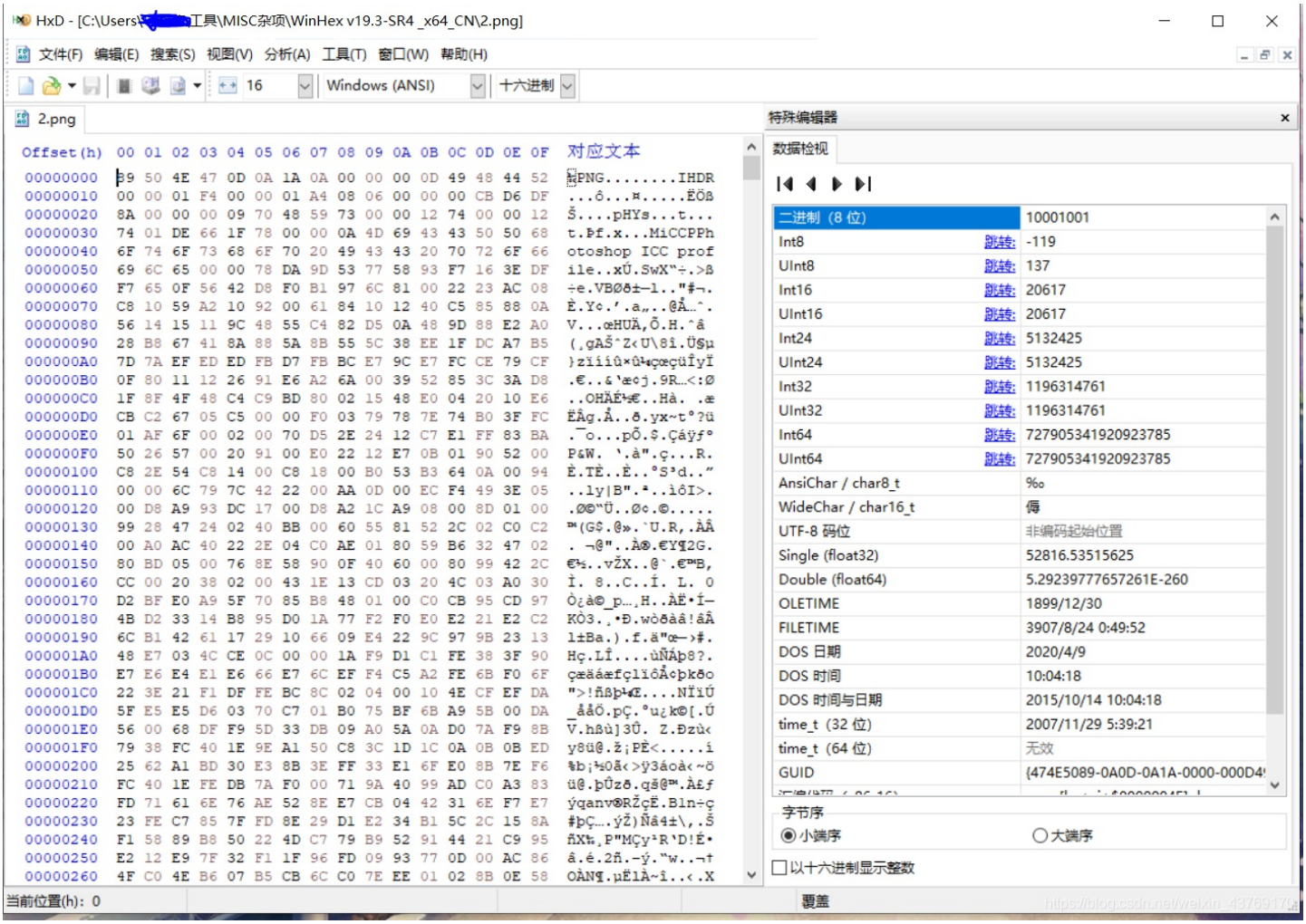
misc 隐写入门

原创

照旧的你好 于 2020-03-14 10:42:52 发布 159 收藏

用HxD打开该png文件



解这题首先要了解PNG的文件结构

文件头位置由固定字节描述：89 50 4E 47 0D 0A 1A 0A(16进制)，其中第一个字节0x89超出了ASCII字符的范围，这是为了避免某些软件将PNG文件当做文本文件来处理。文件中剩余的部分由3个以上的PNG的数据块（Chunk）按照特定的顺序组成。

PNG定义了两种类型的数据块，一种是称为关键数据块(critical chunk)，这是标准的数据块，另一种叫做辅助数据块(ancillary chunks)，这是可选的数据块。关键数据块定义了4个标准数据块，每个PNG文件都必须包含它们，PNG读写软件也都必须要支持这些数据块。虽然PNG文件规范没有要求PNG编译码器对可选数据块进行编码和译码，但规范提倡支持可选数据块。

。。。

根据图片格式, crc爆破
在hxd里修改，第二行的A4改成F4，重新打开显示图片。

```
89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52  %PNG........IHDR
00 00 01 F4 00 00 01 F4 08 06 00 00 00 CB D6 DF  ...ô...ô.....ËÖß
```

2.crc爆破

```python
import struct
import  binascii
import os

#根据图片中的CRC校验值，计算出正确的长和宽
fi=open(r'E:\python\crc.png','rb').read()

#12-15字节代表固定的文件头数据块的标示，16-19字节代表宽度，20-23字节代表高度，24-28字节分别代表
# Bit depth、ColorType、Compression method、Filter method、Interlace method
#29-32字节为CRC校验和

for i in range(10000):#宽度0-9999搜索
    for j in range(10000):
        data=fi[12:16]+struct.pack('>I',i)+struct.pack('>I',j)+fi[24:29]  #pack函数将int转为bytes,>表示大端00 00 00 02,I表示4字节无符号int;<表示小端 02 00 00 00
        crc=binascii.crc32(data)&0xffffffff #byte的大小为8bits而int的大小为32bits,转换时进行与运算避免补码问题0x932f8a6b
        if crc==struct.unpack('>I',fi[29:33])[0]&0xffffffff : #解开为无符号整数
            print("宽度为：",hex(i))
            print("长度为：",hex(j))
            os.kill()
```