音频隐写

与音频相关的 CTF 题目主要使用了隐写的策略，主要分为 MP3 隐写，LSB 隐写，波形隐写，频谱隐写等等。

常见手段

通过 binwalk 以及 strings 可以发现的信息不再详述。

MP3 隐写

原理

MP3 隐写主要是使用 Mp3Stego 工具进行隐写，其基本介绍及使用方法如下

> MP3Stego will hide information in MP3 files during the compression process. The data is first compressed, encrypted and then hidden in the MP3 bit stream.

```Python
encode -E hidden_text.txt -P pass svega.wav svega_stego.mp3
decode -X -P pass svega_stego.mp3
```

例题

> ISCC-2016: Music Never Sleep

初步观察后，由 strings 无发现，听音频无异常猜测使用隐写软件隐藏数据。

得到密码后使用 Mp3Stego 解密。

```Apache
decode.exe -X ISCC2016.mp3 -P bfsiscc2016
```

得到文件 iscc2016.mp3.txt:

```C#
Flag is SkYzWEk0M1JOWlNHWTJTRktKUkdJTVpXRzVSV0U2REdHTVpHT1pZPQ== ???
```

Base64 && Base32 后得到 flag。

波形

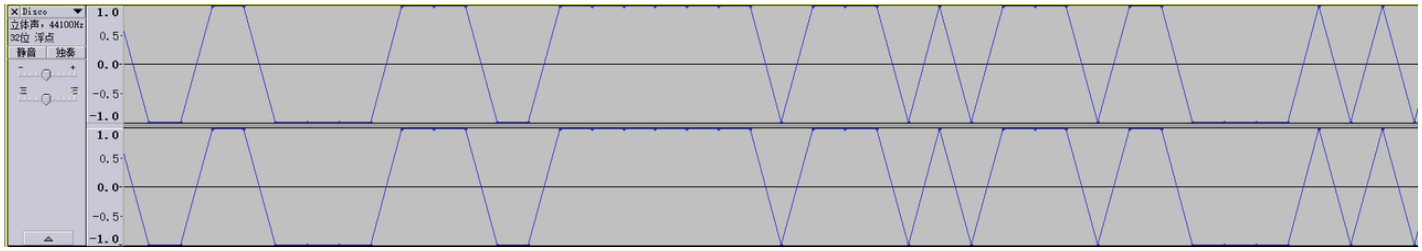
原理

通常来说，波形方向的题，在观察到异常后，使用相关软件（Audacity, Adobe Audition 等）观察波形规律，将波形进一步转化为 01 字符串等，从而提取转化出最终的 flag。

例题

> ISCC-2017: Misc-04

其实这题隐藏的信息在最开始的一段音频内，不细心听可能会误认为是隐写软件。



以高为 1 低为 0，转换得到 01 字符串。

```Python
110011011011001100001110011111110111010111011000010101110101010110011011101011101110110111011100111111101
```

转为 ASCII，摩斯密码解密，得到 flag。

Note

一些较复杂的可能会先对音频进行一系列的处理，如滤波等。例如 JarvisOJ - 上帝之音 Writeup

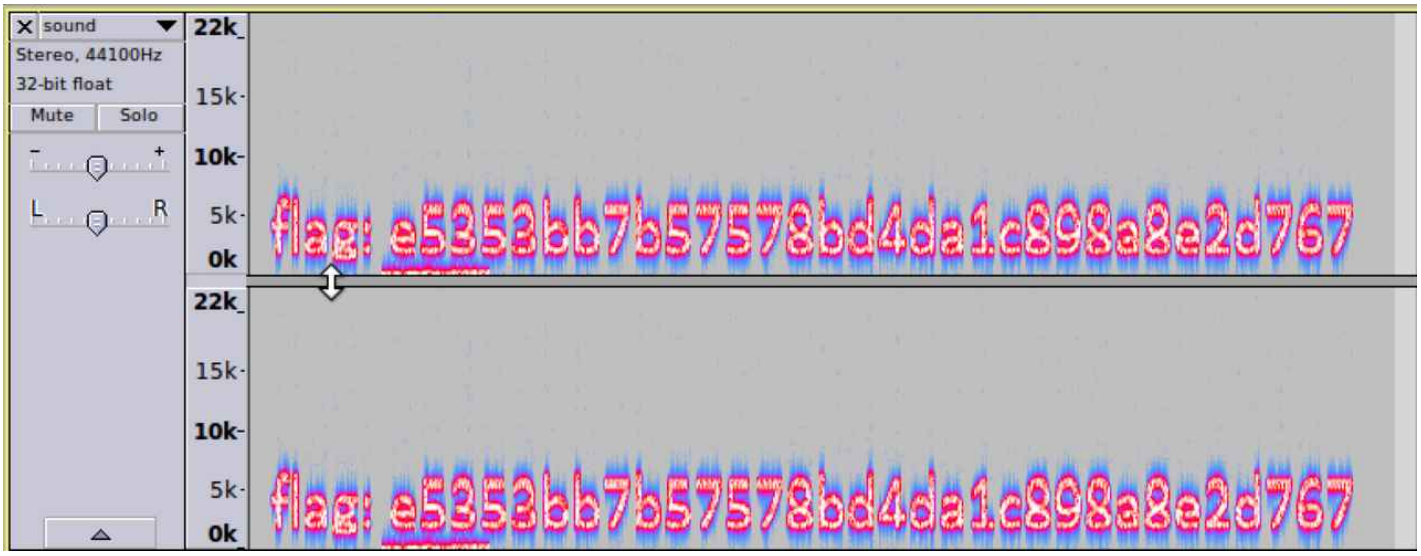
频谱

原理

音频中的频谱隐写是将字符串隐藏在频谱中，此类音频通常会有一个较明显的特征，听起来是一段杂音或者比较刺耳。

例题

> Su-ctf-quals-2014:hear_with_your_eyes



LSB 音频隐写

原理

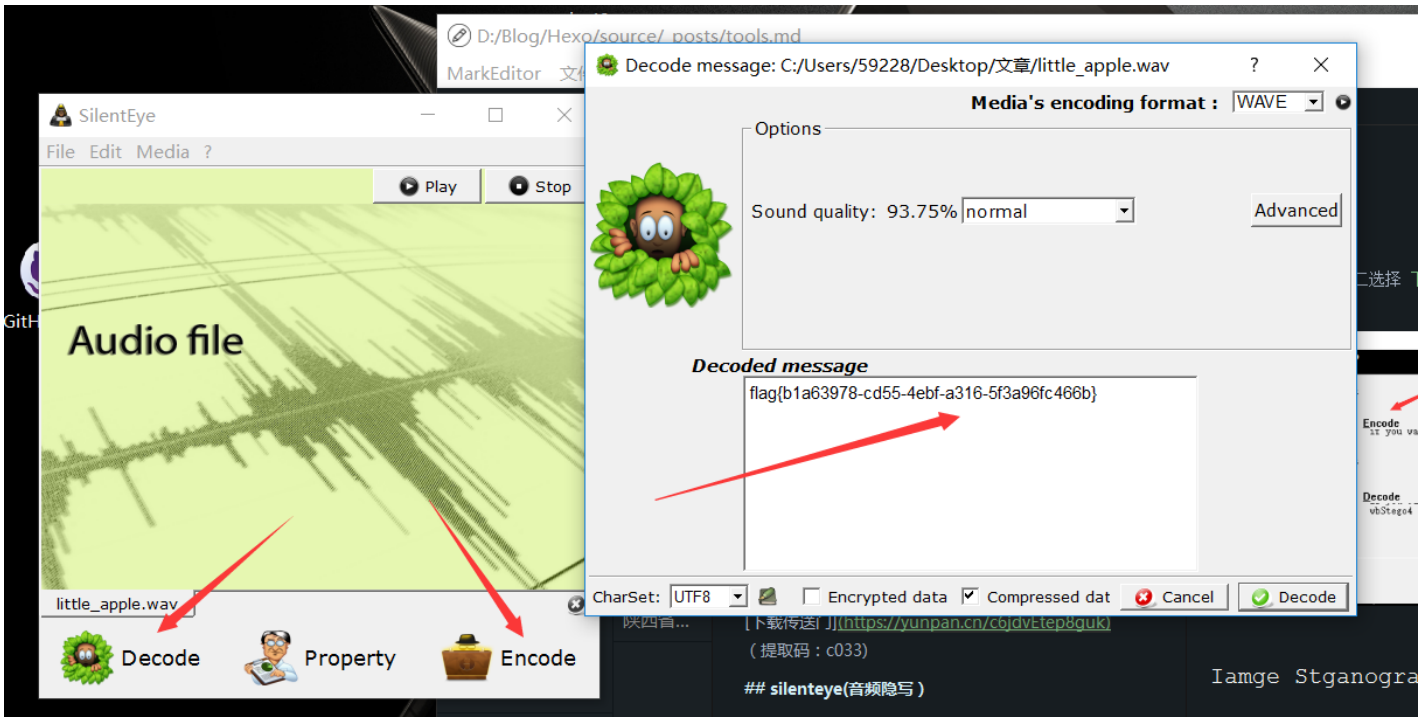
类似于图片隐写中的 LSB 隐写，音频中也有对应的 LSB 隐写。主要可以使用 Silenteye 工具，其介绍如下：

> SilentEye is a cross-platform application design for an easy use of steganography, in this case hiding messages into pictures or sounds. It provides a pretty nice interface and an easy integration of new steganography algorithm and cryptography process by using a plug-ins system.

例题

> 2015 广东省强网杯 - Little Apple

直接使用 silenteye 即可。



延伸

- 音频中的 LSB
- 隐写术总结