

CTF之图片隐写术解题思路

转载

[bamanju0574](#) 于 2019-09-09 20:02:00 发布 5958 收藏 27

文章标签: [java 数据库](#)

原文链接: <http://www.cnblogs.com/-chenxs/p/11493898.html>

版权

参考大佬的博客: https://blog.csdn.net/a_small_rabbit/article/details/79644078

深有感触,觉得写得比较全,因此将此篇文章记录在此方便以后用得到。

0x01图像隐写术进行数据隐写分为以下几类:

- 1.在图片右击查看属性,在详细信息中隐藏数据
- 2.将数据类型进行改写(rar或者zip数据改为jpg等格式)
- 3.根据各种类型图像的固定格式,隐藏数据
- 在编译器中修改图像开始的标志,改变其原来图像格式
- 在图像结束标志后加入数据
- 在图像数据中加入数据,不影响视觉效果情况下修改像素数据,加入信息
- 4.利用隐写算法将数据隐写到图片中而不影响图像(仅限于jpg图像) 隐写常用的算法有F5, guess jsteg jphide。

0x02破解隐写术方法及步骤

- 1.查看图像属性详细信息是否有隐藏内容
- 2.利用winhex或nodepad++打开搜索ctf,CTF, flag,key等关键字是否存在相关信息
- 3.检查图像的开头标志和结束标志是否正确,若不正确修改图像标志恢复图像,打开查看是否有flag或ctf信息,(往往gif属于动图,需要分帧查看各帧图像组合所得数据 若不是直接的ctf或flag信息 需要考虑将其解码)
- jpg图像开始标志: FF D8 结束标志: FF D9
- gif图像开始标志: 47 49 46 38 39 61 (GIF89)结束标志: 01 01 00 3B
- bmp图片开始标志: 42 4D //92 5B 54 00 00 00 00 00 结束标志: 00
- png图片开始标志: 89 50 结束标志: 60 82
- 4.将图片放置在kail系统中,执行binwalk xxx.jpg 查看图片中是否是多个图像组合或者包含其他文件(若存在多幅图像组合,再执行foremost xxx.jpg会自动分离;若检测出其他文件修改其后缀名即可,如zip)
- 5.使用StegSolve对图像进行分通道扫描,查看是否为LSB隐写
- 6.在kail下切换到F5-steganography,在java Extract运行
- 命令: java Extract 123456.jpg图片的绝对地址 -p 123456
- 判断是否为F5算法隐写
- 7.在kali系统中使用outguess-master工具(需要安装),检测是否为guess算法隐写

0x03算法隐写的具体操作

1.F5算法隐写

具体操作:在kail下切换到F5-steganography,在java Extract运行

命令: java Extract 123456.jpg图片的绝对地址 -p 123456

2.LSB算法隐写

具体操作:在Stegsolve.jar分析data Extract的red blue green

3.guess算法隐写

具体操作：在kail下切换到outguess目录下，直接用命令即可

命令:outguess -r /root/angrybird.jpg(绝对路径) 123.txt(信息存放的文本)

0x04工具使用

1.MP3stego

```
encode -E hidden_text.txt -P pass svega.wavsvega_stego.mp3
```

```
Decode.exe -X -P pass(密码) svega_stego.mp3(要拷贝到目录下) //解码
```

2.stegdetect

Stegdetect可以检测到通过JSteg、JPHide、OutGuess、Invisible Secrets、F5、appendX和Camouflage等这些隐写工具隐藏的信息

s – 修改检测算法的敏感度，该值的默认值为1。检测结果的匹配度与检测算法的敏感度成正比，算法敏感度的值越大，检测出的可疑文件包含敏感信息的可能性越大。

d – 打印带行号的调试信息。

t – 设置要检测哪些隐写工具（默认检测jopi），可设置的选项如下：

j – 检测图像中的信息是否是用jsteg嵌入的。

o – 检测图像中的信息是否是用outguess嵌入的。

p – 检测图像中的信息是否是用jphide嵌入的。

i – 检测图像中的信息是否是用invisible secrets嵌入的。

命令：stegdetect.exe -tjopi -s10.0 xxx.jpg

转载于:<https://www.cnblogs.com/-chenxs/p/11493898.html>