

深入理解 JPEG 图像格式 Jphide 隐写

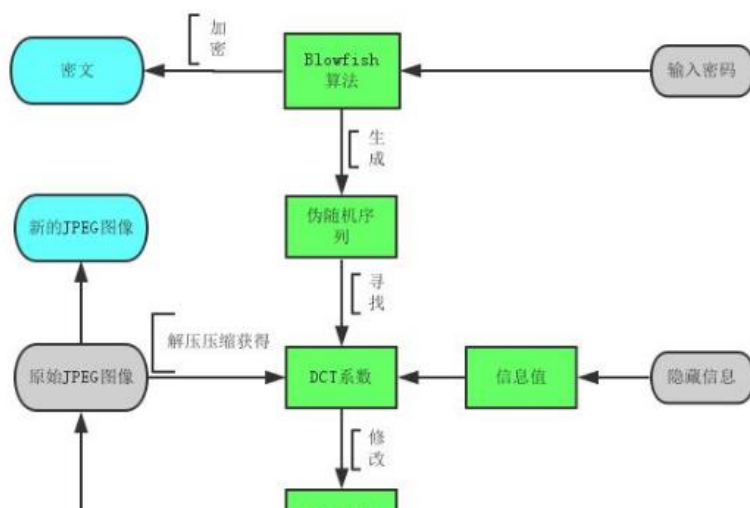
1、隐写原理

Jphide 是基于最低有效位 LSB 的 JPEG 格式图像隐写算法，使用 JPEG 图像作为载体是因为相比其他图像格式更不容易发现隐藏信息，因为 JPEG 图像在 DCT 变换域上进行隐藏比空间域隐藏更难检测，并且鲁棒性更强，同时 Blowfish 算法有较强的抗统计检测能力。

由于 JPEG 图像格式使用离散余弦变换(Discrete Cosine Transform ,DCT) 函数来压缩图像，而这个图像压缩方法的核心是：通过识别每个 8×8 像素块中相邻像素中的重复像素来减少显示图像所需的位数，并使用近似估算法降低其冗余度。因此，我们可以把 DCT 看作一个用于执行压缩的近似计算方法。因为丢失了部分数据，所以 DCT 是一种有损压缩(Loss Compression) 技术，但一般不会影响图像的视觉效果。

2、隐写过程

Jphide 隐写过程大致为：先解压压缩 JPEG 图像，得到 DCT 系数；然后对隐藏信息用户给定的密码进行 Blowfish 加密；再利用 Blowfish 算法生成伪随机序列，并据此找到需要改变的 DCT 系数，将其末位变为需要隐藏的信息的值。最后把 DCT 系数重新压回成 JPEG 图片，下面是个人对隐写过程理解画出的大致流程图。



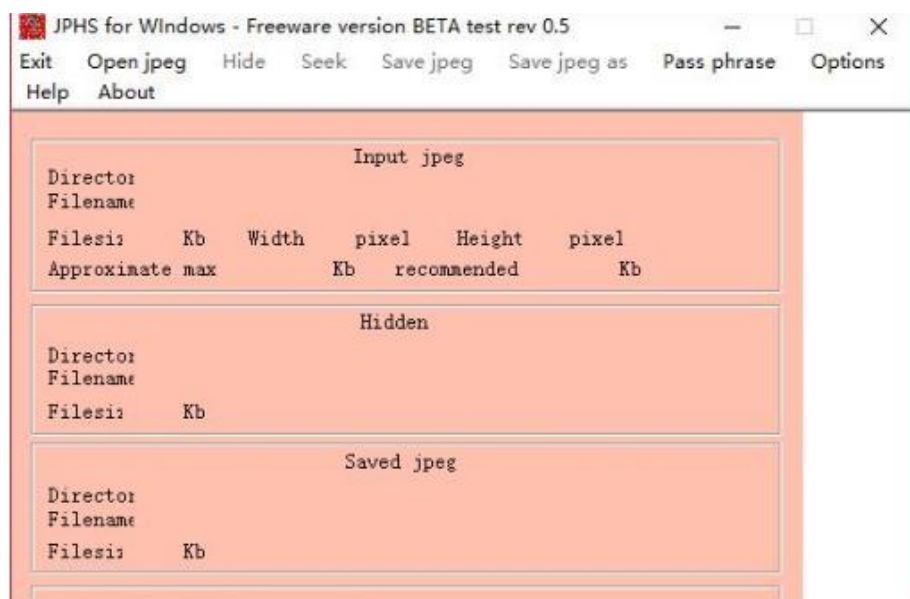
3、隐写实现

3.1、Stegdetect

实现 JPEG 图像 Jphide 隐写算法工具有多个，比如由 Neils Provos 开发通过统计分析技术评估 JPEG 文件的 DCT 频率系数的隐写工具 Stegdetect，它可以检测到通过 JSteg、JPHide、OutGuess、Invisible Secrets、F5、appendX 和 Camouflage 等这些隐写工具隐藏的信息，并且还具有基于字典暴力破解密码方法提取通过 Jphide、outguess 和 jsteg-shell 方式嵌入的隐藏信息。

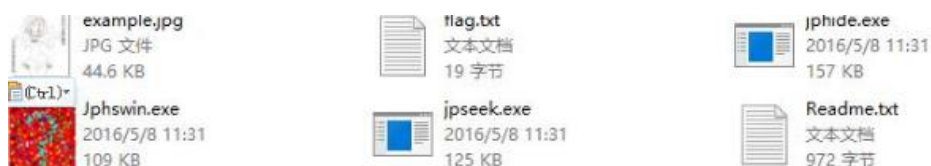
3.2JPHS

而这里介绍另一款 JPEG 图像的信息隐藏软件 JPHS，它是由 Allan Latham 开发设计实现在 Windows 和 Linux 系统平台针对有损压缩 JPEG 文件进行信息加密隐藏和探测提取的工具。软件里面主要包含了两个程序 JPHIDE 和 JPSEEK，JPHIDE 程序主要是实现将信息文件加密隐藏到 JPEG 图像功能，而 JPSEEK 程序主要实现从用 JPHIDE 程序加密隐藏得到的 JPEG 图像探测提取信息文件，Windows 版本的 JPHS 里的 JPHSWIN 程序具有图形化操作界面且具备 JPHIDE 和 JPSEEK 的功能。



1.Windows 用户请下载 JPHS-05 for Windows,同时也提供下载 Linux 版本。

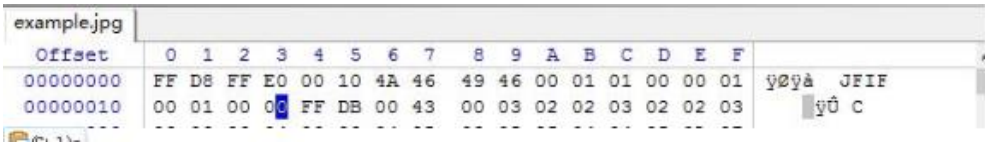
2.分别准备一个 JPEG 格式的图片(example.jpg)和一个文本文件(flag.txt)。



由于 JPEG 文件使用的数据存储服务有多种不能一一演示，这里用最常用的 JPEG 格式-JPEG 文件交换格式 (JPEG File Interchange Format , JFIF) 作为示例。

这里简单介绍 JPEG 文件交换格式的 JPEG 图片的图像开始标记 SOI(Start of Image) 和应用程序保留标记 APP0 (Application 0) , JPEG 文件交换格式的 JPEG 图片开始前 2 个字节是图像开始标记为 0xFFD8 , 之后 2 个字节接着便是应用程序保留标记为 0xFFE0 , 应用程序保留标记 APP0 包含 9 个具体字段 , 这里介绍前三个字段 , 第一个字段是数据长度占 2 个字节 , 表示包括本字段但不包括标记代码的总长度 , 这里为 10 个字节 , 第二个字段是标识符占 5 个字节 0x4A46494600 表示 “JFIF0” 字符串 , 第三个字段是版本号占 2 个字节 , 这里

是 0x0101，表示 JFIF 的版本号为 1.1，但也可能为其它数值，从而代表了其它版本号。



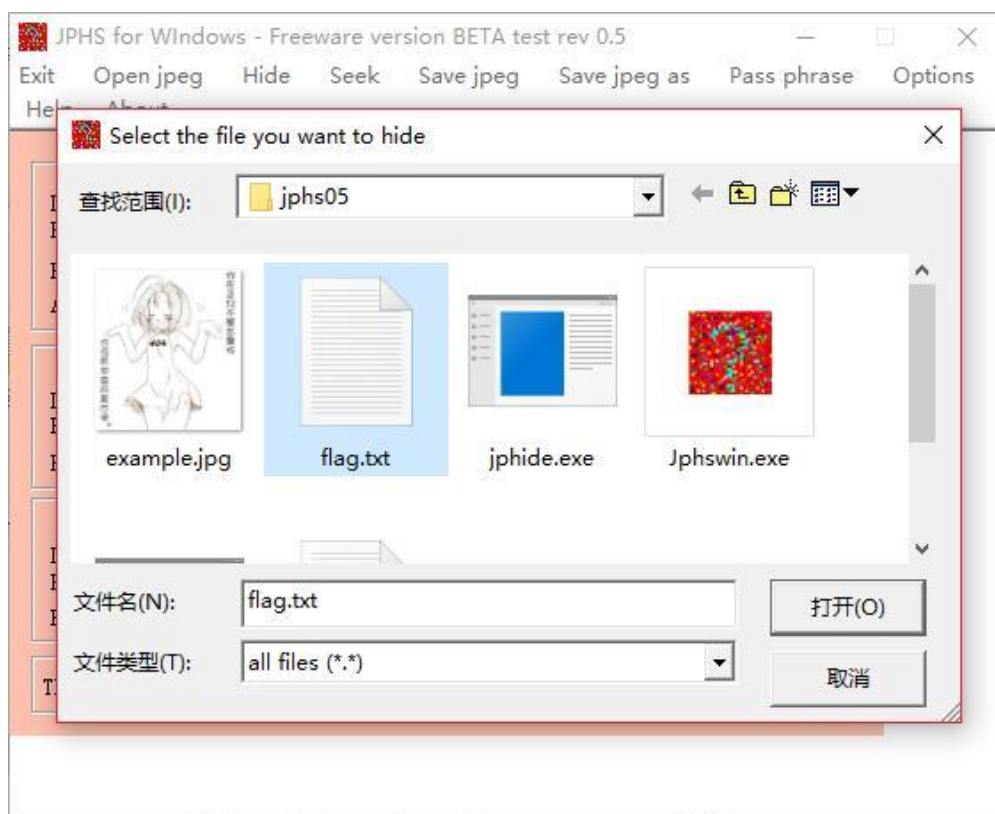
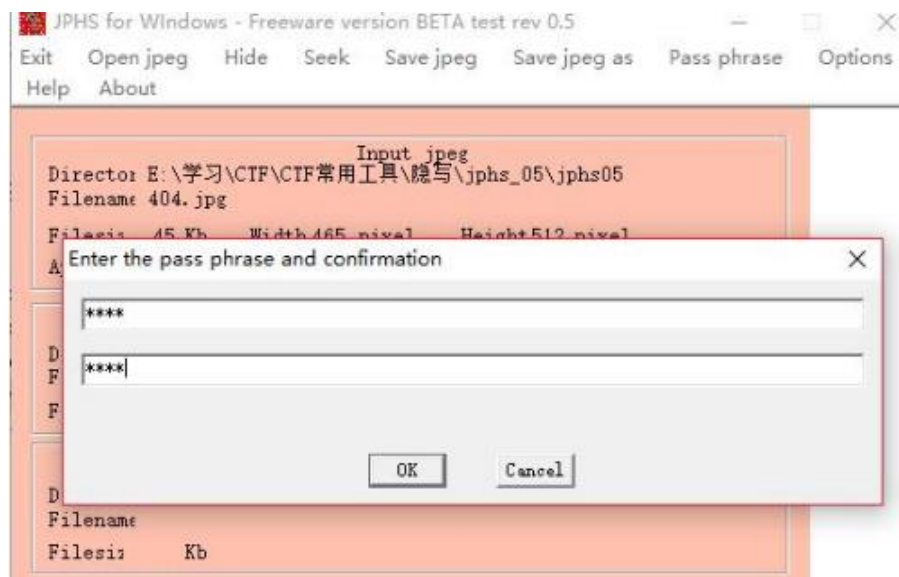
example.jpg	Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
	00000000	FF	D8	FF	E0	00	10	4A	46	49	46	00	01	01	00	00	01	ÿøÿà JFIF
	00000010	00	01	00	00	FF	DB	00	43	00	03	02	02	03	02	02	03	ÿÜ C

3.Windows 版本可以使用具有图形化操作界面的 Jphswin，选择 “Open jpeg” 打开示例 JPEG 格式图片 example.jpg

如果你选择的不是 JPEG 格式的图片程序会自动退出，你可以 16 进制编辑器如 Winhex 查看图片的图像开始标记 SOI 和应用程序保留标记 APP0，当载入 JPEG 格式图片会显示一些图片的属性。



4.选择“Hide”选项之后在两次文本框输入相同的密码，这里以输入 flag 作为密码为例，然后输入要包含隐藏信息的文本。



5.选择 “Save jpeg as” 选项将图片另存为 jpeg 格式并输入文件的名称为新的图像文件如 C4n-u-find-f14g.jpg。



6.之后便可以看到生成结果和相关信息。



7.第 2 步到第 7 步做的是 Jhide 方式信息隐藏 ,接下来我们从 C4n-u-find-f14g.jpg 图片提取出隐藏信息。



8.如果之前你并不知道图片是基于什么方式进行信息隐藏，你可以使用 Stegdetect 先进行探测。

Stegdetect 的主要选项如下：

- q 仅显示可能包含隐藏内容的图像。
- n 启用检查 JPEG 文件头功能，以降低误报率。如果启用，所有带有批注区域的文件将被视为没有被嵌入信息。如果 JPEG 文件的 JFIF 标识符中的版本号不是 1.1，则禁用 OutGuess 检测。
- 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。
- V 显示软件版本号。

如果检测结果显示该文件可能包含隐藏信息，那么 Stegdetect 会在检测结果后面使用 1 ~ 3 颗星来标识 隐藏信息存在的可能性大小，3 颗星表示隐藏信息存在的可能性最大。

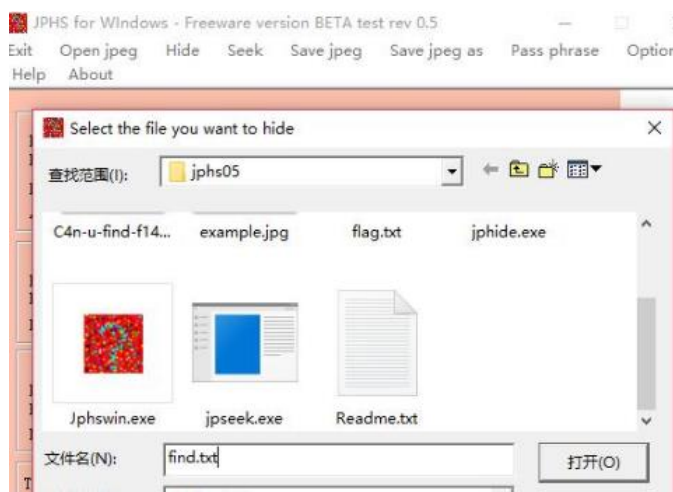
从下图可以看出很可能是 Jphide 的信息隐藏方式：



```

C:\WINDOWS\system32\cmd.exe
E:\学习\CTF\CTF常用工具\隐写\stegdetect-0.4>stegdetect.exe -tjopi -s 10.0 C4n-u-find-f14g.jpg
C4n-u-find-f14g.jpg : jphide(***)
```

9.在知道隐藏方式之后可以开始进行信息提取，和使用 JPHS 进行信息隐藏过程类似，打开需要提取隐藏信息的图片 C4n-u-find-f14g.jpg，输入对应密码（在不知道密码的情况不可以尝试 Stegdetect 工具里的 Stegbreak 程序进行基于字典的暴力攻击）flag，密码验证通过 JPHS 会自动提取隐藏信息，之后便可以另存提取出的信息。



10.打开提取得到的 find.txt 便可以得到我们想要的隐藏信息。

