BUUCTF Misc 二维码



<u>神音sss</u> ① 于 2021-04-28 22:43:52 发布 ① 353 ☆ 收藏 3

分类专栏: <u>CTF</u>

版权声明:本文为博主原创文章,遵循<u>CC 4.0 BY-SA</u>版权协议,转载请附上原文出处链接和本声明。

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

版权



CTF 专栏收录该内容

15 篇文章 0 订阅 订阅专栏



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

гоо гоо	<pre>root@oxygen:~# cd 123 root@oxygen:~/123# binwalk -e QR_code.png</pre>			
DEC	IMAL HEXADECIMAL	DESCRIPTION		
0 ed	ΘχΘ	PNG image, 280 x 280, 1-bit colormap, non-interlac		
471	0×1D7	Zip archive data, encrypted at least v2.0 to extra		
ct, 650	compressed size: 29, unco 0x28A	mpressed size: 15, name: 4number.txt End of Zip archive, footer length: 22 https://blog.csdn.net/oxygensss		

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





提取 +		flag.zip	٩	
< > 🖻	位置(L): 🗎 /			
名称	~	大小	类型	已修改
anumber.txt	8	15 字节	纯文本文档	2016年1月15日 10:04
 	取消(C) flag.zip"需要密码 图码(P): 			¢
			https://bl	og.csdn.net/oxygensss

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

<pre>root@oxygen:~/123# binwalk -D=png QR_code.png</pre>			
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, uncomp	ressed 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: Target File: MD5 Checksum: Signatures: ^{con} extra DECIMAL	2021-04-28 2 /root/123/0R c552424a0e73 de386 cted HEXADECIMAL	2:01:58 code.png a0614c41a4ae0cc6 C_code.png-0. extracted DESCRIPTION	3b7f t&R_code.png-1. extracted	OR_code.png
0 471 compressed s: 650	0x0 0x1D7 ize: 29, uncom 0x28A	PNG image, 28 Zip archive d pressed size: 15 End of Zip ar	0 x 280, 1-bit ata, encrypted , name: 4numbe chive, footer	c colormap, non-interlaced at least v2.0 to extract, er.txt length: 22
Scan Time: Target File: MD5 Checksum: Signatures:	2021-04-28 2 /root/123/_0 d41d8cd98f00 386	2:01:58 R_code.png-1.ext b204e9800998ecf8	racted/4number 427e	.txt
DECIMAL	HEXADECIMAL	DESCRIPTION		
root@oxvgen:~	/123#			https://blog.csdn.net/oxygensss

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I,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)-I 密码长度-u 压缩文件名 fcrackzip-b-c1-I4-u flag.zip

但备注一下stegslove的用法:



File	Analyse	Help	
	File Format		
	Data Extract		
	Stereogram Solver		
	Frame Browser		
	Image Combiner		

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

٢	– 🗆 X			
Extract Preview				
Bit Planes	Order settings			
Alpha 7 6 5 4 3 2 1 0	Extract By 🖲 Row 🔾 Column			
Red 7 6 5 4 3 2 1 0	Bit Order (
Green 7 6 5 4 3 2 1 0	Bit Plane Order			
Blue 7 6 5 4 3 2 1 0	● RGB ○ GRB			
	⊖ RBG ⊖ BRG			
Preview Settings	⊖ GBR ⊖ BGR			
Include Hex Dump In Preview 🖌				
Preview Save Text Save Bin Cancel plotteside net/oxydenses				

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(0255,0255,0~255)

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)

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

Bit Order 🔘 MSB First 💿 LSB First

Rit Plane Order

_

这个作用在于对提取出来的最低位使用lsb解码算法

<u>م</u>	- 🗆 ×
Extract Preview	
0017666c61677b50 6e675f4c73625f59flag{P ng_Lsb_	Y
30755f4b306e772l 7dffffffffffff 0u_K0nw! }	
ffffffffffffffffffffffffffffff	
ffffffffffffffffffffffffffffff	
fffffffffffffffffffffffffffffff	
ffffffffffffffff fffffffffffff	
ffffffffffffffff fffffffffffff	
ffffffffffffffffffffffffffffff	
ffffffffffffffffffffffffffffff	
Dit Disease	O-d
Bit Planes	Order settings
Alpha 7 6 5 4 3 2 1 0	Extract By Row Column
Red 7 6 5 4 3 2 1 0	Bit Order 🔾 MSB First 💿 LSB First
Green 7 6 5 4 3 2 1 0	Bit Plane Order
Blue ☐ 7 ☐ 6 ☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1 🗾 0	○ RGB ○ GRB
	○ RBG ○ BRG
Preview Settings	
Include Hex Dump In Preview	⊖ GBR (● BGR
Preview Save Text Save	Bin Cancel/blog.csdn.nat/oxygansss