BUUCTF Misc 二维码

原创

神音sss 于 2021-04-28 22:43:52 发布 353 收藏 3

分类专栏： CTF

本文链接：https://blog.csdn.net/oxygensss/article/details/116244067

题目：

下载下来后有一张二维码图片

把图片放进stegslove中，然后什么都没有发现。
所以估计里面有压缩包，用Kali分离出来
1.分解压缩包 binwalk -e QR_code.png

```
cd 123
```

```
binwalk -e QR_code.png
```

要进入文件所在的目录才会提取成功

```
root@oxygen:~# cd 123
root@oxygen:~/123# binwalk -e QR_code.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 280 x 280, 1-bit colormap, non-interlac
ed
471           0x1D7           Zip archive data, encrypted at least v2.0 to extra
ct, compressed size: 29, uncompressed size: 15, name: 4number.txt
650           0x28A           End of Zip archive, footer length: 22
```



```
dd if=QR_code.png of=flag.zip skip=471 bs=1
```

dd分离隐形文件

dd if=QR_code.png of=flag.zip skip=471 bs=1 //flag.zip解压时需要密码

if=file：输入文件名，缺省为标准输入

of=file：输出文件名，缺省为标准输出

skip=blocks：从输入文件开头跳过 blocks 个块后再开始复制

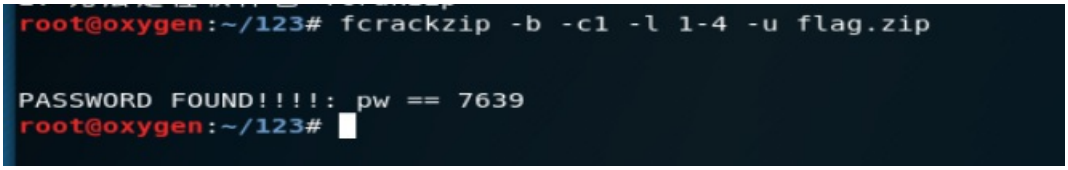
bs=bytes：同时设置读写块的大小为 bytes ，可代替 ibs 和 obs

```
请尝试执行 "dd --help" 来获取更多信息。
root@oxygen:~/123# dd if=QR_code.png of=flag.zip skip=471 bs=1
记录了201+0 的读入
记录了201+0 的写出
201 bytes copied, 0.000696919 s, 288 kB/s
root@oxygen:~/123#
```

```
root@oxygen:~/123# ls
flag.zip  _QR_code.png-0.extracted  _QR_code.png.extracted
QR_code.png  _QR_code.png-1.extracted
root@oxygen:~/123#
```

找到压缩包，点开发现需要密码

进行暴力破解



OK

**

binwalk的常见其他命令：

**

分解某种类型的文件 -D 或者 -dd

```
binwalk -D=png QR_code.png
```

```
root@oxygen:~/123# binwalk -D=png QR_code.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 280 x 280, 1-bit colormap, non-interlaced
471           0x1D7           Zip archive data, encrypted at least v2.0 to extract,
 compressed size: 29, uncompressed size: 15, name: 4number.txt
650           0x28A           End of Zip archive, footer length: 22
```

.-M 递归分析扫描出来的文件（可以跟-e -D 配合使用）

```
binwalk -eM QR_code.png
```

```
root@oxygen:~/123# binwalk -eM QR_code.png

Scan Time:     2021-04-28 22:01:58
Target File:   /root/123/QR_code.png
MD5 Checksum:  c552424a0e73a0614c41a4ae0cc63b7f
Signatures:    386

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 280 x 280, 1-bit colormap, non-interlaced
471           0x1D7           Zip archive data, encrypted at least v2.0 to extract,
 compressed size: 29, uncompressed size: 15, name: 4number.txt
650           0x28A           End of Zip archive, footer length: 22


Scan Time:     2021-04-28 22:01:58
Target File:   /root/123/_QR_code.png-1.extracted/4number.txt
MD5 Checksum:  d41d8cd98f00b204e9800998ecf8427e
Signatures:    386

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

root@oxygen:~/123#
```


使用fcrackzip，暴力破解压缩包

下载安装

```
sudo apt-get install fcrackzip
```

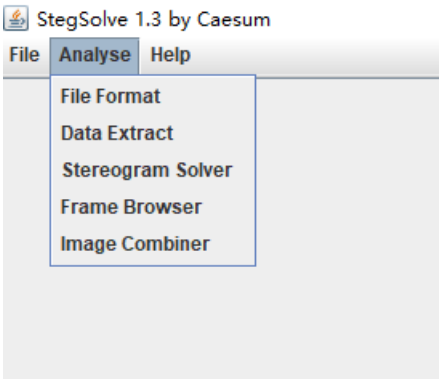
```
fcrakczip -b -c1 -l 1-4 -u flag.zip
```

c后面那个是数字1，在后面那个是字母l ,看起来有点像

-b 暴力破解模式 -c 指定掩码类型（a=a-z;1=0-9;!=特殊字符） -l 密码长度 -u 压缩文件名

fcrackzip -b -c1 -l 4 -u flag.zip

但备注一下stegslove的用法：



在分析里面从上到下的依次意思是

File Format:文件格式

Data Extract:数据提取

Steregram Solve:立体试图 可以左右控制偏移

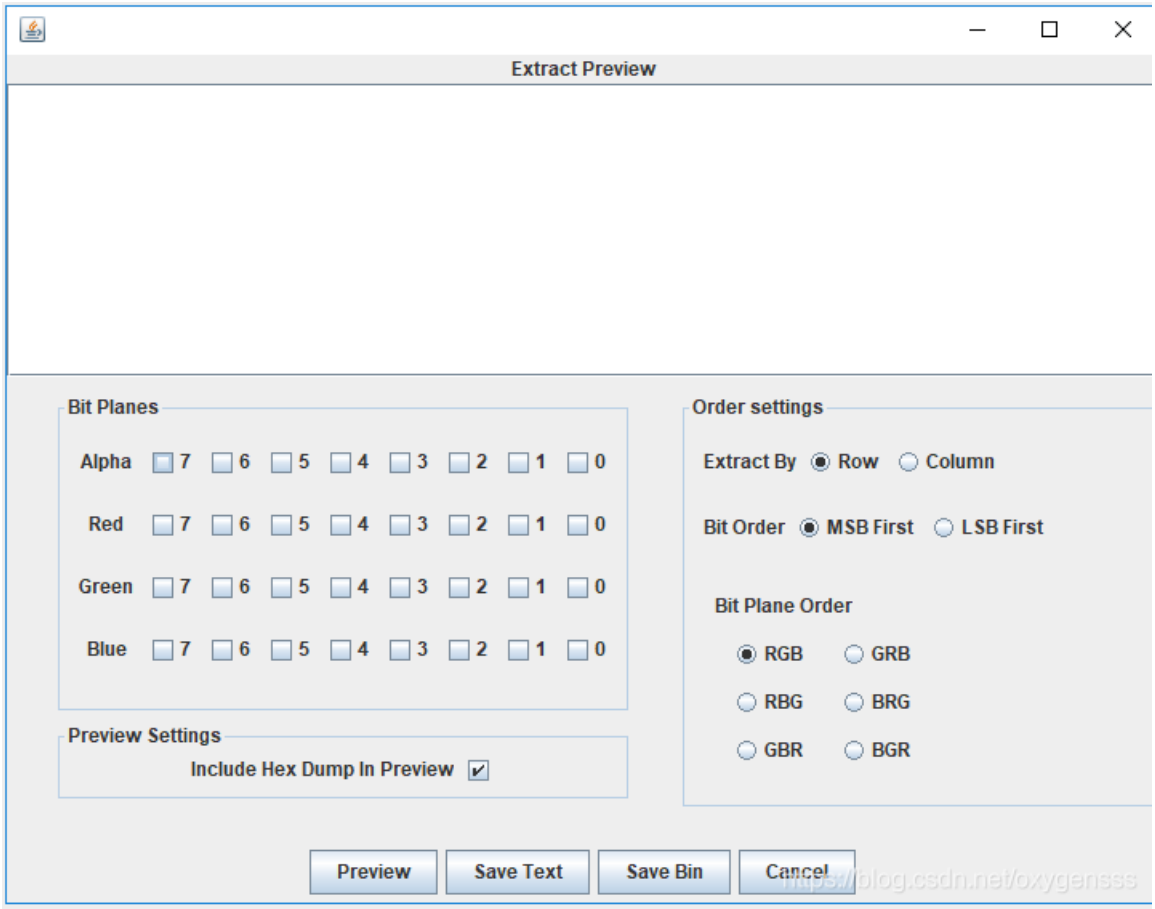
Frame Browser:帧浏览器

Image Combiner:拼图，图片拼接

用法：

1.File Format:这里你会看见图片的具体信息有时候有些图片隐写的flag会藏在这里

2.Data Extrac



左边一大部分RGB是红绿蓝 但他们的值代表的实际上是亮度

R的数字越大，则代表红色亮度越高；R的数字越小，则代表红色亮度越低。G，B同理

R的亮度各有256个级别，GB同理。即从0到255，合计为256个。从数字0到255的逐渐增高，我们人眼观察到的就是亮度越来越大，红色、绿色或蓝色越来越亮。然而256是2的8次方 所以你会看见上图的7~0 一共8个通道

而Alpha就是透明度 该通道用256级灰度来记录图像中的透明度信息，定义透明、不透明和半透明区域

alpha的值为0就是全透明，alpha 的值为 255 则表示不透明

因此左半部分就理解了

右半部分就是Extra By(额外的)和Bit Order（位顺序）和Bit Plane Order（位平面的顺序）

1）.Extra By(额外的)：分为row（行）和column（纵）

每个像素用R，G，B三个分量表示，那么一张图片就像一个矩阵，矩阵的每个单位就是（0_{255}, $_0255$，0~255）

也就会有是纵排列和行排列了，一般事先访问行再访问列（如果相反会引起ve使用方法）

2）.Bit Order（位顺序）:MSB是一串数据的最高位，LSB是一串数据的最低位。

3）.Bit Plane Order（位平面的顺序)

整个图像分解为8个位平面，从LSB(最低有效位0)到MSB（最高有效位7）随着从位平面0 到位平面7，位平面图像的特征逐渐变得复杂，细节不断增加。（一般我们的图片如果是RGB那么就是24位 3乘8嘛）

4）Bit Plane Order（位平面的顺序）:一般图片是24位 也就是3个8 大家可以想像成三明治 比如BGR就是B为三明治第一层 G为第二层 R为第三层。

3.Steregram Solve:立体试图 可以左右控制偏移 可以放张图片试一下就知道这个是什么意思了

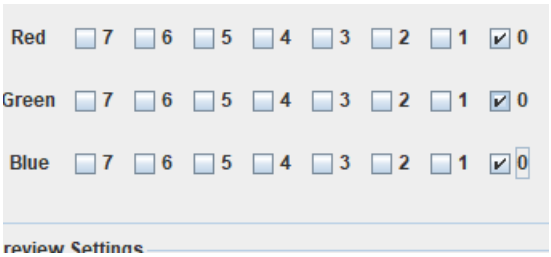
4.Frame Browser:帧浏览器 主要是对GIF之类的动图进行分解，把动图一帧帧的放，有时候会是二维码

5.Image Combiner:拼图，图片拼接（意思显而易见）

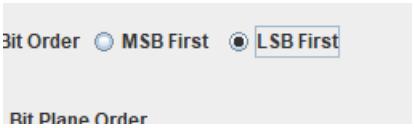
接下来会带大家实战去深入理解一下Data Extract里面ctf经常用到的LSB隐写主要是讲了RGBA（Alpha是透明度）的颜色通道

而LSB隐写就是修改RGB颜色分量的最低二进制位也就是最低有效位（LSB），而人类的眼睛不会注意到这前后的变化，（人类的眼睛只能识别一部分颜色的变化）

如果我们修改lsb那么颜色依然和没修改的一样，并且修改的话每个像数可以携带3比特的信息。



这个作用是在于把最低位的二进制全部提取出来



这个作用在于对提取出来的最低位使用lsb解码算法