# LSB低位隐写（菜鸡理解）

原创

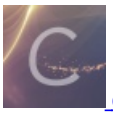
小元砸 于 2020-12-09 23:18:12 发布 906 收藏 3

分类专栏： ctf



本文链接：https://blog.csdn.net/qq_49480008/article/details/110943032







## LSB低位隐写

近期新学会的加密方式（不愧是我）

## LSB低位隐写原理

基于不可感知的要求，即数据的变化几乎不会引起使用的者的察觉，将水印信息嵌入到数据的最低有效位（Least Significant Bit），也就是将图片的RGB数值转换为二进制数据，然后用水印替换掉最低位，这种变化对于人眼来说是不可察觉的。当然，水印的形式也是多种多样的，有图片，文字等。这里主要讲文字（图片还是没搞太懂，之后肯定会学会的，立个flag！！）

## 注意！！

1.预先下的包（py代码解决）
from PIL import Image
2.LSB低位隐写一般用于bmp和png图片

## 思路

（1）获取要隐藏的数据，一般这里不管是什么，我们都可以理解为字符串。
（2）将获取到的字符串二值化，即按照一定规则转换为二进制数据，一般是8位（不涉及中文隐写，中文占2个），不够的前面补0，一定要测试好对应的解码方法。
（3）准备好宿主图像，安装好python环境和PIL。
（4）获取图像信息（主要是高度和宽度），根据二值化后的字符串的长度，对宿主图像的像素进行遍历，然后将数据依次写入对应像素的最低有效位，写入完成之后跳出循环，对目标图像进行持久化即可得到载密图像。

## 代码

```
120
121
122
123
124
125
#coding=utf-8
#coding=utf-8
try:
    from PIL import Image
```

```
    from PIL import ImageFile
except:
    import os
    os.system('pip install Pillow')
    from PIL import Image
    from PIL import ImageFile

ImageFile.LOAD_TRUNCATED_IMAGES=True

def full_eight(str):
    return str.zfill(8)
def get_text_bin(strr):
    string=""
    s_text=strr.encode()
    for i in range(len(s_text)):
        string=string+full_eight(bin(s_text[i]).replace('0b',''))
    return string
def mod(x,y):
    return x%y
def tell_you_bad(str1,str2,str3):
    im=Image.open(str1)
    width=im.size[0]
    height=im.size[1]
    count=0
    key=get_text_bin(str2)
    keylen=len(key)
    for h in range(0,height):
        for w in range(0,width):
            pixel=im.getpixel((w,h))
            a=pixel[0]
            b=pixel[1]
            c=pixel[2]
            if count==keylen:
                break
            a=a-mod(a,2)+int(key[count])
            count+=1
            if count==keylen:
                im.putpixel((w,h),(a,b,c))
                break
            b=b-mod(b,2)+int(key[count])
            count+=1
            if count==keylen:
                im.putpixel((w,h),(a,b,c))
                break
            c=c-mod(c,2)+int(key[count])
            count+=1
            if count==keylen:
                im.putpixel((w,h),(a,b,c))
                break
            if count%3==0:
                im.putpixel((w,h),(a,b,c))
    im.save(str3)
def tell_you_fun(le,str1):
    a=""
    b=""
    im = Image.open(str1)
    lenth = le*8
    width = im.size[0]
    height = im.size[1]
    count = 0
```

```
    for h in range(0, height):
        for w in range(0, width):
            pixel = im.getpixel((w, h))
            if count%3==0:
                count+=1
                b=b+str((mod(int(pixel[0]),2)))
                if count ==lenth:
                    break
            if count%3==1:
                count+=1
                b=b+str((mod(int(pixel[1]),2)))
                if count ==lenth:
                    break
            if count%3==2:
                count+=1
                b=b+str((mod(int(pixel[2]),2)))
                if count ==lenth:
                    break
        if count == lenth:
            break
    st=""
    for i in range(0,len(b),8):
        stra = int(b[i:i+8],2)
        st+=chr(stra)
    return st
def main():

    print("加密(1) OR 解密(2):",end=' ')
    choice=int(input())
    if choice==1:
        try:
            print("[+]加密图片:",end=' ')
            old=input()
            print("[+]加密文字(以#号结束):",end=' ')
            kkk=input()
            print("[+]加密图片保存重命名:",end=' ')
            new=input()
            tell_you_bad(old,kkk,new)
            print("[+]Fun Picture done!")
        except:
            print("[-]未找到此图片，请检查图片名和路径")

    if choice==2:
        le = 30
        try:
            print("[+]解密图片:",end=' ')
            new = input()
            word=tell_you_fun(le,new).split('#')
            print('[+]Picture Tell You: ',word[0])
        except:
            print("[-]未找到此图片，请检查图片名和路径")
if __name__=="__main__":
    main()
```

## 总结

1.这种方法隐写比较容易发现
2.加密过程有点难理解

写的还是比较粗糙，会慢慢添加的