2017年陕西省网络空间安全技术大赛——种棵树吧——Writeup

转载

baikeng3674 于 2017-04-18 09:59:00 发布 81 收藏
文章标签： 数据结构与算法
原文链接：http://www.cnblogs.com/WangAoBo/p/6726322.html

2017年陕西省网络空间安全技术大赛——种棵树吧——Writeup

- 下载下来的zip解压得到两个jpg图片，在Kali中使用binwalk查看文件类型如下图：

有两个发现：

> - 1111.jpg 隐藏了一个压缩文件，可解压得到另一个文件1.gif
> - 2222.jpg 中隐藏了一段奇怪的字符 **Post-order{YR!eVa-gLAoxd_j{pw}8zkUnGulHh:r65f2lFsEi*}** (刚开始以为这个就是flag= =)

- 从1111.jpg中分离出1.gif， 直接把jpg后缀改为zip，解压即可
- 分离出了1.gif， 惊喜地发现1.gif是一片空白的

- 再次用binwalk查看1.gif的文件类型，发现1.gif是没有文件头的，如下图：

- 于是就想到了用16进制修改工具（winhex， 01editor）等等加上gif的文件头 **47 49 46 38**, 并保存图片

> 常见文件的文件头 http://www.cnblogs.com/WangAoBo/p/6366211.html

- 保存后的图片已经可以正常显示了，是一个3帧的动图，每一帧如下（gif查看器，ps，stegsolve均可分离gif的每一帧）

可以看出这个gif描述了一个字符串 **In-order{RY!heHVal-goAl{dxj_GpnUw8}kzu*Er:s56fFl2i}**

结合之前从2222.jpg得到的字符串 **Post-order{YR!eVa-gLAoxd_j{pw}8zkUnGulHh:r65f2lFsEi*}**

中的**In-order**和**Post-order**推断这与树的遍历有关

即：

- 中序遍历序列：RY!heHVal-goAl{dxj_GpnUw8}kzu*Er:s56fFl2i
- 后序遍历序列：YR!eVa-gLAoxd_j{pw}8zkUnGulHh:r65f2lFsEi*

由上述分析可以还原出树的结构，还原之后就可以很容易的看出flag了（建议用一张大的纸= =）

flag为**flag{n52V-jpU6d_kx8zw}**

补充：

分离1111.jpg的方法还有 **dd if=1111.jpg of=1111-1.zip skip=125330 bs=1**， 此方法来自阿良

找到2222.jpg中字符串的方法还有：

1. **strings 2222.jpg**列出可打印字符串，此方法来自珏哥
2. 直接用记事本打开2222.jpg
3. 官方WP的直接右键查看属性

转载于:https://www.cnblogs.com/WangAoBo/p/6726322.html