**2.6** <https://play.picoctf.org/practice/challenge/117?category=4&page=3>

### **Surfing the Waves**

#### **Description**

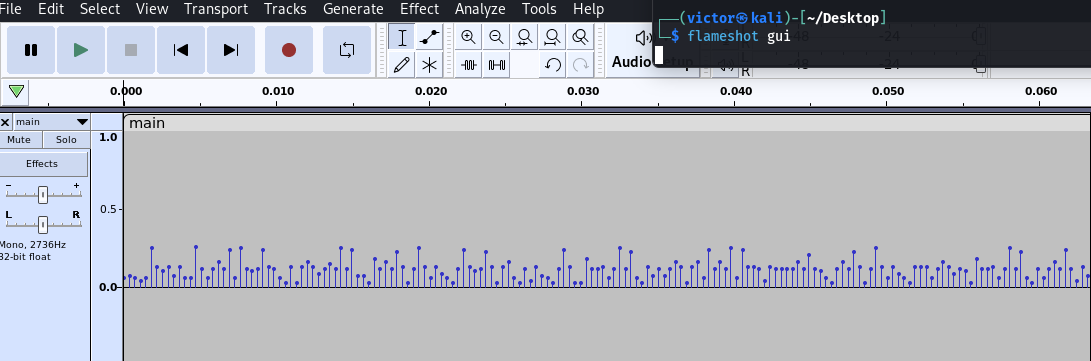
While you're going through the FBI's servers, you stumble across their incredible taste in music. One [main.wav](https://mercury.picoctf.net/static/b13503a965f7ea4d66fdec4b7f88ec9e/main.wav) you found is particularly interesting, see if you can find the flag!

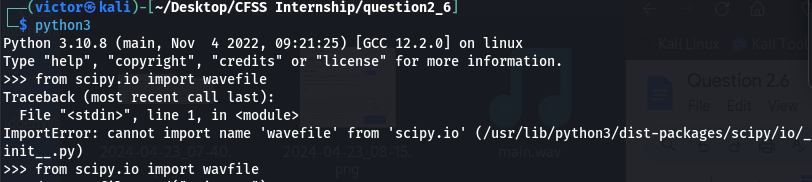
#### **Hints**

Music is cool, but what other kinds of waves are there?

Look deep below the surface.

Solution

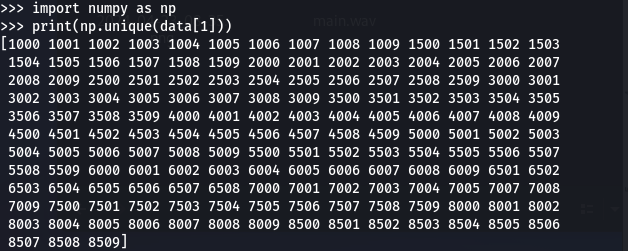
I tried exploring the file using Audacity to see any peculiarities. After zooming in, I found that the file contained discrete sets of numbers above zero so I tried using Python instead.  
  
  
I started by opening Python and importing the necessary libraries. Specifically, I imported the scipy.io.wavfile module to handle the WAV file.



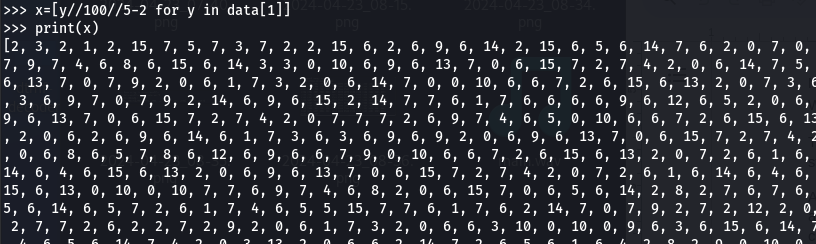
Next, I read the WAV file into memory using the wavfile.read() function. The data from the WAV file was stored in a variable named data.



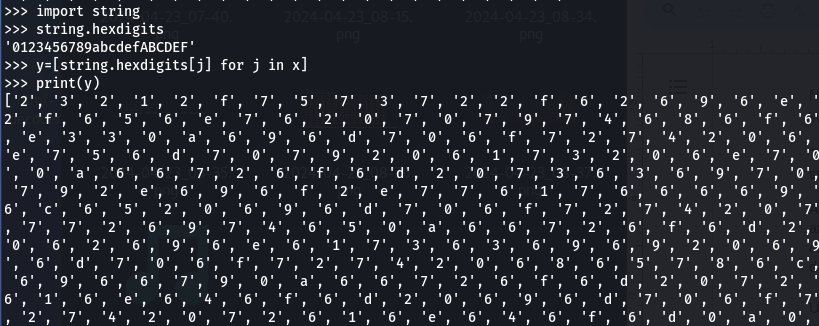
To analyze the audio data, I utilized NumPy to find the number of unique samples present. This step provided insights into the characteristics of the audio signal.



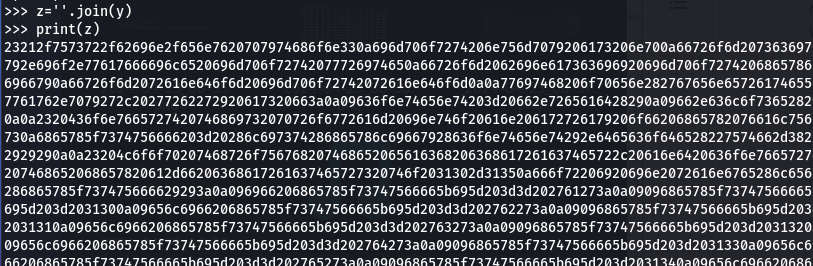
After carefully observing the unique samples, I noticed numbers closer to zero with levels around 16. To prepare the data for further analysis, I performed a transformation. This involved creating an array, dividing it by 100, then dividing by 5, and finally subtracting 2 to convert the data into hexadecimal values.



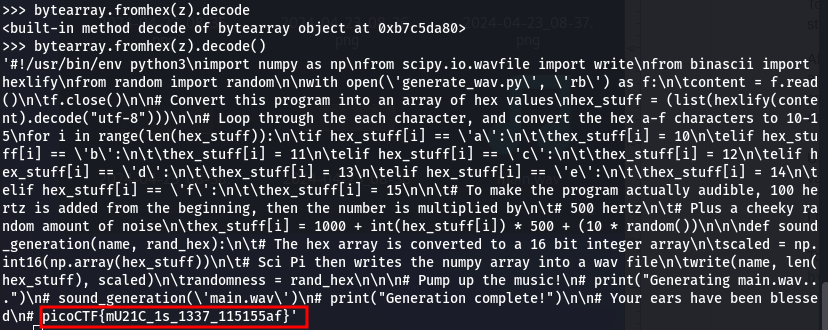
Using the string.hexdigits module, I converted the transformed data into hexadecimal values.



I then joined the hexadecimal characters into a single string.



To decode the hexadecimal string into ASCII characters, I used the bytearray.fromhex().decode() method. This step revealed the flag.



By following these steps, I successfully obtained the flag: picoCTF{mU21C\_1s\_1337\_115155af}.

