你需要了解下各种图片的基本结构框架

隐写：

binwalk 7z.exe show file in this binary file

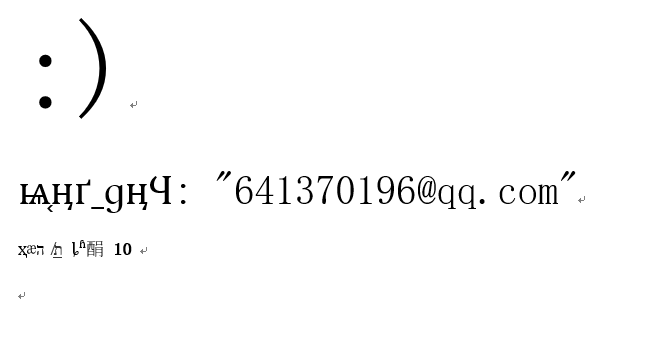
binwalk -c 7z.exe get inner file in this

用StegSolve(基于Java开发的流行图片隐写分析软件)打开图片，浏览图片各个颜色通道，发现Blue plane 0通道时图片左上角突然多了一些内容，应该就是隐藏的二进制信息。所以使用StegSolve中Analyse选项下Data Extract功能查看Blue plane 0通道。如果图片的灰色通道有特殊信息，或者图片就是灰度图像，那么你用py得到01序列，然后视为二进制（有编码转换器或者放到WinHex等）没准就可以得到你想要的隐写内容。

音频：

打开之后播放几次没有听出什么异常，然后分析一波，既然是基于提示是语音认证，那么解题方向应该不是数字隐藏（应该是指在里面藏文件或者其他内容），然后试着改变音频轨道和效果，分析音频频谱，也没有发现什么，之后观察音频波形，发现波形很有规律，总共就两种波形，刚好对应二进制01。

txt改成word： 里面的笔记变成这个



raspistill -w 800 -h 600 -tl 0 -t 0 -o - | tee -a %d.jpeg > /dev/null

通过tee不断向原图片中追加，却还能按照正常的图片进行显示。

winhex使用指南：

搜索有好几种，不要使用C-F快捷键，自己点工具栏中的Search，按F3搜索下一个。

PE文件的最后一个段后面是可以添加附加数据（称之为overlay），实际应用：软件要处理一些数据流文件，这些数据文件被单独的保存在硬盘上，当我们使用notepad的打开功能的时候，就可以去读取数据文件了。如果将这些需要读取的数据放到pe文件的后面，让程序自动的运行打开的功能，而不需要打开操作。如果数据不在PE的区段中，代码执行时就不会从磁盘被映射到内存，就是说如果在文件中存在不在区段里面的数据，这部分数据就被认为是附加数据(overlay). overlay是不映射到内存的，他将被程序以打开自己的方式来读取数据。通过PEID查壳可以查出来是否在末尾包含overlay。