vishwaCTF

CHALLENGE NAME :

[MYSTERIOUS OLD CASE]

DEV :

[ABHISHEK MALLAV]

CATEGORY :

[STEGNOGRAPHY]

LEVEL :

[MEDIUM]

2024

vishwaCTF

CHALLENGE NAME : [MYSTERIOUS OLD CASE]

Challenge Description :

You as a FBI Agent are working on a old case involving a ransom of $200,000 after some digging you recovered an audio recording.

Solution :

The Audio is reversed and pasted in between the music clips.

After reversing the audio the following information is revealed

I am Dan Cooper it is 24/11/1971, now I have left from Seattle and headed towards Reno. I have got all my demands fulfilled. I have done some changes in the flight log and uploaded it to a remote server, the file is encrypted the hint for decryption is the airliner that I am flying in. Most importantly the secret key is split and hidden at every element of the Fibonacci series starting from 2.

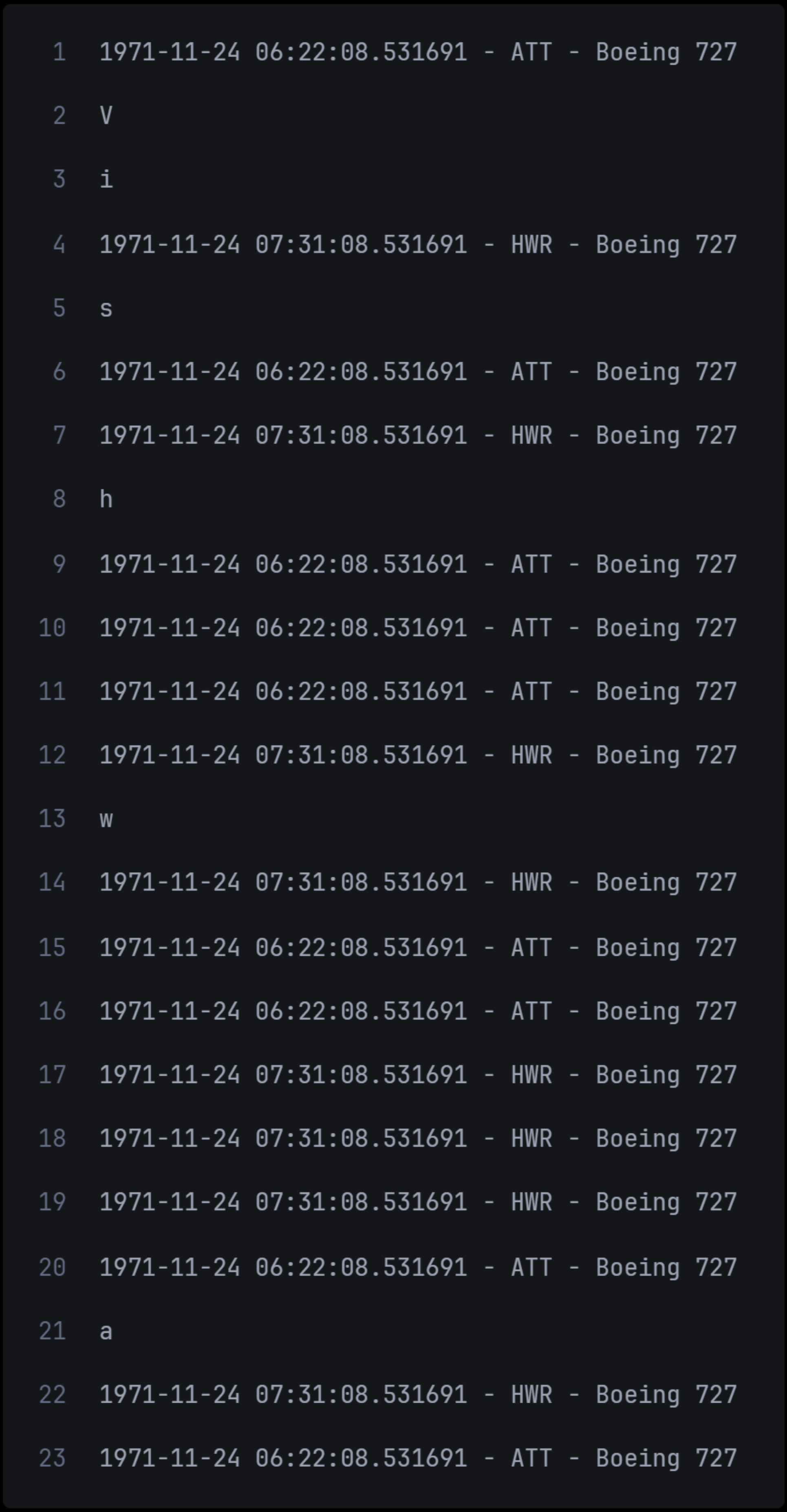
The Challenge is based on the mysterious case of Dan Cooper (DB Cooper) which happened in 1971. So there are some reference in the text above.

After Looking into the metadata of the audio file we find more clues.

- the zip file is 100 MB not 7 GB
- DB Cooper
- 727/305
- 1971
- password for the zip is all lowecase with no spaces
- https://drive.google.com/file/d/1bkuZRLKOGWB7tLNBseWL34BoyI379QbF/view?usp=drive_lin

- After opening the Google Drive link we get the flight_log.zip file as mentioned in the text.
- The Zip is password protected and the hints for its decryption are given in the text and the metadata *(hint for decryption is the airliner that I am flying in), (password for the zip is all lowecase with no spaces)* password is **northwestorientairlines**
- After extracting the zip file we get 1000 flight logs.

- Now based on the information in the metadata and the actual flight DB Cooper was flying in the flight log to look for is flight 305. (all other flight logs have 2024 in their year and only flight 305 has the year 1971 and the plane Boeing 727)
- After opening the flight log of flight 305 we can see a pattern of the flag appearing



```
1971-11-24 06:22:08.531691 - ATT - Boeing 727
V
i
1971-11-24 07:31:08.531691 - HWR - Boeing 727
s
1971-11-24 06:22:08.531691 - ATT - Boeing 727
1971-11-24 07:31:08.531691 - HWR - Boeing 727
h
1971-11-24 06:22:08.531691 - ATT - Boeing 727
1971-11-24 06:22:08.531691 - ATT - Boeing 727
1971-11-24 06:22:08.531691 - ATT - Boeing 727
1971-11-24 07:31:08.531691 - HWR - Boeing 727
w
1971-11-24 07:31:08.531691 - HWR - Boeing 727
1971-11-24 06:22:08.531691 - ATT - Boeing 727
1971-11-24 06:22:08.531691 - ATT - Boeing 727
1971-11-24 07:31:08.531691 - HWR - Boeing 727
1971-11-24 07:31:08.531691 - HWR - Boeing 727
1971-11-24 07:31:08.531691 - HWR - Boeing 727
1971-11-24 06:22:08.531691 - ATT - Boeing 727
a
1971-11-24 07:31:08.531691 - HWR - Boeing 727
1971-11-24 06:22:08.531691 - ATT - Boeing 727
```

The pattern of the flag is in the Fibonacci sequence starting from 2.

- So to extract the flag without looking for the keywords of the flag manually.

  We need to develop a program to do so.

  At first the keywords are easy to find but soon it gets tedious the flight log has 300,000 lines of text

  Here is an example of a python script to extract the flag.

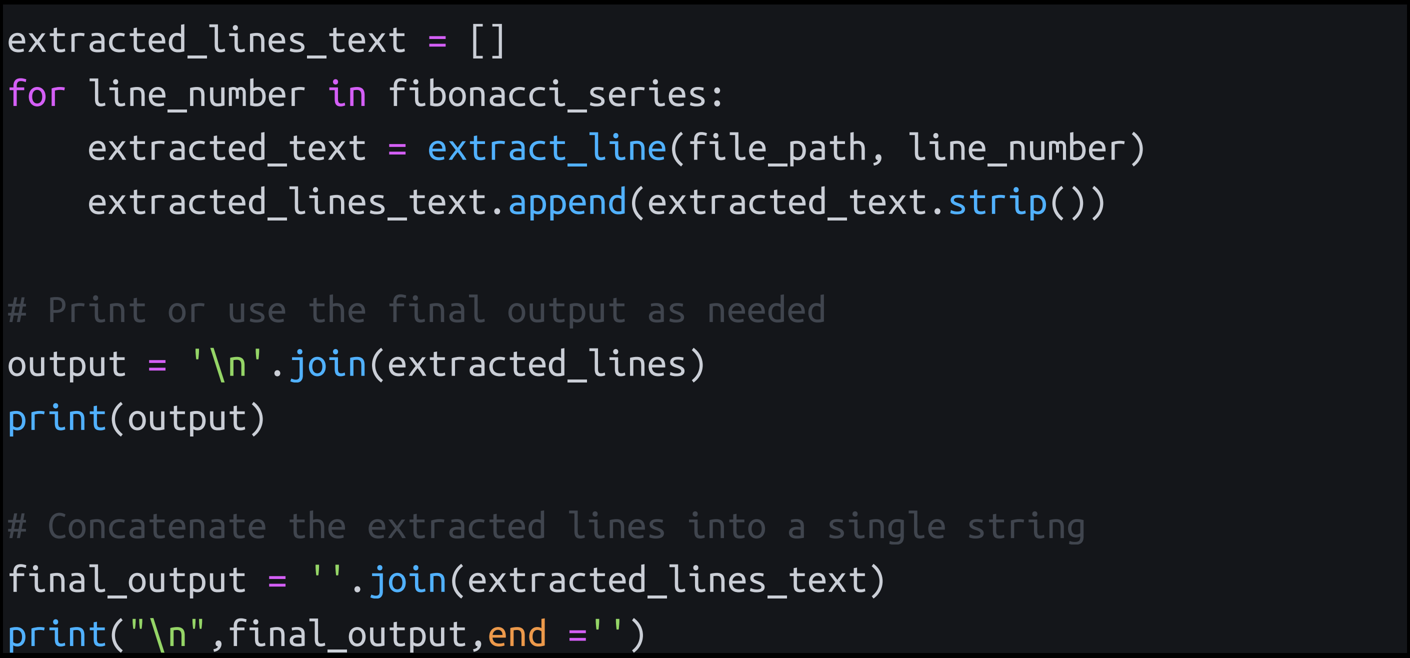
```python
# Making a list of Fibonacci Series
def fibonacci(n):
    fib_series = [1, 2]
    while len(fib_series) < n:
        fib_series.append(fib_series[-1] + fib_series[-2])
    return fib_series

# Function to Extract text from a line number
def extract_line(file_path, line_number):
    with open(file_path, 'r') as file:
        lines = file.readlines()
        if 2 <= line_number <= len(lines):
            return lines[line_number - 1]
        else:
            return f"Line number {line_number} is out of range."

# Replace 'your_file.txt' with the actual path to your text file
file_path = 'Flight-305.txt'

# Get the Fibonacci series starting from 2 to exclude the first line
fibonacci_series = fibonacci(26)[1:]  # Adjust the parameter based on your needs

# Extract text from lines in the Fibonacci series
extracted_lines = []
for line_number in fibonacci_series:
    extracted_text = extract_line(file_path, line_number)
    extracted_lines.append(f"Line {line_number}: {extracted_text.strip()}")
```

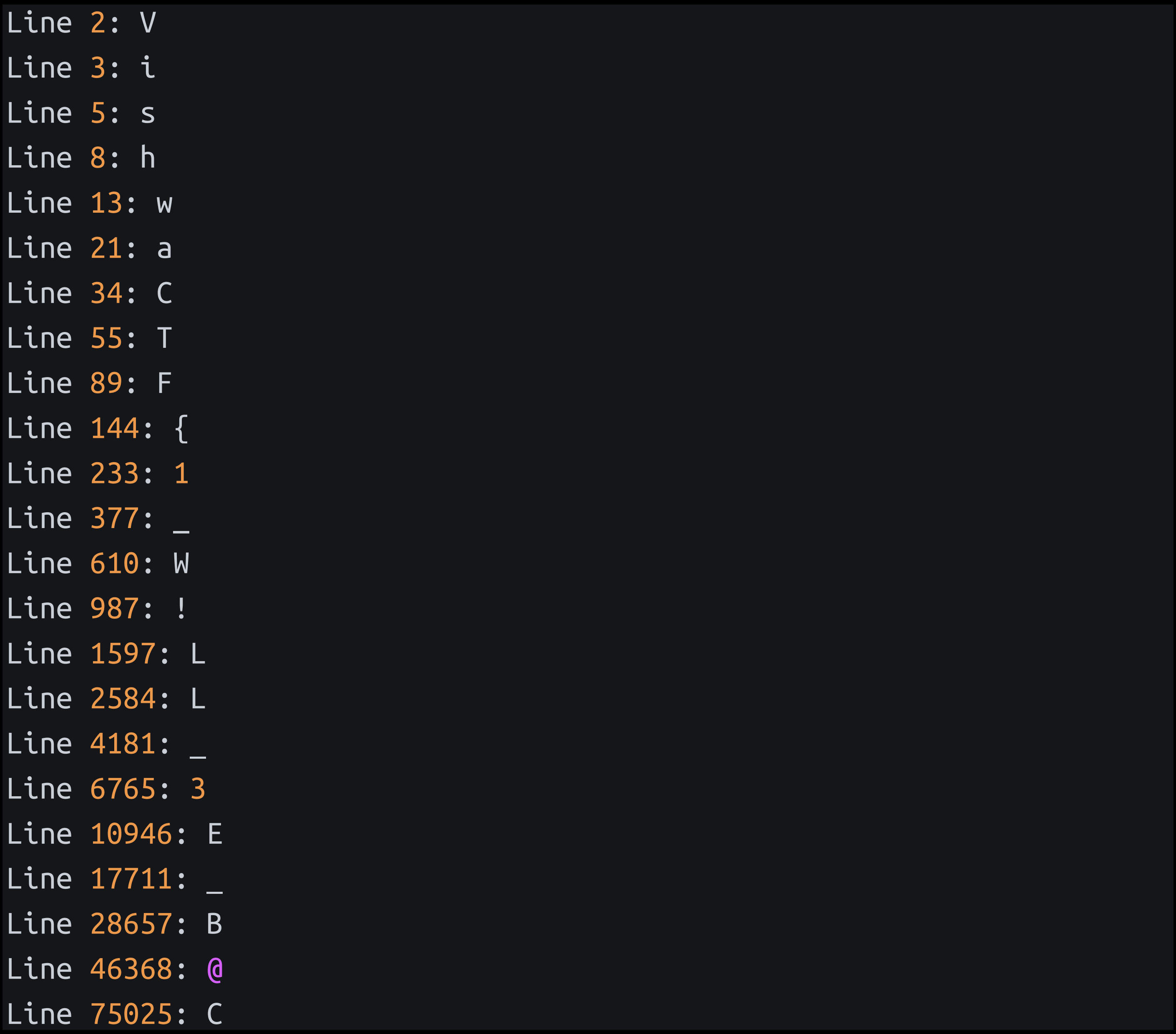


```python
extracted_lines_text = []
for line_number in fibonacci_series:
    extracted_text = extract_line(file_path, line_number)
    extracted_lines_text.append(extracted_text.strip())

# Print or use the final output as needed
output = '\n'.join(extracted_lines)
print(output)

# Concatenate the extracted lines into a single string
final_output = ''.join(extracted_lines_text)
print("\n",final_output,end ='')
```

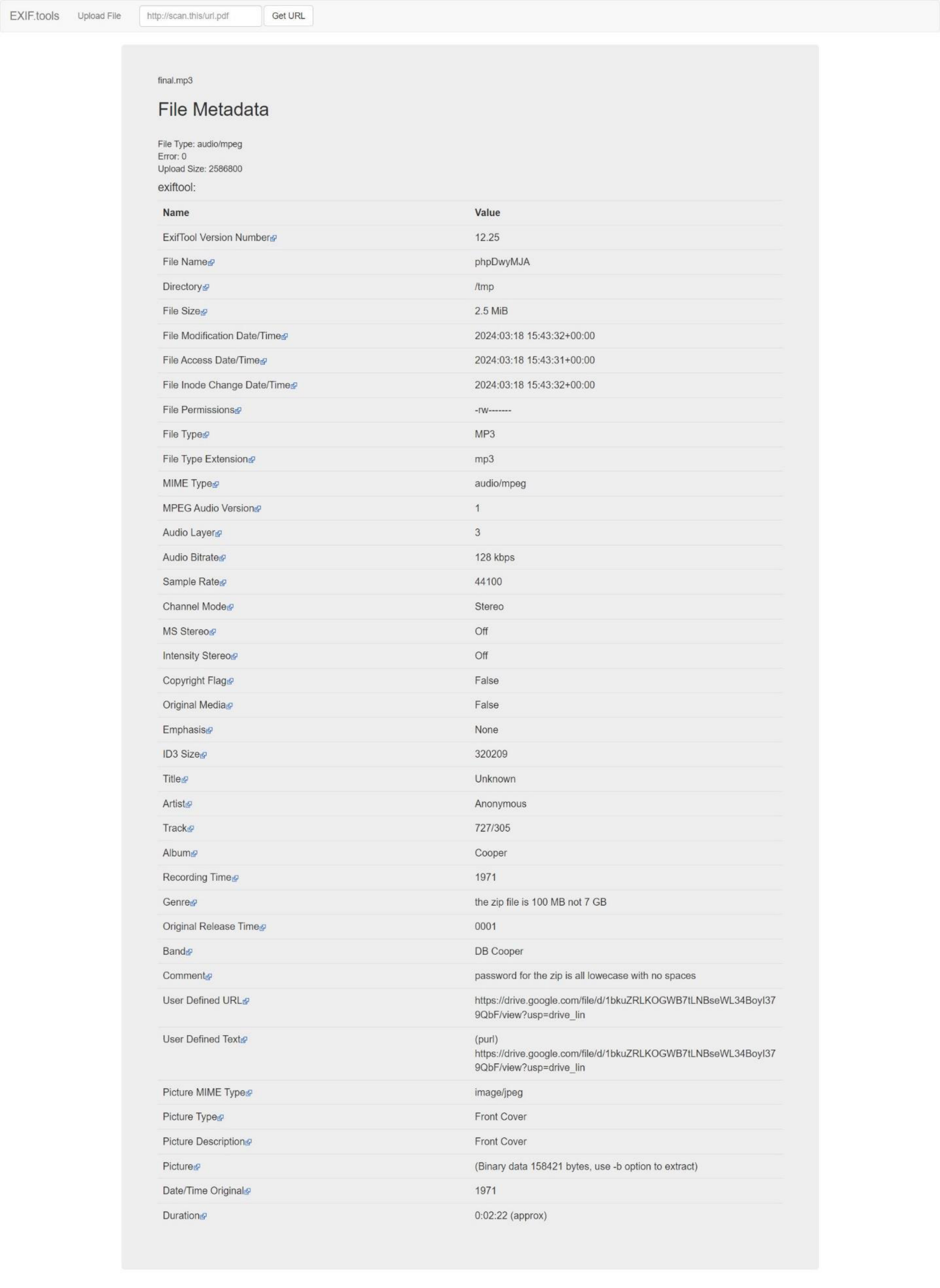
The Output of the Script is



```
Line 2: V
Line 3: i
Line 5: s
Line 8: h
Line 13: w
Line 21: a
Line 34: C
Line 55: T
Line 89: F
Line 144: {
Line 233: 1
Line 377: _
Line 610: W
Line 987: !
Line 1597: L
Line 2584: L
Line 4181: _
Line 6765: 3
Line 10946: E
Line 17711: _
Line 28657: B
Line 46368: @
Line 75025: C
Line 121393: K
Line 196418: }

VishwaCTF{1_W!LL_3E_B@CK}
```

Metadata of the audio



final.mp3

File Metadata

File Type: audio/mpeg
Error: 0
Upload Size: 2586800

exiftool:

Name	Value
ExifTool Version Number	12.25
File Name	phpDwyMJA
Directory	/tmp
File Size	2.5 MiB
File Modification Date/Time	2024:03:18 15:43:32+00:00
File Access Date/Time	2024:03:18 15:43:31+00:00
File Inode Change Date/Time	2024:03:18 15:43:32+00:00
File Permissions	-rw-------
File Type	MP3
File Type Extension	mp3
MIME Type	audio/mpeg
MPEG Audio Version	1
Audio Layer	3
Audio Bitrate	128 kbps
Sample Rate	44100
Channel Mode	Stereo
MS Stereo	Off
Intensity Stereo	Off
Copyright Flag	False
Original Media	False
Emphasis	None
ID3 Size	320209
Title	Unknown
Artist	Anonymous
Track	727/305
Album	Cooper
Recording Time	1971
Genre	the zip file is 100 MB not 7 GB
Original Release Time	0001
Band	DB Cooper
Comment	password for the zip is all lowecase with no spaces
User Defined URL	https://drive.google.com/file/d/1bkuZRLKOGWB7tLNBseWL34Boyl379QbF/view?usp=drive_lin
User Defined Text	(purl) https://drive.google.com/file/d/1bkuZRLKOGWB7tLNBseWL34Boyl379QbF/view?usp=drive_lin
Picture MIME Type	image/jpeg
Picture Type	Front Cover
Picture Description	Front Cover
Picture	(Binary data 158421 bytes, use -b option to extract)
Date/Time Original	1971
Duration	0:02:22 (approx)