

# CTF—RGB三原色

原创

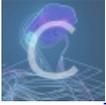
小常吃不了了 于 2021-07-27 17:34:20 发布 315 收藏

分类专栏: [CTF](#) 文章标签: [java unctf](#)

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/weixin\\_52620919/article/details/119148986](https://blog.csdn.net/weixin_52620919/article/details/119148986)

版权



[CTF 专栏收录该内容](#)

32 篇文章 1 订阅

订阅专栏

将题目附件现在下来打开, 发现很多的三个一组的数字排列, 排列之后可以看到一共有61366组。

```
61344 255,255,255
61345 255,255,255
61346 255,255,255
61347 255,255,255
61348 255,255,255
61349 255,255,255
61350 255,255,255
61351 255,255,255
61352 255,255,255
61353 255,255,255
61354 255,255,255
61355 255,255,255
61356 255,255,255
61357 255,255,255
61358 255,255,255
61359 255,255,255
61360 255,255,255
61361 255,255,255
61362 255,255,255
61363 255,255,255
61364 255,255,255
61365 255,255,255
61366 255,255,255
```

其实不难发现这应该是RGB三原色, 我们就找个白色画布, 把文件读进来。

(1) 文档一共61366所以我们用在线工具进行质因数分解

# 质因数在线计算器

发布时间: 2013-12-24 21:44:43 作者: 伊人

!本文不许可它人转载,只许可引用链接.



因为我们要的图片是二维的, 所以只需要两个质数【 $503 \times 122$ 】

- (2) 打开电脑的画图功能生成一张空白的, 大小为【 $503 \times 122$ 】的图片并保存在桌面
- (3) 用java脚本将【数字文档】的数据写入【图片】

相关代码:

```
public class test {  
  
    public static void main(String[] args) throws IOException {  
  
        BufferedReader br = new BufferedReader(new InputStreamReader(  
  
            new FileInputStream("D:\\misc100.txt")));  
  
        int i,j;  
  
        String line = br.readLine();  
  
        int rgb[] = new int[3];  
  
        File file = new File("D:\\2.jpg");  
  
        BufferedImage bi = null;  
  
        bi = ImageIO.read(file);  
  
        int width = bi.getWidth();  
  
        int height = bi.getHeight();  
  
        int minx = bi.getMinX();  
  
        int miny = bi.getMinY();
```

```

int mIny = bi.getHeight();

for (i = 0; i < width; i++){

    for (j = 0; j < height; j++){

        if(line == null) break;

        String[] rgbs=line.toString().split(",");

        rgb[0]=new Integer(rgbs[0]);

        rgb[1]=new Integer(rgbs[1]);

        rgb[2]=new Integer(rgbs[2]);

        bi.setRGB(i,j,Integer.parseInt(Integer.toHexString(rgb[0])+Integer.toHexString(rgb[1])+Integer.toHexString(rgb[2]),16));

        //bi.setRGB(i,j,Integer.parseInt("ffffff",16))

        line = br.readLine();

    }

}

ImageIO.write(bi, "JPEG", file);

br.close();

}

}
```

```

在这里插入代码片

注意代码中的文件位置需要根据实际情况进行更改

运行代码之后可以在你设定的路径下得到一张图片。

打开就可以看到 flag



[https://blog.csdn.net/weixin\\_52620919](https://blog.csdn.net/weixin_52620919)

如果图片看不清的话就调整一下像素，256\*245，再运行一下

flag( )

flag( youc@n'tseem

youc@n'tseeme )

me )