MISC总结——隐写术(一）

转载

xuchen16 于 2018-10-08 16:21:15 发布 8686 收藏 29
分类专栏： ctf 文章标签： MISC 隐写术 附加式的图片隐写


ctf 专栏收录该内容
66 篇文章 6 订阅
订阅专栏

转载自：https://www.cnblogs.com/lxz-1263030049/p/9388511.html

一直有这个想法，打算把关于misc类型的题目总结一下，希望能够提醒自己一直学习

也希望能够帮助到那些需要帮助的人

本文参考自：先知社区：https://xz.aliyun.com/t/1833

隐写术介绍：

隐写术是关于信息隐藏，即不让计划的接收者之外的任何人知道信息的传递事件（而不只是信息的内容）的一门技巧与科学。

英文写作Steganography，而这篇内容将带大家了解一下CTF赛场上常见的图片隐写方式，以及解决方法。有必要强调的是，隐写术与密码编码是完全不同的概念。

第一部分：附加式的图片隐写

在附加式的图片隐写术中，我们通常是用某种程序或者某种方法在载体文件中直接附加上需要被隐写的目标，

然后将载体文件直接传输给接受者或者发布到网站上，然后接受者者根据方法提取出被隐写的消息，这一个过程就是我们这里想提到的附加式图片隐写。

而在CTF赛事中，关于这种图片隐写的大概有两种经典方式，一是直接附加字符串，二是图种的形式出现

实验部分：

找到隐写术目录，打开图片隐写，打开图片隐写第一部分文件夹
在该文件夹找到 xscq.jpg，
双击打开图片，我们先确认一下图片内容并没有什么异常
正如前文所说，我们这个实验部分讲的是附加字符串的隐写方式，所以我们用Strings检查一下图片
在Strings工具的搜索下，我们看到了一串base64编码后的字符串
最终解码后，flag：flag{welcome_to_xianzhi}

strings使用方法

strings命令在对象文件或二进制文件中查找可打印的字符串。字符串是4个或更多可打印字符的任意序列，以换行符或空字符结束。 strings命令对识别随机对象文件很有用。

选项：

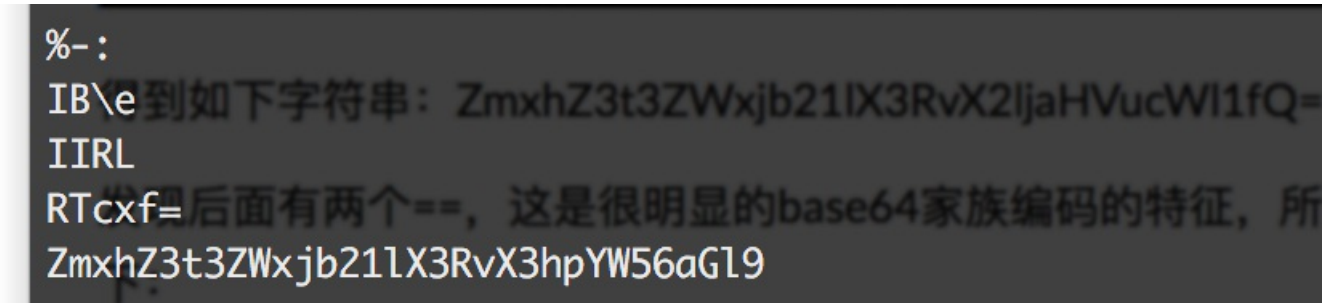
- -a --all：扫描整个文件而不是只扫描目标文件初始化和装载段
- -f –print-file-name：在显示字符串前先显示文件名

- -t --radix={o,d,x} ：输出字符的位置，基于八进制，十进制或者十六进制
- -e --encoding={s,S,b,l,B,L} ：选择字符大小和排列顺序:s = 7-bit, S = 8-bit, {b,l} = 16-bit, {B,L} = 32-bit

Tips 我们使用strings + 文件名字的命令即可
具体步骤如下：

在cmd中打开strings工具，使用如下命令

```
strings ctf.jpg
```



得到如下字符串：ZmxhZ3t3ZWxjb21lX3RvX3hpYW56aGl9
我们尝试用base64解码，代码过程如下：

```
Python 2.7.12 (v2.7.12:d33e0cf91556, Jun 27 2016, 15:24:40) [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import base64
>>> base64.b64decode('ZmxhZ3t3ZWxjb21lX3RvX3hpYW56aGl9')
'flag{welcome_to_xianzhi}'
>>>
```

这里也可以使用在线解密工具

有必要提到的是，为什么字符串要附加在文件的后面呢?那是因为，如果图片附加在中间，有可能破坏了图片的信息，

如果字符串附加在图片的头部位置，又破坏了文件头，可能导致图片无法识别。关于文件格式的具体内容，我们下一个部分的隐写还会提到。

第二部分：图种形式隐写

图种：
一种采用特殊方式将图片文件（如jpg格式）与rar文件结合起来的文件。该文件一般保存为jpg格式，可以正常显示图片，

当有人获取该图片后，可以修改文件的后缀名，将图片改为rar压缩文件，并得到其中的数据。
图种这是一种以图片文件为载体，通常为jpg格式的图片，然后将zip等压缩包文件附加在图片文件后面。

因为操作系统识别的过程中是，从文件头标志，到文件的结束标志位，当系统识别到图片的结束标志位后，默认是不再继续识别的，所以我们在通常情况下只能看到它是只是一张图片。

实验部分：

```
在实验机中找到隐写术目录，打开图片隐写，打开图片隐写第一部分文件夹
在该文件夹找到cqzb.jpg，
双击打开图片，我们先确认一下图片内容并没有什么异常
对图片进行检测，确认是不是图种
使用winhex打开图片，并分离图片，得到一个压缩包
打开压缩包得到flag，flag：flag{This is easy}

在linux中是binwalkk命令进行分离
命令如下：
```

```
binwalk cqzb.jpg
```

我们可以发现，binwalk自动识别出来了zip文件，而且偏移也告诉我们了,当然我们这里如果使用

```
binwalk cqzb.jpg -e
```

这样的命令，是很快就能把ZIP文件给提取出来的，但是这里我想讲的是如何用winhex等16进制编辑器，将压缩包提取出来。

使用winhex16进制编辑器提取ZIP文件

- 首先需要了解一下什么是文件头
  文件头就是是位于文件开头的一段承担一定任务的数据。一般都在开头的部分。以jpg图片和zip压缩包文件为例。
- 图6和图7分别是jpg图片的文件头以及jpg图片的结尾。
  我们如何，找到JPG图片和ZIP图片呢？
  JPG图片的文件头和结束标志

上图，FF D8 FF E1就是JPG图片的文件头，一般当我们看到文件开头是如此的格式，我们就能认为这是一个JPG图片了。

上图以 03 FF D9为结束标志，这是JPG图片的结束标志位。

ZIP文件的文件头和结束标志

上图 50 4B 03 04就是ZIP文件的文件头，一般以PK表示。

- 找到cqzb.jpg 中隐藏的ZIP文件

上文我们讲述了，JPG图片的结束标识是03 FF D9,ZIP文件的文件头是50 4B 03 04，我们只需要在winhex中找到ZIP文件的文件头即可，

滑动滚条到最底下。上文讲了一般附加的位置是在原本文件的后面，所以我们果断滑动滚动条到最后。

从图中我们可以明显看到cqzb.jpg明显不是以FF D9结尾，而且我们在上面不远的地方发现了zip的文件头50 4B 03 04，所以我们可以断定这是个图种文件了

- 分离ZIP文件
  下一步我们该如何用winhex截取我们所需要的文件呢？
  我们选取以50开头以及到末尾的的数据，右键单击，选择编辑，复制选块到新文件，保存新文件为zip格式命名规则即可。

  保存为ZIP文件，解压缩后就能得到flag，所以最后的flag是flag{This is easy}

这里我说的是使用比较传统的分离方法

还有一些简单的操作（由于时间比较晚了，我直接写命令了）

在linux中使用

foremost进行分离

具体命令如下：

```
foremost 1.jpg
```

这样就可以了

也可以使用binwalk

具体步骤想一下哦！！！！（百度也是可以的）

题目.zip(0.042 MB) 下载附件