BUGKU上几道有意思的题

原创

Sanky0u 于 2017-07-26 23:38:51 发布 3884 收藏

文章标签： ctf misc bugku

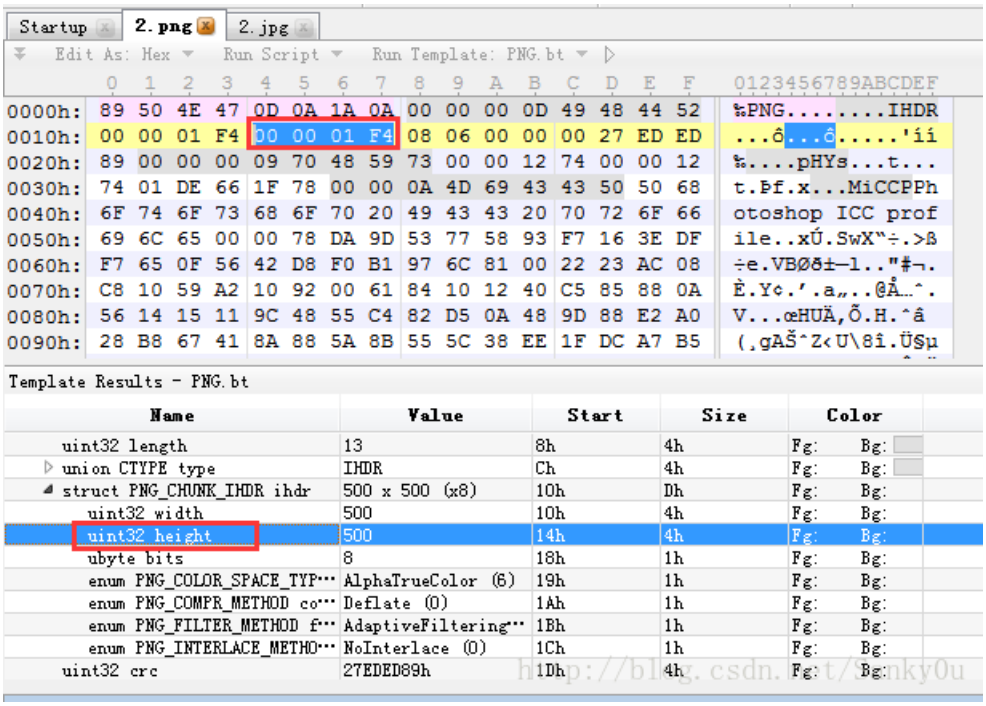
本文链接：https://blog.csdn.net/Sanky0u/article/details/76167670

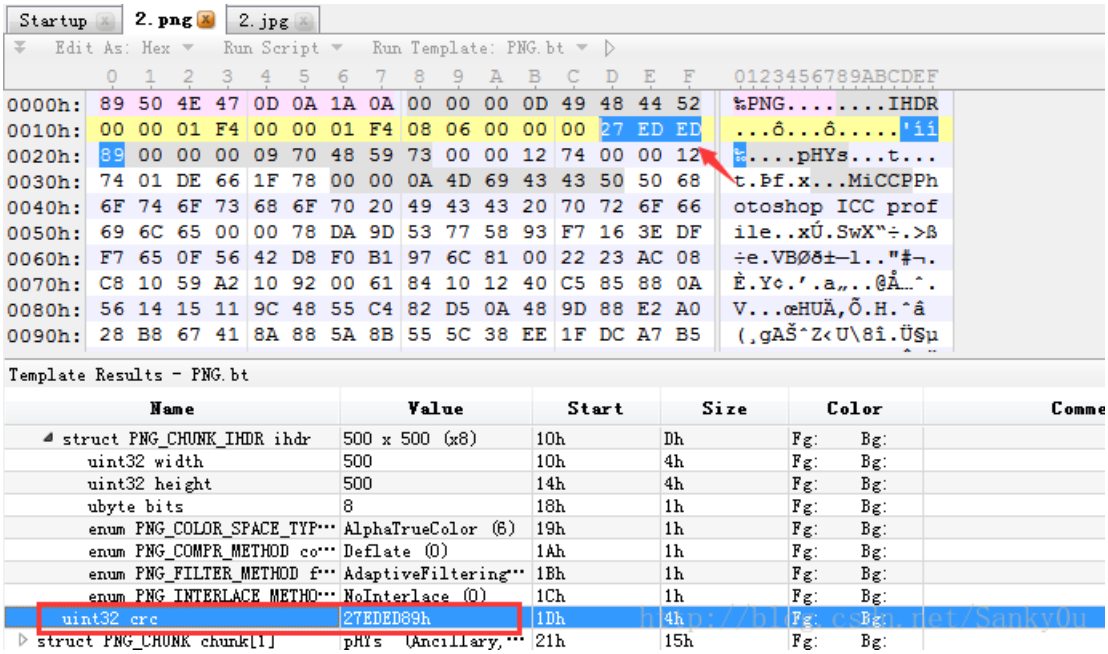
misc

隐写2

这道题真的很难想到竟然是改变了图片的高度，不知道有没有什么方法可以看出来，我是最后搜索了一下别人writeup，看到提示才知道的。

之后就用 010editor 打开 2.png 图片，这个软件特别好的是有png.bt插件，可以帮助看16进制的png图片的各个值都代表什么。





点开IHDR结构中，找到height那一行，选中后上面对应的16进制数颜色也会改变，改之前这个值是000001A4，也就是420，我把它改成和宽一样。这时如果保存再打开文件会提示文件损坏，是因为没有通过CRC校验，所以改完之后还要改这个值影响到的CRC校验值。

修改完高度值和CRC值之后再保存再打开就可以看到flag了。有篇 隐写技巧——利用PNG文件格式隐藏Payload 讲解了png文件格式和CRC计算代码，讲得很好。

下面附上那篇文章中的CRC值的计算代码（找代码的时候发现可能有好几种CRC校验值，但是能通过校验就可以了）：

```c
#include <stdio.h>
#include <string.h>

unsigned int GetCrc32(char* InStr,unsigned int len){
  unsigned int Crc32Table[256];
  int i,j;
  unsigned int Crc;
  for (i = 0; i < 256; i++){
    Crc = i;
    for (j = 0; j < 8; j++){
      if (Crc & 1)
        Crc = (Crc >> 1) ^ 0xEDB88320;
      else
        Crc >>= 1;
    }
    Crc32Table[i] = Crc;
  }

  Crc=0xffffffff;
  for(int m=0; m<len; m++){
    Crc = (Crc >> 8) ^ Crc32Table[(Crc & 0xFF) ^ InStr[m]];
  }

  Crc ^= 0xFFFFFFFF;
  return Crc;
}
int main(int argc, char* argv[])
{
    char buf[17]={0x49,0x48,0x44,0x52,0x00,0x00,0x00,0x1A,0x00,0x00,0x00,0x1A,0x08,0x04,0x00,0x00,0x00}
    unsigned int crc32=GetCrc32(buf,sizeof(buf));
    printf("%08X\n",crc32);
    return 0;
}
```

linux ??????

这题也很有意思，文件解压后的flag原来是一个linux的文件系统文件，可以用mount命令来将文件系统中的内容放入一个文件夹，然后从文件夹里可以看到。

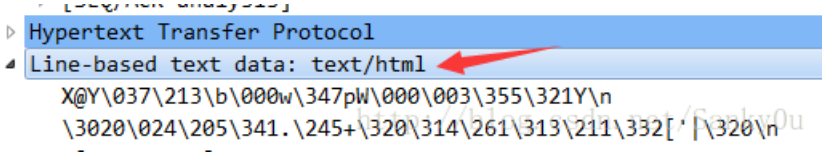
```
C:\Python27\Scripts>python binwalk E:\CTF题目\bugku\misc\1.tar\test\flag

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------

0             0x0             Linux EXT filesystem, rev 1.0, ext3 filesystem dat
a, UUID=66ce56f1-5b57-492f-82f3-ac0678797879
310272        0x4BC00         Linux EXT filesystem, rev 1.0, ext3 filesystem dat
a, UUID=66ce56f1-5b57-492f-82f3-ac0678797879
322560        0x4EC00         Linux EXT filesystem, rev 1.0, ext3 filesystem dat
a, UUID=66ce56f1-5b57-492f-82f3-ac0678797879
348160        0x55000         Linux EXT filesystem, rev 1.0, ext3 filesystem dat
a, UUID=66ce56f1-5b57-492f-82f3-ac0678797879
357376        0x57400         Linux EXT filesystem, rev 1.0, ext3 filesystem dat
a, UUID=66ce56f1-5b57-492f-82f3-ac0678797879
374784        0x5B800         Linux EXT filesystem, rev 1.0, ext3 filesystem dat
a, UUID=66ce56f1-5b57-492f-82f3-ac0678797879
8388608       0x800000        Linux EXT filesystem, rev 1.0, ext3 filesystem dat
a, UUID=66ce56f1-5b57-492f-82f3-ac0678797879http://blog.csdn.net/Sanky0u
```

之后将这个文件放入linux系统中执行如下命令，每条命令的执行结果我记不太清了，只记得最后test文件夹中有flag.txt 和lost+found两个文件，flag.txt中存放的就是flag。

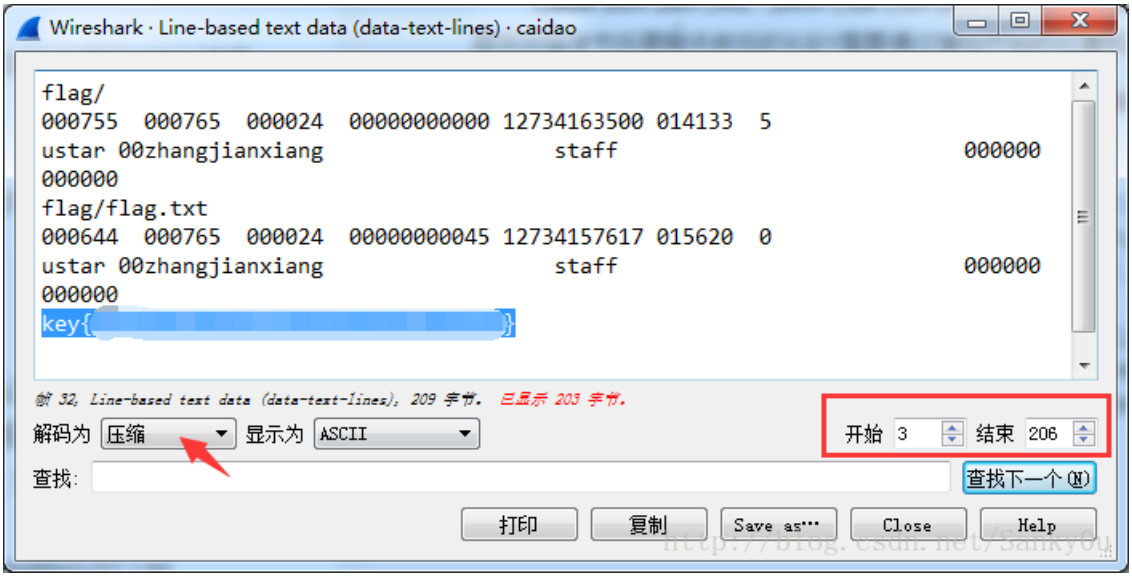
```
#file flag
#mkdir test
#mount flag test
#cd test
#ls
flag.txt lost+found
```

中国菜刀，不再web里？

开始思路是菜刀连接之前会先上传木马，然后再用菜刀连接。后面找到了3.php，但是没有没有找到flag。最后参考了http://www.bugku.com/thread-11-1-1.html，才知道在哪里。
因为博主已经写得很清楚了，我就补充一点细节吧。
要显示全部对应的全部分组字节要点击箭头所指的那一行



显示分组字节后要删去前后的X@Y需要通过增加开始的位置和减少结束的位置来实现。



之所以解码为压缩是因为读取的是flag.tar.gz文件的内容，解压缩后才能看到里面的明文内容。很神奇吧！
还有下面两题也很有意思：
1.misc里面的《宽带信息泄露》，用http://www.pc841.com/article/20150204-42006.html 中的方法可以解决，而且提供了工具，不过打开conf.bin之后看到的是一段xml代码，和链接样例显示的不一样，但是不影响找到用户名，搜索“username”就可以。
2.Crypto中的《一段base64》，除了中间涉及到各种编码可以用Converter解密之外，http://www.admin5.com/article/20080310/75252.shtml 中提到的这些编码在网页挂马中的利用也是很眼前一亮，也许是我接触得太少，不过没关系，慢慢积累，今天收获很多，开心！