COMPASS CTF Tutorial 4: Forensics and Steganography

COMPASS CTF 教程【4】: 取证隐写

30016794 Zhao, Li (Research Assistant)

COMPuter And System Security Lab, Computer Science and Technology Department, College of Engineering (CE), SUSTech University.
南方科技大学 工学院 计算机科学与技术系 计算机与系统安全实验室

2023 年 8 月 12 日

杂项

Misc（Miscellaneous）指在 CTF 中无法分类在 Web、PWN、Crypto、Reverse 中的题目。Misc 题目种类繁多，但大致可以分为以下几种：[1]

- 提供参赛者参与的题目，如签到题；
- 考察传统分类之外的知识，如内容安全、安全运维、网络编程等；
- 考察思维发散能力，即”脑洞题”；
- 考察知识的广度和深度；
- 考察快速学习能力；

Misc 类型的题目既考察了各领域的基本知识，也是培养信息安全技术兴趣的极好材料。

杂项分类

1. **参与性**：不考察太多知识，偏重娱乐性，使参赛者参与、感受 CTF 的乐趣。
2. **安全领域知识**：包括内容安全、安全运维、网络编程等，常出现在 Misc 中，为出现频率最高的题型。
3. **思维发散**：解题依靠经验和猜想，考验参赛者和出题人的”脑洞”。
4. **知识广度和深度**：精髓在于从日常事物中发现不一样的东西，考察对常见事物的深度理解。
5. **快速学习**：考查偏门技术，但深度要求不高。侧重信息获取和吸收能力，搜索引擎是好助手。

CTF 竞赛中的杂项题目

难度虽各异，但 Misc 是初学者最易上手的 CTF 题目类型之一。我们将介绍几类代表性 Misc 题目，包括隐写术、压缩包分析和取证技术。

直接附加

大部分文件有其固定的文件结构，常见的图片格式如 PNG、JPG 等都是由一系列特定的数据块组成的。例如，PNG 文件由 IHDR（文件头数据块）、PLTE（调色板数据块）、IDAT（图像数据块）、IEND（图像结束数据）四个标准数据块和一些辅助数据块组成。每个数据块由 Length（长度）、Chunk Type Code（数据块类型码）、Chunk Data（数据块数据）和 CRC（循环冗余校验码）四部分组成。

PNG 文件结构

PNG 文件总是由固定的字节（89 50 4E 47 0D 0A 1A 0A）开始，我们一般可以根据这个来识别该文件是一个 PNG 文件。图像结束数据 IEND 用来标记 PNG 文件已经结束。IEND 数据块的长度总是 00 00 00 00，数据标识总是 49 45 4E 44，因此 CRC 固定为 AE 42 60 82。所以，一般 PNG 文件以固定字节 00 00 00 00 49 45 4E 44 AE 42 60 82 作为结束，其后的内容会被大部分图片查看软件忽略，所以可以在 IEND 数据块后增加其他内容，这样并不会影响图片的查看，增加的内容普通情况下不会被发现。

实验操作

选取一张 PNG 图片，使用 Windows 自带的图片查看器“Photos”打开，如图 1 所示。使用二进制编辑器打开该 PNG 图片，观察其文件头和文件尾，见图 2 和图 3。

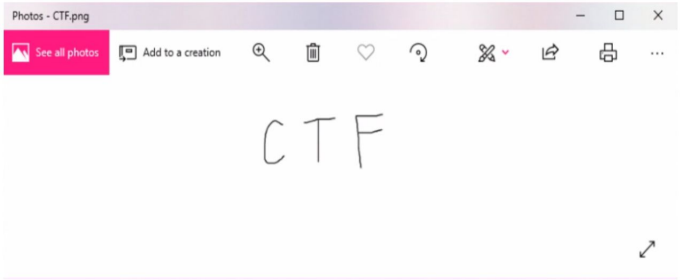


图: PNG 图片示例

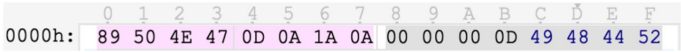

0 1 2 3 4 5 6 7 8 9 A B C D E F
0000h: 89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52

图: 文件头

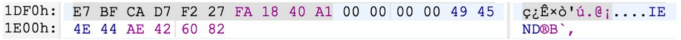

1DF0h: E7 BF CA D7 F2 27 FA 18 40 A1 00 00 00 00 49 45
1E00h: 4E 44 AE 42 60 82
ç¿Ê×ò'ú.@¡....IE
ND®B`,

图: 文件尾

修改文件结尾内容

可以在文件结尾任意添加内容，如直接在文件结尾添加字符'HELLO WORLD'。仍用“Photos”打开这个文件，发现其与修改前并没有任何变化，刚刚添加的'HELLO WORLD' 并不会显示在图片上。

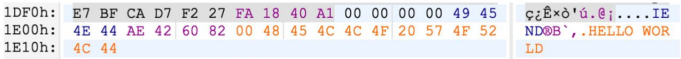


图: 向文件尾添加内容

不仅是字符，我们甚至可以将其他文件整个添加到图片后，通过图片查看器也不会看到任何变化。

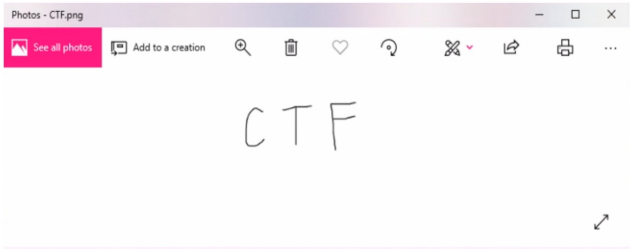


图: 图片的内容未发生变化

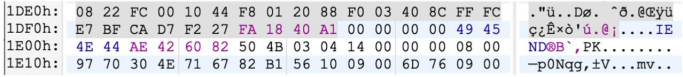
分离附加在图片后面的文件

要分离出附加在图片后面的文件，可以通过观察二进制中隐含的文件头信息来判断图片中附加的文件类型，常见的文件头、文件尾分别如下：

- JPEG（jpg）：**文件头**，FF D8 FF；**文件尾**，FF D9。
- PNG（png）：**文件头**，89 50 4E 47；**文件尾**，AE 42 60 82。
- GIF（gif）：**文件头**，47 49 46 38；**文件尾**，00 3B。
- ZIP Archive（zip）：**文件头**，50 4B 03 04；**文件尾**，50 4B。
- RAR Archive（rar）：**文件头**：52 61 72 21。
- Wave（wav）：**文件头**，57 41 56 45。
- AVI（avi）：**文件头**，41 56 49 20。
- MPEG（mpg）：**文件头**，00 00 01 BA **或者** 00 00 01 B3。
- Quicktime（mov）：**文件头**，6D 6F 6F 76。

使用 Binwalk 工具

我们也可以使用 Binwalk 工具分离图片中附加的其他文件。Binwalk 是一款开源的固件分析工具，可以根据固件中出现的各类文件的一些特征，识别或提取这些文件，因此在 CTF 中 Binwalk 常常用于从一个文件中提取出它包含的其他文件。如 PNG 图片结尾附加上一个 ZIP 文件的二进制内容。
Binwalk 可以自动分析一个文件中包含的多个文件并将它们提取出来。



```
1DE0h:  08 22 FC 00 10 44 F8 01 20 88 F0 03 40 8C FF FC   ."ü..Dø. ˆð.@Œÿü
1DF0h:  E7 BF CA D7 F2 27 FA 18 40 A1 00 00 00 00 49 45   ç¿Ê×ò'ú.@¡....IE
1E00h:  4E 44 AE 42 60 82 50 4B 03 04 14 00 00 00 08 00   ND®B`‚PK........
1E10h:  97 70 30 4E 71 67 82 B1 56 10 09 00 6D 76 09 00   —p0Nqg‚±V...mv..
```

图: 附加了 ZIP 的 PNG 文件

```
→  book binwalk -e CTF.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 680 x 1088, 8-bit/color RGB, non-interlaced
91            0x5B            Zlib compressed data, compressed
7686          0x1E06          Zip archive data, at least v2.0 to extract, compressed size: 594006, unc
ompressed size: 620141, name: 1500965-e698568d37389be9.png
601860        0x92F04         End of Zip archive
```

图: 使用 binwalk 进行分析

EXIF

EXIF（Exchangeable Image File Format，可交换图像文件格式）可以用来记录数码照片的属性信息和拍摄数据。EXIF 可以被附加在 JPEG、TIFF、RIFF 等文件中，为其增加有关数码相机拍摄信息的内容、缩略图或图像处理软件的一些版本信息。

选取一张 Windows 自带的示例图片（JPEG 格式），通过右键查看它的属性，可以看到保存了作者、拍摄日期、版权等信息。

EXIF 数据结构可以参考自 http://www.fifi.org/doc/jhead/exif-e.html。用二进制打开这个图片，对比 EXIF 结构，可以看到其中的一些 EXIF 信息。我们可以使用二进制编辑器手工修改其中的信息，也可以使用一些工具（如 exiftool）进行 EXIF 文件信息的查看和修改。

用命令“exiftool -comment=ExifModifyTesting ./Lighthouse.jpg”为这张图片添加标签，用命令“exiftool ./Lighthouse.jpg”可以查看 EXIF 信息，可以发现增加了一个 comment 标签，内容为 ExifModifyTesting。我们可以利用这种方式将一些信息隐藏其中。


Lighthouse.jpg 属性
常规
文件校验
详细信息
属性
值
说明
标题
主题
分级
标记
备注
来源
作者
Tom Alphin
拍摄日期
2008/2/11/周一 11:32
程序名称
获取日期
版权
Microsoft Corporation
图像
图像 ID
分辨率
1024 x 768
宽度
1024 像素
高度
768 像素
水平分辨率
96 dpi
垂直分辨率
96 dpi
删除属性和个人信息
确定
取消
应用(A)

图: 图像的 EXIF 信息

FFE1	APP1 Marker		
SSSS	APP1 Data	APP1 Data Size	
45786966 0000		Exif Header	
49492A0008000000/4d4d002a00000008		TIFF Header(Little Endian) / TIFF Header(Big Endian)	
XXXX···		IFD0 (main image)	Directory
LLLLLLLL			Link to IFD1
XXXX···		Data area of IFD0	
XXXX···		Exif SubIFD	Directory
00000000			End of Link
XXXX···		Data area of Exif SubIFD	
XXXX···		Interoperability IFD	Directory
00000000			End of Link
XXXX···		Data area of Interoperability IFD	
XXXX···		Makernote IFD	Directory
00000000			End of Link
XXXX···		Data area of Makernote IFD	
XXXX···		IFD1(thumbnail image)	Directory
00000000			End of Link
XXXX···		Data area of IFD1	
FFD8XXXX···XXXXFFD9		Thumbnail image	

图: EXIF 数据结构

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	FF	D8	FF	E0	00	10	4A	46	49	46	00	01	02	01	00	60	ÿØÿà..JFIF.....`
0010h:	00	60	00	00	FF	EE	00	0E	41	64	6F	62	65	00	64	00	.`..ÿî..Adobe.d.
0020h:	00	00	00	01	FF	E1	0D	FE	45	78	69	66	00	00	4D	4D	....ÿá.þExif..MM
0030h:	00	2A	00	00	00	08	00	08	01	32	00	02	00	00	00	14	.*.......2......
0040h:	00	00	00	6E	01	3B	00	02	00	00	00	0B	00	00	00	82	...n.;..........,
0050h:	47	46	00	03	00	00	00	01	00	05	00	00	47	49	00	03	GF..........GI..
0060h:	00	00	00	01	00	58	00	00	82	98	00	02	00	00	00	16	.....X..,˜......
0070h:	00	00	00	8D	9C	9D	00	01	00	00	00	16	00	00	00	00	....œ...........
0080h:	EA	1C	00	07	00	00	07	A2	00	00	00	00	87	69	00	04	ê.......¢....‡i..
0090h:	00	00	00	01	00	00	00	A3	00	00	01	0D	32	30	30	39	.......£....2009
00A0h:	3A	30	33	3A	31	32	20	31	33	3A	34	38	3A	33	32	00	:03:12 13:48:32.
00B0h:	54	6F	6D	20	41	6C	70	68	69	6E	00	4D	69	63	72	6F	Tom Alphin.Micro
00C0h:	73	6F	66	74	20	43	6F	72	70	6F	72	61	74	69	6F	6E	soft Corporation
00D0h:	00	00	05	90	03	00	02	00	00	00	14	00	00	00	E5	90	.............å.
00E0h:	04	00	02	00	00	00	14	00	00	00	F9	92	91	00	02	00	..........ù'‘...
00F0h:	00	00	03	37	37	00	00	92	92	00	02	00	00	00	03	37	...77..’’......7
0100h:	37	00	00	EA	1C	00	07	00	00	07	B4	00	00	00	00	00	7..ê......´.....
0110h:	00	00	00	32	30	30	38	3A	30	32	3A	31	31	20	31	31	...2008:02:11 11
0120h:	3A	33	32	3A	35	31	00	32	30	30	38	3A	30	32	3A	31	:32:51.2008:02:1
0130h:	31	20	31	31	3A	33	32	3A	35	31	00	00	05	01	03	00	1 11:32:51.....
0140h:	03	00	00	00	01	00	06	00	00	01	1A	00	05	00	00	00	.....

图: 二进制编辑器中的 EXIF 信息

```
Legacy IPTC Digest              : 693209D7C351232255ED533263933194
Marked                          : True
Creator                         : Tom Alphin
Rights                          : © Microsoft Corporation
Current IPTC Digest             : 693209d7c351232255ed533263933194
Application Record Version      : 2
By-line                         : Tom Alphin
Copyright Notice                : © Microsoft Corporation
IPTC Digest                     : 693209d7c351232255ed533263933194
Copyright Flag                  : True
Comment                         : ExifModifyTesting
Image Width                     : 1024
Image Height                    : 768
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:4:4 (1 1)
Image Size                      : 1024x768
Megapixels                      : 0.786
Create Date                     : 2008:02:11 11:32:51.77
Date/Time Original              : 2008:02:11 11:32:51.77
Thumbnail Image                 : (Binary data 3223 bytes, use -b option to extract)
```

图: 增加新的 EXIF 信息条目

EXIF 文件类型

EXIF 文件主要用于 JPEG 和 TIFF 文件格式，这两种格式广泛用于照片和图像存储。EXIF 可以记录很多有关照片的信息，包括日期和时间、相机设置、描述、版权信息，甚至可能包含 GPS 位置信息。这些元数据可以帮助图像处理软件更好地处理照片，也能为用户提供更丰富的信息。

读取和修改 EXIF 信息

大多数现代图像浏览器和编辑器都能读取 EXIF 信息。有一些立体声图像软件，例如 Adobe Photoshop，还可以编辑或删除这些信息。此外，还有一些专门的工具，如 ExifTool，专门用来读取和修改 EXIF 信息。它可以用于添加、修改或删除 EXIF 信息，非常适合批量处理照片。

EXIF 信息的用途

EXIF 信息的主要用途是记录照片的拍摄条件并帮助图像编辑，但现在它也被用于更多的情况。例如，它可以被用来判断照片是否被修改过，或者在法庭上作为证据提供拍摄说明。事实上，EXIF 信息已经成为数字取证的重要部分。

EXIF 信息的潜在风险

尽管 EXIF 信息在很多情况下都是非常有用的，但是它也可能带来一些潜在的隐私风险。例如，如果你不小心上传了包含 GPS 信息的照片，那么任何看到这个照片的人都可能知道你的拍摄地点。因此，在分享照片之前，删除敏感的 EXIF 信息是很重要的。

LSB

Least Significant Bit (LSB) 是一种隐藏信息的方法。在大多数 PNG 图像中，每个像素都由 R、G、B 三原色组成，每种颜色一般用 8 位数据表示（0x00 ~ 0xFF）。由于人眼无法区分其最低有效位的微小变化，我们可以利用这一点来在每个像素的 R、G、B 颜色分量的最低有效位隐藏信息，这样每个像素可以携带 3 位的信息。

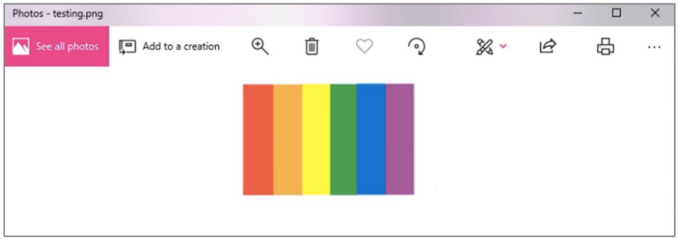


图: 原始图像

LSB

一种方法是先准备一张图片，然后将一个字符串通过 LSB 的方式隐藏在图片中。调用 lsb_encode() 方法，生成隐写后的图片，肉眼并不能看出明显的变化。调用 lsb_decode() 方法，可以提取出隐写的内容。
在 CTF 中，经常使用的检测 LSB 隐写痕迹的工具是 Stegsolve。此工具还可以查看图片不同的通道，进行图片的异或对比等操作。

```
#coding:utf-8
from PIL import Image

def lsb_decode(l, infile, outfile):
    f = open(outfile,"wb")
    abyte=0
    img = Image.open(infile)
    lenth = l*8
```

图: 写入 LSB 信息：代码（上）

```
    width = img.size[0]
    height = img.size[1]
    count = 0
    for h in range(0, height):
        for w in range(0, width):
            pixel = img.getpixel((w, h))
            for i in range(3):
                abyte = (abyte<<1)+(int(pixel[i])&1)
                count+=1
                if count%8 == 0:
                    f.write(chr(abyte))
                    abyte = 0
            if count >= lenth :
                break
        if count >= lenth :
            break
    f.close()

def str2bin(s):
    str = ""
    for i in s :
        str +=(bin(ord(i))[2:]).rjust(8,'0')
    return str

def lsb_encode(infile,data,outfile):
    img = Image.open(infile)
    width = img.size[0]
    height = img.size[1]
    count = 0
    msg = str2bin(data)
    mlen = len(msg)
    for h in range(0,height):
        for w in range(0,width):
            pixel = img.getpixel((w,h))
            rgb=[pixel[0],pixel[1],pixel[2]]
            for i in range(3):
                rgb[i] = rgb[i] & 0xfe + int(msg[count])&1
                count+=1
                if count >= mlen :
                    img.putpixel((w,h),(rgb[0],rgb[1],rgb[2]))
                    break
            img.putpixel((w,h),(rgb[0],rgb[1],rgb[2]))
            if count >=mlen :
                break
        if count >= mlen :
            break
    img.save(outfile)

#原图
old = "./testing.png"

#隐写后的图片
new = "./out.png"
```

图: 写入 LSB 信息：代码（中）

```
#需要隐藏的信息
enc = "LSB_Encode_Testing"

#信息提取出后所存放的文件
flag = "./get_flag.txt"

lsb_encode(old,enc,new)
lsb_decode(18,new,flag)
```

图：写入 LSB 信息：代码（下）

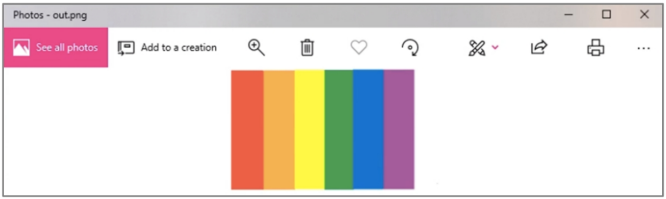


图: 处理后的图像（人眼不能分辨区别）

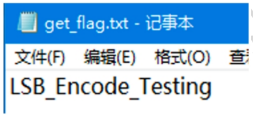


图: 写入的文本内容

LSB

用 Stegsolve 打开生成的隐写图片 out.png，并提取 R、G、B 三个通道的最低有效位，同样可以提取刚刚隐藏在图片中的字符串。
对于 PNG 和 BMP 图片中的 LSB 等常见的隐写方式，我们可以使用 zsteg 工具（https://github.com/zed-0xff/zsteg）进行自动化的识别和提取。

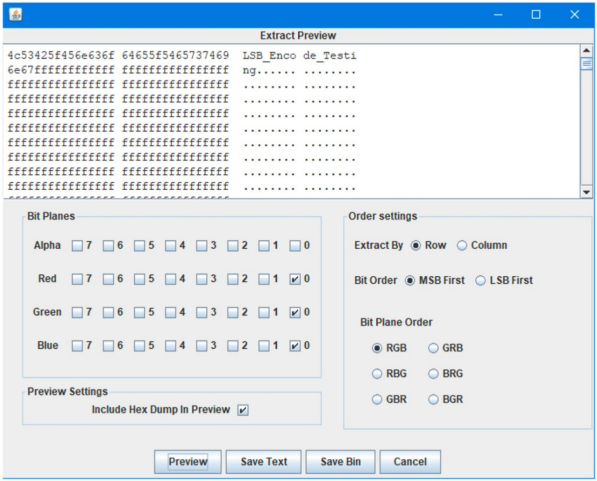


图: 通过 Stegsolve 读取 LSB 信息

盲水印

数字水印技术可以将信息嵌入图片、音频等数字载体中，但以人类的视觉或者听觉无法分辨，只有通过特殊的手段才能读取。图片中的盲水印可以添加在图片的空域或频域。盲水印的添加与提取可以使用如 BlindWaterMark (https://github.com/chishaxie/BlindWaterMark) 这样的工具。类似技术在音频中也常有应用，如音频中的频谱隐写，我们可以使用 Adobe Audition 等工具直接查看频谱从而获取隐藏信息（flag）。

频域介绍

什么是频域？频域是对信号的频率分析结果。要将空域中的信号转换到频域，就要用到傅里叶变换（Fourier Transform）。频域图像可以通过将水印编码后随即分布到各频率，与原图的频域进行叠加，然后将叠加的水印频谱进行傅里叶逆变换，即可得到添加了盲水印的图片。

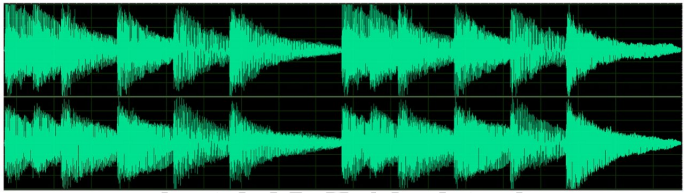


图: 频域图示例

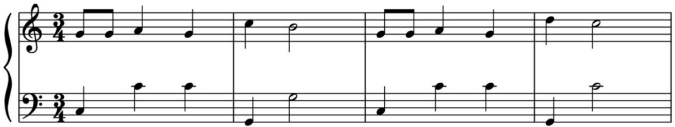


图: 频域对应的音符

隐写术小结

图片隐写的方式有很多种。广义上，只要通过某种方式将信息隐藏到图片中且难以通过普通方式发现，就可以称为图片隐写。本节对一些常见的图片隐写方式进行了简单介绍，读者在理解图片隐写常见的基本原理后，可自行尝试通过不同方式对图片进行隐写。

参考文献

[1] Nu1L. 从 0 到 1：CTFer 成长之路 [EB/OL]. https://book.douban.com/subject/35200558/.