MISC总结——隐写术(三）

转载

weixin_34119545 于 2018-07-30 21:35:00 发布 405 收藏 4
文章标签： 网络 python photoshop
原文链接：https://yq.aliyun.com/articles/649048

这一篇是继上一篇之后的另一篇关于隐写术在ctf比赛中常见的套路问题：

MISC总结——隐写术(一)详情请见：https://www.cnblogs.com/lxz-1263030049/p/9388511.html

MISC总结——隐写术(二)详情请见：https://www.cnblogs.com/lxz-1263030049/p/9388602.html

本文参考自：先知社区：https://xz.aliyun.com/t/1844

隐写术介绍：

隐写术是关于信息隐藏，即不让计划的接收者之外的任何人知道信息的传递事件（而不只是信息的内容）的一门技巧与科学。

英文写作Steganography，而这篇内容将带大家了解一下CTF赛场上常见的图片隐写方式，以及解决方法。有必要强调的是，隐写术与密码编码是完全不同的概念。

第一部分：基于LSB原理的图片隐写

首先：我们需要明白什么是LSB隐写：

LSB，最低有效位，英文是Least Significant Bit 。我们知道图像像素一般是由RGB三原色（即红绿蓝）组成的，

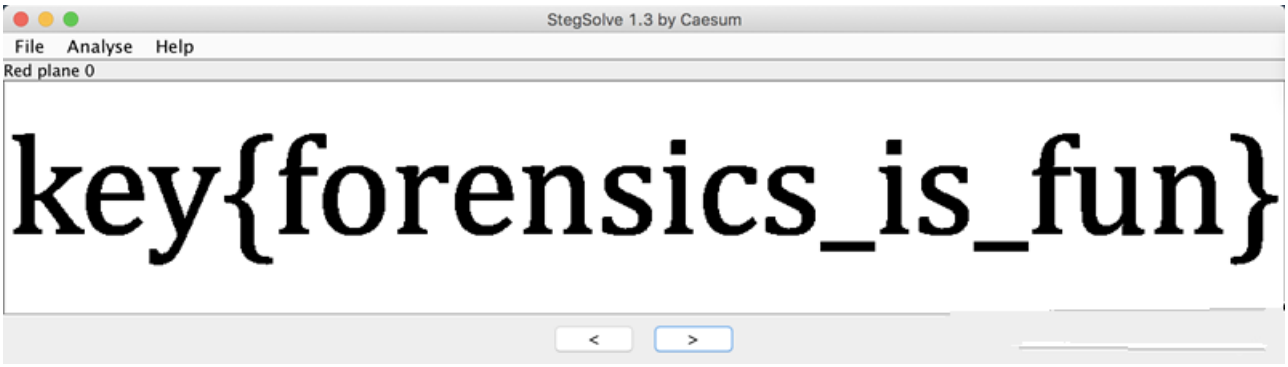
每一种颜色占用8位，0x00~0xFF，即一共有256种颜色，一共包含了256的3次方的颜色，颜色太多，而人的肉眼能区分的只有其中一小部分，

这导致了当我们修改RGB颜色分量种最低的二进制位的时候，我们的肉眼是区分不出来的。

Stegosolve介绍 ：
CTF中，最常用来检测LSB隐写痕迹的工具是Stegsolve，这是一款可以对图片进行多种操作的工具，包括对图片进行xor,sub等操作，对图片不同通道进行查看等功能

在实验中找到隐写术目录，打开图片隐写，打开图片隐写第三部分文件夹
在该文件夹找到chal.png
双击打开图片，我们先确认一下图片内容并没有什么异常
使用Stegsolve打开图片，在不同的通道查看图片
在通道切换的过程中，我们看到了flag
最后的flag是flag:key{forensics_is_fun}

用Stegsolve打开图片，并在不同的通道中切换：



最后得到flag:key{forensics_is_fun}

思考的问题：

1. 我们如何实现这种LSB隐写的？是否可以通过photoshop这样的工具实现？
2. 查阅更多关于LSB隐写的资料。

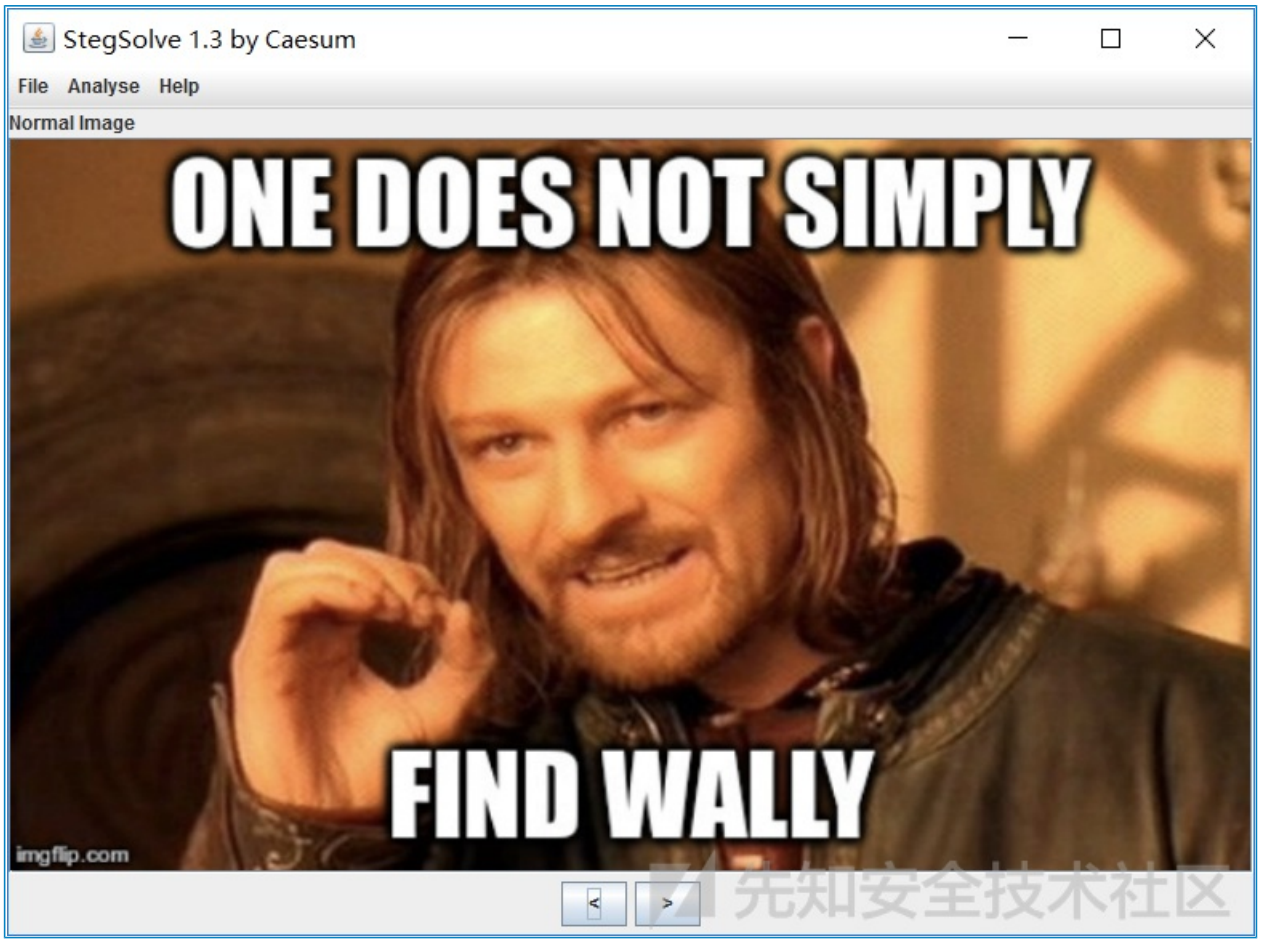
有难度的LSB隐写：

我们从第一个部分可以知道，最简单的隐写我们只需要通过工具Stegsolve切换到不通通道，我们就可以直接看到隐写内容了，

那么更复杂一点就不是这么直接了，而是只能这样工具来查看LSB的隐写痕迹，再通过工具或者脚本的方式提取隐写信息。

在实验中找到隐写术目录，打开图片隐写，打开图片隐写第三部分文件夹 在该文件夹找到LSB.bmp
双击打开图片，我们先确认一下图片内容并没有什么异常
使用Stegsolve打开图片，在不同的通道查看图片
在通道切换的过程中，来判断隐写痕迹
编写脚本提取隐写信息。
最后打开提取后的文件，得到flag

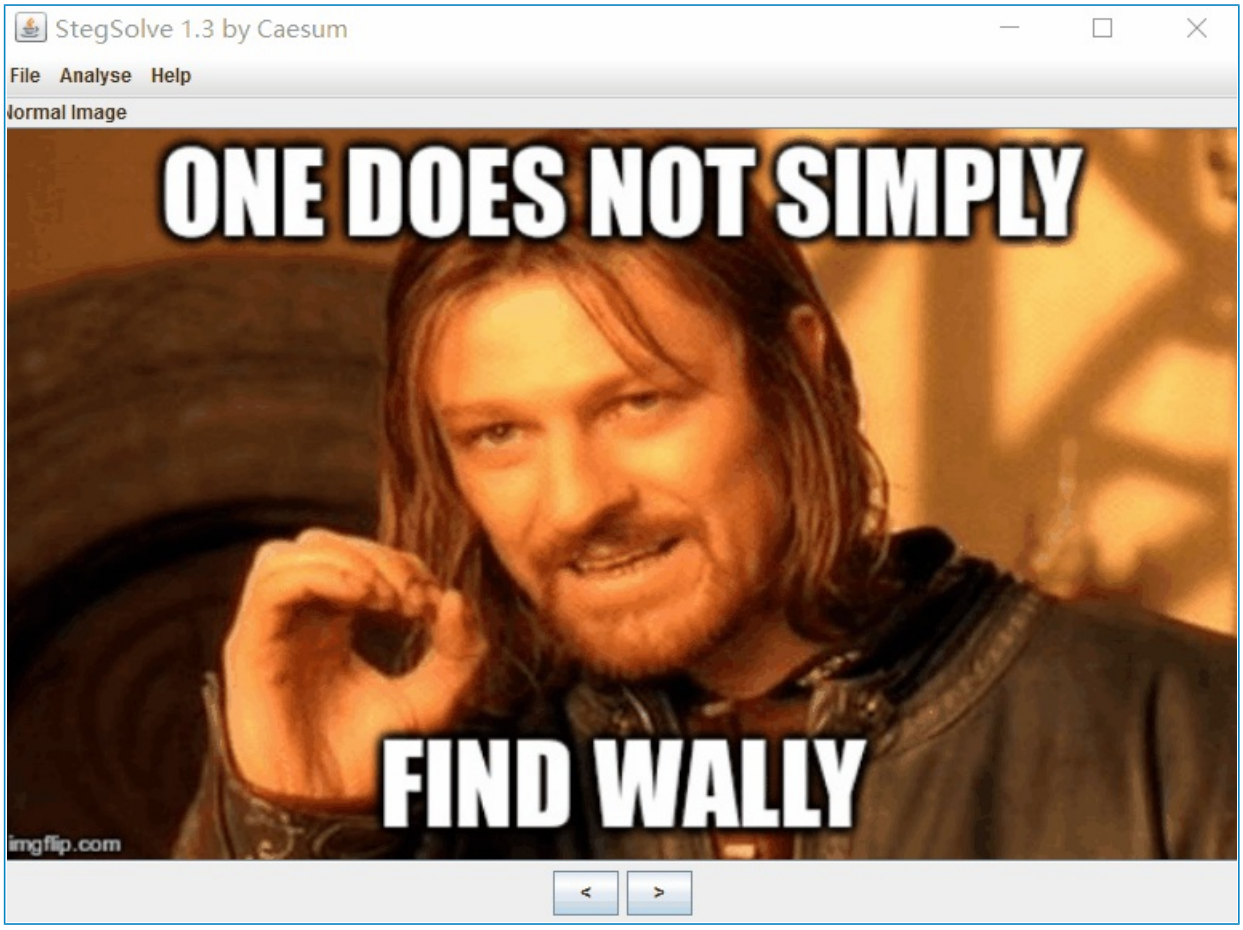
首先：从Stegsolve中打开图片
首先点击上方的File菜单，选择open，在题目文件夹中找到这次所需要用的图片whereswaldo.bmp



其次：切换到不同通道，通过痕迹来判断是否是LSB隐写

分析是否有可能是LSB隐写，我们开始点击下面的按钮，切换到不同通道，我们逐渐对比不同通道我们所看到的图片是怎么样子的。

我们发现在Red plane0和Greee plane 0以及B略 plane 0出现了相同的异常情况，我们这里基本可以断定就是LSB隐写了



编写代码提取信息

因为是LSB隐写，我们只按位提取RGB的最低位即可，代码如下：

```
from PIL import Image

im = Image.open("extracted.bmp")
pix = im.load()
width, height = im.size

extracted_bits = []
for y in range(height):
    for x in range(width):
        r, g, b = pix[(x,y)]
        extracted_bits.append(r & 1)
        extracted_bits.append(g & 1)
        extracted_bits.append(b & 1)

extracted_byte_bits = [extracted_bits[i:i+8] for i in range(0, len(extracted_bits), 8)]
with open("extracted2.bmp", "wb") as out:
    for byte_bits in extracted_byte_bits:
                byte_str = ''.join(str(x) for x in byte_bits)
        byte = chr(int(byte_str, 2))
        out.write(byte)
```

打开我们需要提取信息的的图片，y，x代表的是图片的高以及宽度，进行一个循环提取。
运行代码，extracted.py，打开图片即可。

思考的问题：

1. 我们这里用的LSB隐均对R,G,B，三种颜色都加以修改是否可以只修改一个颜色？
2. 参考2016 HCTF的官方Writeup学习如何实现将一个文件以LSB的形式加以隐写。

题目.zip (0.624 MB) 下载附件

作者： 落花四月
出处： https://www.cnblogs.com/lxz-1263030049/
关于作者：潜心于网络安全学习。如有问题或建议，请多多赐教！

力！