

LSB隐写总结

原创

[坚强的女程序员](#) 于 2018-02-14 11:18:15 发布 12541 收藏 10

分类专栏: [CTF](#) 文章标签: [LSB](#) [CTF](#)

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/qq_33438733/article/details/79324763

版权



[CTF 专栏收录该内容](#)

61 篇文章 4 订阅

订阅专栏

前言

又是一年情人节!

单身狗表示不关我事.- 写篇博文祝贺庆祝下。

一、LSB简介

LSB(英文 least significant bit)即最低有效位。LSB加密是信息隐藏中最基本的方法。由于人们识别声音或图片的能力有限, 因此我们稍微改动信息的某一位是不会影响我们识别声音或图片的。

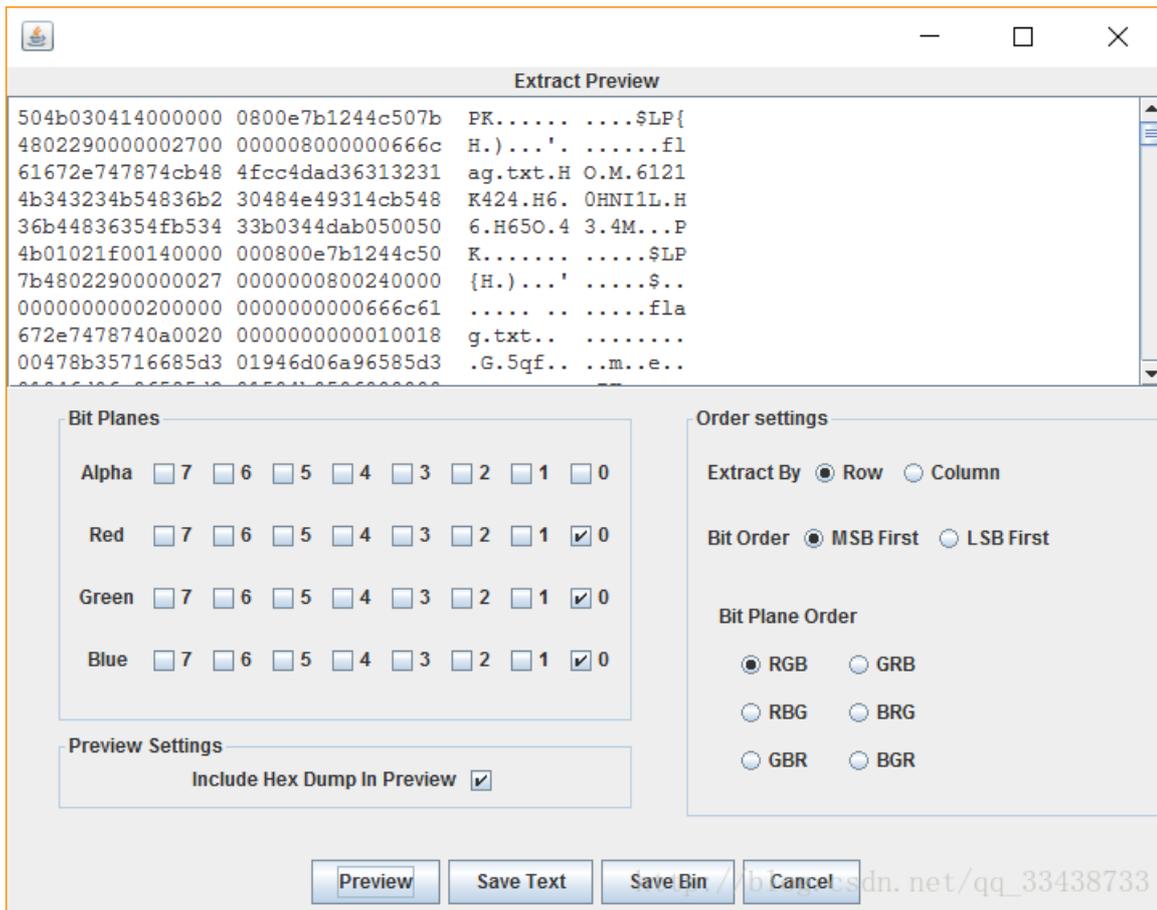
二、用法

通常来说LSB加密用在无损压缩的数据格式文件中, 例如图像中的bmp格式和音频的wav格式。由于这两种格式未对源数据进行有损压缩, 因此可以将信息隐藏起来。

BMP文件中的使用

对于图像文件LSB的特征很明显，通常将信息隐藏在某一个颜色通道中。我们可以查看图片的每个像素点的RGB值，或者使用stegsolve工具进行查看。

由于图像是由像素构成的，每个像素有8位(对于BMP图像来说),通常最后一位的变化，通过肉眼是无法察觉的



MSB是最高有效位，Bit Plane Order代表颜色的排序方式

WAV文件中的使用

对于WAV文件来说LSB的隐藏没有那么明显，但我们同样可以类比BMP图像中的方法，这里我那国外的一道隐写题为例。

题目文件

首先说明一点，这题使用ruby解的，所以最好先去了解一下ruby语言。

主要利用的就是ruby中的wav-file模块。

基本上看懂wav-file模块就可以了。

首先我们读取文件，利用模块查看文件的基本信息

```
f = open("input.wav")
format = WavFile::readFormat(f)
dataChunk = WavFile::readDataChunk(f)
f.close
```

基本信息:

```
Format ID: 1
Channels: 2
Sampling Ratio: 44100 (Hz)
Byte per Sec: 176400
Bit per Sample: 16
Block Size: 4
```

根据提示可知使用"s*"进行解包

```
bit = 's*' if format.bitPerSample == 16 # int16_t
bit = 'c*' if format.bitPerSample == 8 # signed char
wavs = dataChunk.data.unpack(bit) # read binary
```

解包后使用map进行字符串的拼接，对于WAV文件来说LSB位于第一位

```
wavs = wavs.map{|w| w[0]}.join
```

最后在对得到的数据重新进行打包

```
flag=[wavs].pack("b**")
```

最后就可得到想要的东西了。!

ruby我懂的也不是很多,主要是python的wave模块的处理没有ruby处理来的直接,因此这里采用ruby,大家觉得自己python学的不错,完全可以用python来解决

总结

祝大家情人节快乐! 单身狗的忧伤。-