

CTF-bugku-misc-[隐写3]-png高度隐写和pngCRC校验

原创

沧海一粟日尽其用 已于 2022-03-08 20:31:44 修改 143 收藏 1

文章标签: [web安全](#) [安全](#)

于 2022-03-08 20:22:25 首次发布

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/m0_43405474/article/details/123361993

版权

CTF-bugku-misc-[隐写3]-png高度隐写和pngCRC校验

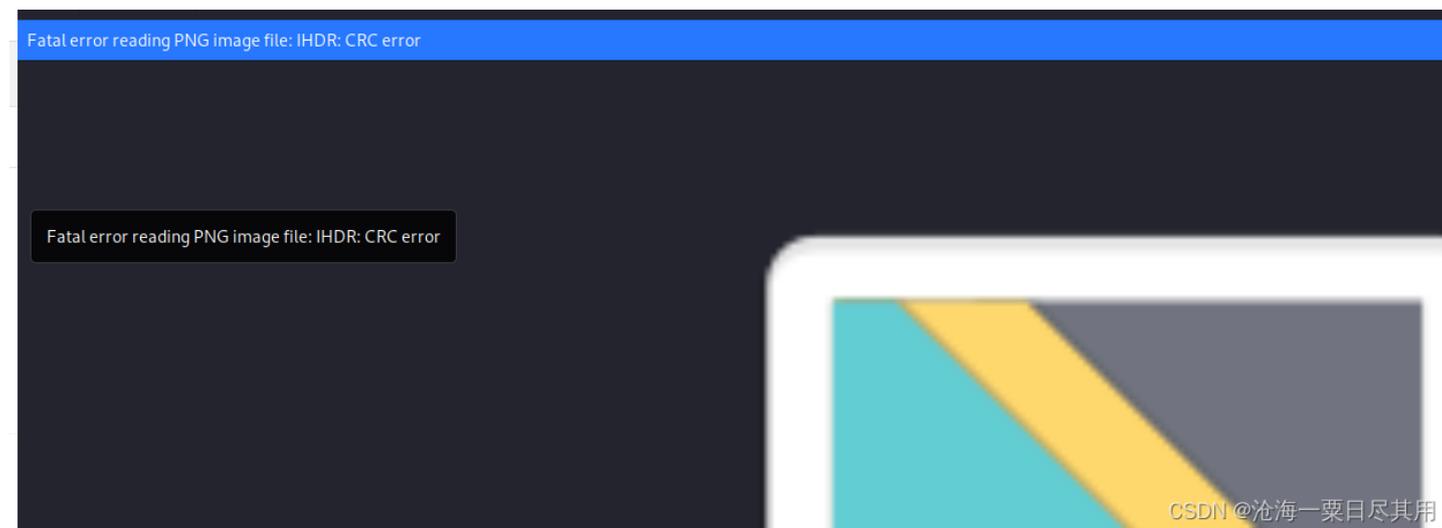
- 1.题目
- 2.png文件结构分析
- 3.改变图片的高度+CRC校验
- 4.修复图片获得flag

看很多教程都教直接改高度恢复图片拿到flag, 但是我尝试了很久, 并不奏效。特定记录了一下自己探索的过程。

1.题目

一张名为dabai的png图片, 但是不能正常打开。

打开的时候提示CRC校验错误:



思路: 恢复图片拿到flag

2.png文件结构分析

首先来详细的分析一下png的存储结构主要关注头部。

```
00000000h: 89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52 ; PNG.....IHDR
00000010h: 00 00 00 08 00 00 00 08 04 03 00 00 00 36 21 A3 ; .....6! ?
00000020h: B8 00 00 00 03 73 42 49 54 08 08 08 DB E1 4F E0 ; ?...sBIT...tEXt?
00000030h: 00 00 00 27 50 4C 54 45 FF FF 00 FF ED 00 FF C1 ; ...'PLTE . ? ?
00000040h: 00 FF 99 00 FF 66 00 FF 3B 00 FF 0F 00 E2 00 15 ; . ? f. ;. ..?.
00000050h: B7 00 34 8B 00 54 60 00 73 33 00 99 09 00 B2 5F ; ?4?T`.s3?.膜
00000060h: F5 BB DD 00 00 00 09 70 48 59 73 00 00 0B 12 00 ; 夏?...pHYs.....
00000070h: 00 0B 12 01 D2 DD 7E FC 00 00 00 20 74 45 58 74 ; ....逸~?... tEXt
00000080h: 53 6F 66 74 77 61 72 65 00 4D 61 63 72 6F 6D 65 ; Software.Macrome
00000090h: 64 69 61 20 46 69 72 65 77 6F 72 6B 73 20 4D 58 ; dia Fireworks MX
000000a0h: BB 91 2A 24 00 00 00 16 74 45 58 74 43 72 65 61 ; 舜*$....tEXtCrea
000000b0h: 74 69 6F 6E 20 54 69 6D 65 00 30 34 2F 30 31 2F ; tion Time.04/01/
000000c0h: 30 35 22 44 50 99 00 00 00 27 49 44 41 54 78 9C ; 05"DP?..'IDATx?
000000d0h: 63 38 BD B2 3D 95 61 D7 8C B2 10 06 20 C3 99 01 ; c8讲=弄讓?.膜.
000000e0h: C8 30 62 00 32 14 19 80 0C 01 06 10 83 01 C4 00 ; ?b.2.€?...??
000000f0h: 00 24 A7 0B A4 DA 12 06 A5 00 00 00 00 49 45 4E ; .$.?~.清风网络
00000100h: 44 AE 42 60 82 ; 随` http://www.vipcn.com
http://www.CSDN@沧海一粟白虎真用
```

- (1). png的文件头：8个字节 89 50 4E 47 0D 0A 1A 0A 为 png的文件头(固定)
- (2). 4个字节 00 00 00 0D (即为十进制的13)代表头部数据块的长度为13(固定)
- (3). 4个字节 49 48 44 52 (即为ASCII码的IHDR)是文件头数据块的标示(IDCH)(固定)
- (4). 13位数据块(IHDR)
前四个字节代表该图片的宽 00 00 00 08
后四个字节代表该图片的高 00 00 00 08
后五个字节依次为: Bit depth、ColorType、 Compression method、 Filter method、 Interlace method
(可变)
- (5). 剩余四字节为该png的CRC检验码 36 21 A3 B8，由从IDCH到THDR的十七位字节进行crc计算得到。(可变)

3.改变图片的高度+CRC校验

这里打开图片的时候CRC校验不过的话，多半是头部出了问题，CRC是由49 48 44 52 + 后面的13个字节共17个字节做CRC校验得到的。改变高度还不行，因为CRC校验依然不正确。

改完高度之后，要重新计算CRC校验码：
<http://www.ip33.com/crc.html> 【工具网址】

CRC (循环冗余校验) 在线计算

Hex

Ascii

校验文件

需要校验的数据:

49 48 44 52 00 00 02 a7 00 00 02 00 08 06 00 00 00

输入的数据为16进制, 例如: 31 32 33 34

参数模型 NAME:

CRC-32

$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4$

宽度 WIDTH:

32

多项式 POLY (Hex):

04C11DB7

例如: 3D65

初始值 INIT (Hex):

FFFFFFFF

例如: FFFF

结果异或值 XOROUT (Hex):

FFFFFFFF

例如: 0000

输入数据反转 (REFIN)

输出数据反转 (REFOUT)

计算

清空

校验计算结果 (Hex):

BFFCC552

复制

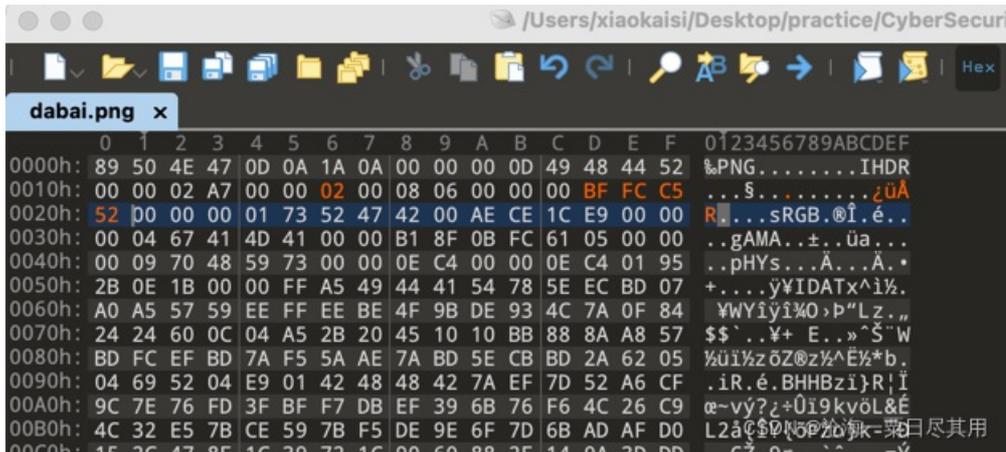
高位在左低位在右, 使用时请注意高低位顺序!!!

校验计算结果 (Bin):

1011111111111001100010101010010

复制

CSDN @ 沧海一粟日尽其用



总之, 高度可以改到足够显示出flag为止, 没有固定值, 只要保证你改完之后, 把相应的CRC值计算出来, 填在相应的位置上即可。

4. 修复图片获得flag



CSDN @沧海一粟日尽其用

flag{He110_d4_ba1}