC78E8872F51C6FC1881882F93D372DEF78E665B0C36C529622A0A4
5588138833A170A2071DDCD18219DB8C0D465D8B6989719645ED9C11C36AE3ABDAEFCFC0ACF023E77C17C78976
67”.decode(‘hex’) result = binascii.hexlify(zlib.decompress(IDAT)) print result

***IDAT隐写-方式二
binwalk 图片名
DECIMAL HEXADECIMAL DESCRIPTION
0x0 PNG image, 1000 x 562, 8-bit/color RGBA, non-interlaced
0x5B Zlib compressed data, compressed
0xDC6 Zlib compressed data, best compression
0x15AFFB Zlib compressed data, default
compression后面是Zlib压缩的数据，写个脚本解压一下：
python提取脚本
from PIL import Image
from zlib import *
data = open(‘图片名’,’rb’).read()[0x15AFFB:]
data = decompress(data)
img = Image.new(‘1’, (25,25))
d = img.load()
for n,i in enumerate(data):
d[(n%25,n/25)] = int(i)*255
f = open(‘flag.png’,’wb’)
img.save(f)

*此处顺便附上01字符串形成二维码的代码（需要PIL库）

```
!/usr/bin/env python
import Image
MAX = 25
pic = Image.new(“RGB”,(MAX, MAX))
str =
“1111111000100001101111111110000010111001011010000011011101010000000001011101101110100100000000101110
1101110101110110100101110110000010101011011010000011111111010101010101111111000000001011101110000000
0011010011000001010011101101111010101001000011100000000000101000000000100100110100010011100111101110
0111100001110111110001100101000110011100001010100011010001111010110000010100010110000011011101100 10
0001110011100100000101111111010000000001101010010001111011111110111000011010110111000001000011001000
1111010111010001101001111100001011101011000111010011100101110100100111011011000110000010110001101 00
0110001111111011010110111011011”
i=0
for y in range (0,MAX):
for x in range (0,MAX):
if(str[i] == ‘1’):
pic.putpixel([x,y],(0, 0, 0))
else:
pic.putpixel([x,y],(255,255,255))
i = i+1
pic.show()
pic.save(“flag.png”)
```

*加密压缩包zip或者rar破解（http://blog.csdn.net/fuzz_nancheng/article/details/53230389）
使用ARCHPR进行破解，密码使用自带密码集或者python进行编写
如果不支持，那么可能是文件头前面还有东西，去掉即可
另一种情况是伪加密参考以下网址：

***png图片-LSB隐写-cloacked-pixel工具（下载https://github.com/cyberinc/cloacked-pixel）
cloacked-pixel在加密的过程中会删除其他数据块，只保留关键数据块IDAT
加密：python lsb.py hide big.png 1.txt 123456
hide：表示加密模式 big.png：待加密的png图片
1.txt：存放payload 123456：加密的密码
运行后生成图片big.png-stego.png
解密：python lsb.py extract big.png-stego.png 3.txt 123456
extract：表示解密模式 big.png-stego.png：待解密的png图片
3.txt：存放导出的payload 123456：解密密码

***png图片-LSB隐写-LSB-Steganography工具
https://github.com/RobinDavid/LSB-Steganography
加密：python LSBSteg.py -image original.png -binary original.bin -steg-out steg.png
解密：python LSBSteg.py -steg-image steg.png -out bin

***jpg图片-使用winhex/010editor/C32Asm打开，查看文件尾FF D9后面是否有其他内容。
copy命令即copy /b 2.jpg+1.zip output.jpg
图片可利用copy命令将1.zip通过连接放到2.jpg后面并只显示2.jpg
也可以直接16进制打开后搜索flag，ctf，{，flage等关键词

***jpg隐写-steghigh
加密：steghide embed -cf picture.jpg -ef secret.txt（需要输入两次密码）
解密：steghide extract -sf picture.jpg（需要输入一次密码）
获取相关信息：steghide info received_file.wav

***jpeg图片隐写-Stegdetect（https://github.com/abeluck/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，
具有基于字典暴力破解密码方法提取通过Jphide、outguess和jsteg-shell方式嵌入的隐藏信息。
探测命令：stegdetect.exe -tjopi -s 10.0 test.jepg
-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。
```

-V 显示软件版本号。
如果检测结果显示该文件可能包含隐藏信息，那么Stegdetect会在检测结果后面用1～3颗星标识
隐藏信息存在的可能性大小，3颗星表示隐藏信息存在的可能性最大。

***jpg图片隐写-Jphide（https://www.hackfun.org/CTF/jphide-steganography.html）
先解压压缩JPEG图像，得到DCT系数；
然后对隐藏信息用户给定的密码进行Blowfish加密；
再利用Blowfish算法生成伪随机序列，并据此找到需要改变的DCT系数，
将其末位变为需要隐藏的信息的值，
最后把DCT系数重新压回成JPEG图片。
用JPHS工具解决

***jepg图片隐写-JPHS（http://io.acad.athabascau.ca/~grizzlie/Comp607/programs.htm）
由Stegdetect检测出来的，这个是分离工具,有图形界面，可以直接导入图片，txt进行加密解密

***JEPG图片-outguess工具（http://download.csdn.net/detail/florak/5620983）
下载后，在linux里cd到outguess目录下
执行命令：./configure && make && make install
这样就可以使用outguess命令了。
执行命令：outguess -r test.jpg outfile.txt
打开outfile.txt即可得到隐藏信息
***F5隐写（https://github.com/matthewgao/F5-steganography）
在文件夹中命令行模式下运行：java Extract test.jpg -p 123456
-p 后面是密码
得到output.txt，打开即可得到隐藏信息
其他隐写：
***MP3隐写-MP3stego
将MP3文件和Decode.exe和Encode.exe放在同一目录下
在命令行模式使用命令“cd 路径”进入该目录下
使用命令Decode.exe -X -P password test.mp3
其中 -X 提取隐藏数据，-P 密码 ，提取出来的信息在test.txt里面
密码可能在文件本身或者网页前端代码或者响应头或者是弱密码

***wav隐写-Audacity
-打开wav文件，Audacity-效果-反向-播放
-Audacity-文件名-频谱图
-声道里面夹杂着莫斯密码，短的代表’.’,长的代表’-’
-Audacity-频谱图-attach-点击频谱-然后点击修改spectrogram setting，把8000改为48000
-电话音分析（http://dialabc.com/sound/detect/）

***wav隐写-silenteye
wav隐写

***MP4隐写-ffmpeg，fMP4
ffmpeg.exe -i test.mp4 -r 30.0 test.bmp
把一帧帧的视频分割为一张张的图片。具体怎么使用可以在linux下man ffmpeg。

***avi隐写-MSU VideoStego

***BMP、TXT、HTM和PDF-wbStego4open
wbStego4open一般可以解决这种隐写
（http://wbstego.wbailer.com）

***PDF隐写-pdfdetach工具：
pdfdetach test.pdf -saveall（还原创建这个pdf的前一个pdf）
还原到第一个pdf后 pdfdetach start.pdf -save 1 -o origin.pdf

***vmdk隐写-Dsfok-tools，FTK
通常Dsfok-tools是用来编辑vmdk文件中的描述符
FTK直接打开文件就可以

***exe文件-Hydan
对于可执行文件的数据隐藏可以使用Hydan
或者直接用txt格式打开
或者用od，ida进行逆向分析

***NTFS文件系统ida隐藏-ntfsinfo

***dmp文件隐写

在线分析网站：http://www.osronline.com/page.cfm?name=analyze

例子：http://bobao.360.cn/ctf/learning/153.html