【隐写|CTF】算法隐写术——LSB 最低有效位隐写

原创

殷阳 于 2020-08-05 21:30:12 发布 1647 收藏 8

分类专栏： # 安全杂项 文章标签： 经验分享

本文链接：https://blog.csdn.net/a714804968/article/details/107824797

介绍

LSB 隐写术，即最低有效位（Least Significant Bit）隐写术。是一种比较简单的基于二进制的隐写方式，能将一个完整的文件隐藏进另一张图片中。图像一般都是以RGB三原色的方式存储的，存储后R（红）、G（绿）、B（蓝）三组数据按顺序规律排列，每个的取值范围为0~255，范围对应的二进制值就是00000000-11111111，LSB就是把一个文件的二进制每一位拆分修改到图片的色彩数据的二进制值最低位。

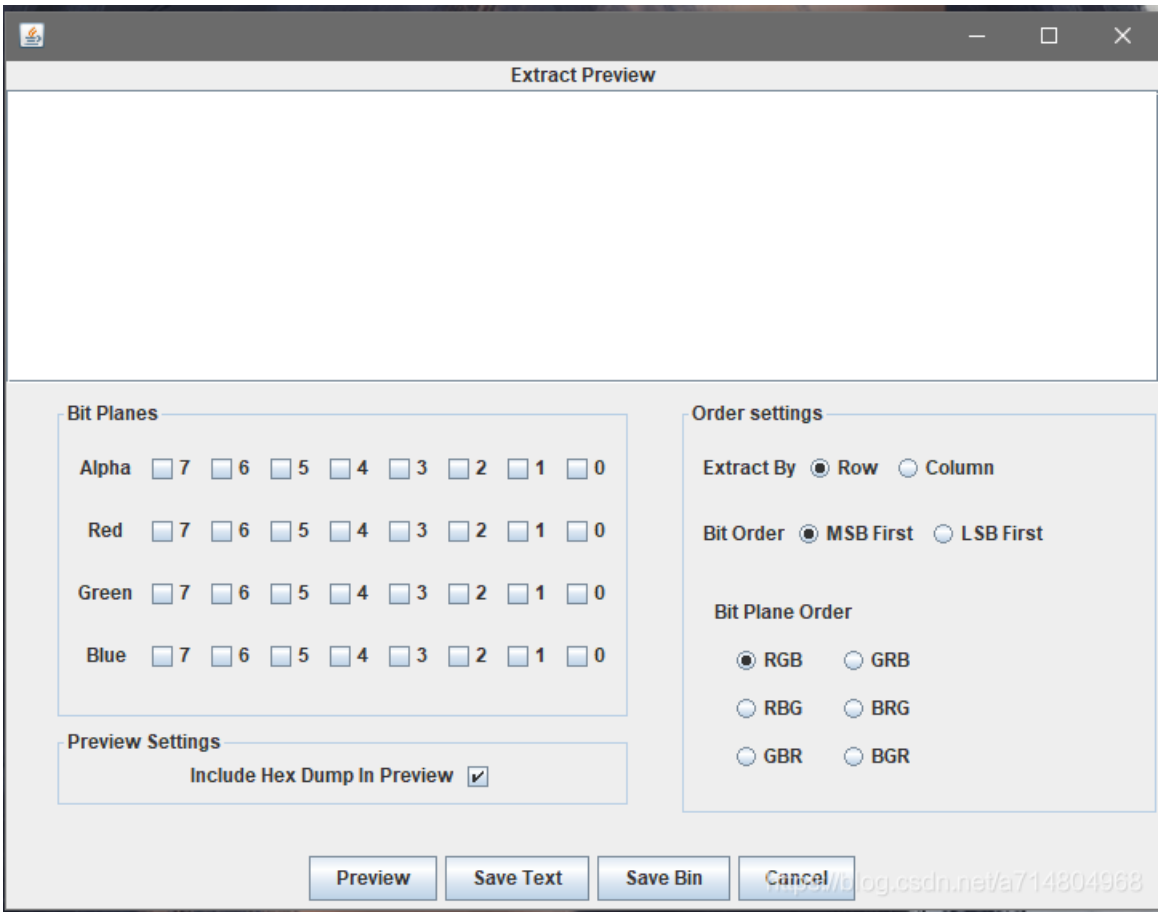
当隐藏信息的载体图片被按照LSB算法修改后，对于文件本身，只是色彩数据每个字节的十进制值存在一个+1或者0的变化，对应色彩的变化更是小到肉眼无法发现。举例最简单的纯红色（RGB 255,0,0），在红色位置修改最低位二进制值后，无明显变化：

注意

- 该隐写方式一般只能用于PNG、BMP这类无损格式，像JPEG这种有损格式是不会有的。（**注意**：判断格式不能看文件扩展名，扩展名并不完全能代表一个图片的格式，需要看文件的二进制数据头）：
    1. **PNG**：89 50 4E 47 0D 0A 1A 0A（文件头）
    2. **BMP**：42 4D （B M）（文件头）
    3. **JPEG**：FF D8（文件头）、FF D9（文件尾）
- 该隐写方式在CTF中属于一种比较常见的隐写方式，当你通过二进制查看原图无法发现线索时，就可以考虑算法隐写，就可以试着通过LSB提取信息，判断是否有效。

工具

StegSolve可以直接对图片进行分离，操作方式：Analyse→Data Extract→选择你要导出的通道→Save bin，一般保存后你就可以得到隐藏的文件信息了。



CSDN下载：StegSolve.jar