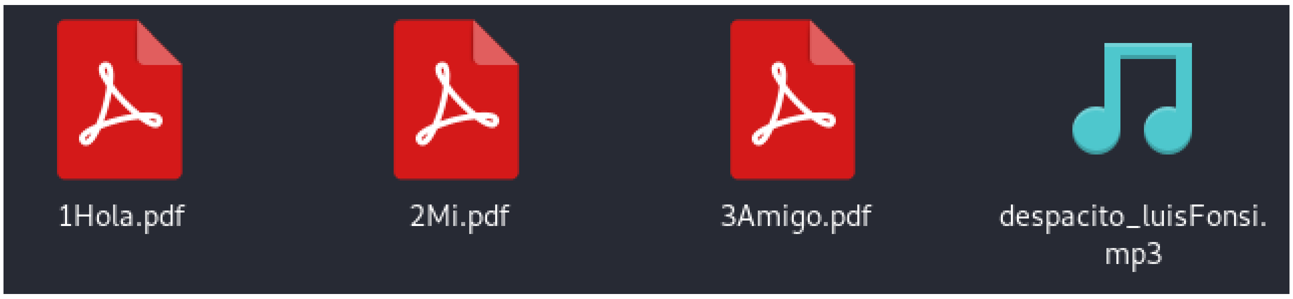
Forensics : Encuéntralo si puedes

Value :	350 Pts
Description :	Luis is very fond of music. Recently he developed a keen interest in CTF challenges. He makes a challenge for yankee and asks him if he could break it and find the code from it. Help yankee to find the secret code. The flag format - p_ctf{OBTAINED_SECRET_CODE}
Attachment :	main.zip

Solution

First let's download the attachment «main.zip» and extract his content. We get a mp3 file and three pdf password protected.



Using ffmpeg against the mp3 file for get some informations and i was able to get a potential hint in comment.

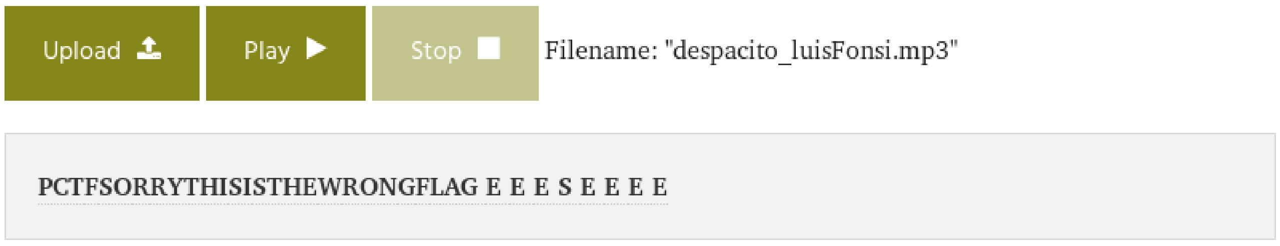
```
root@kali:~/main# ffmpeg -v info -i despacito_luisFonsi.mp3 -f null -
```

```
Output #0, null, to 'pipe:':
  Metadata:
    track           : 01
    Software        : Lavf58.33.100
```

```
artist          : Luis Fonsi ft. Daddy Yankee
genre           : POP
Unknown text information frame: 2017
title           : Despacito
comment         : Better go last than first
```

Additionnaly if we listen the music, we can hear morse code at the beginning. Using a morse audio decoder and it was a rabbit hole.

Source : https://morsecode.world/international/decoder/audio-decoder-adaptive.html

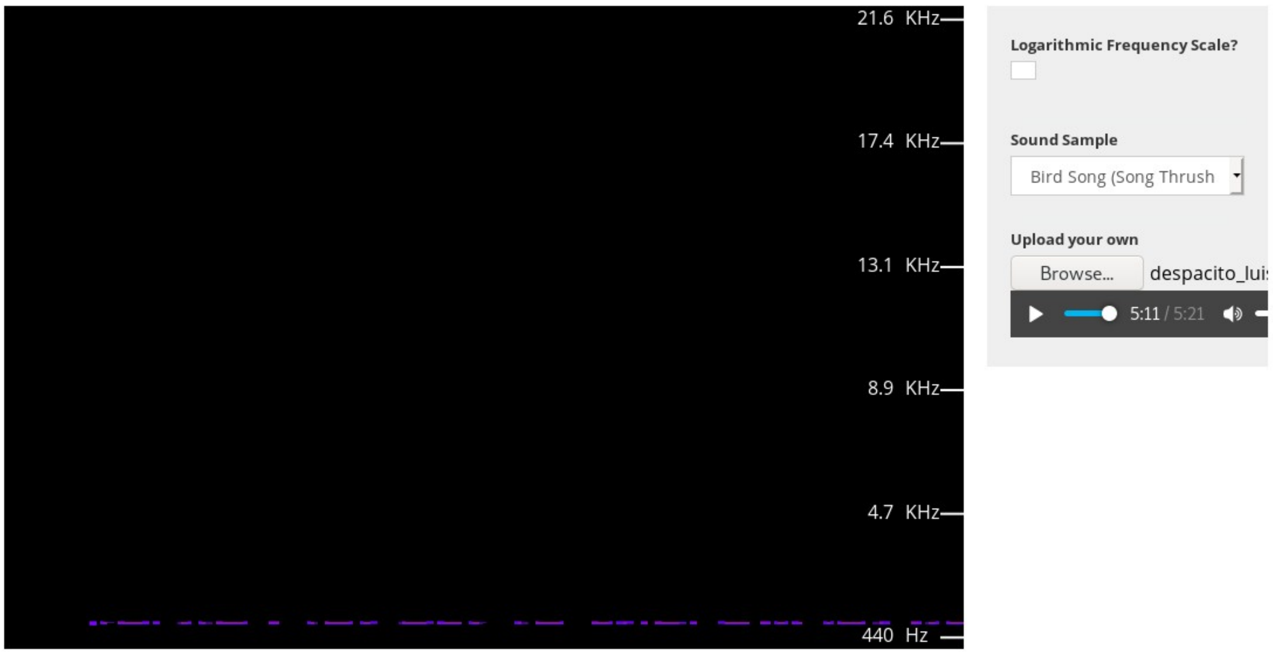


Filename: "despacito_luisFonsi.mp3"

PCTFSORRYTHISISTHEWRONGFLAG E E E S E E E E

But thinking of the comment «Better go last than first», i go at the end of the song, the music continue but there is no sound, seem strange. So i started a spectrum analyzer and started to analyze the spectrum.

Source :https://academo.org/demos/spectrum-analyzer/



Logarithmic Frequency Scale?

Sound Sample: Bird Song (Song Thrush

Upload your own: despacito_lui

5:11 / 5:21

As we can see, at approximatively 5min07 of the song, the morse code appear.

In the morde code, there is an unknown character in five signals. We can guess that it is a space.
Decrypting the message character by character gives ‘**fuerza brute de cinco digitos con minusculas y numeros**’. It seems to be a hint to crack the password of a pdf file.

Decoded from spanish it mean, **brute force of five digits with lowercase and numbers**.

Once we got our instructions from the morse code, we can generate a wordlist with crunch with the parameter, 5 digits, contain numbers and minuscule letter.

```
root@kali:~# crunch 5 5 0123456789abcdefghijklmnopqrstuvwxyz -o
/root/Bureau/custom_wordlist.txt
Crunch will now generate the following amount of data: 362797056 bytes
345 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 60466176

crunch:  88% completed generating output

crunch: 100% completed generating output
```

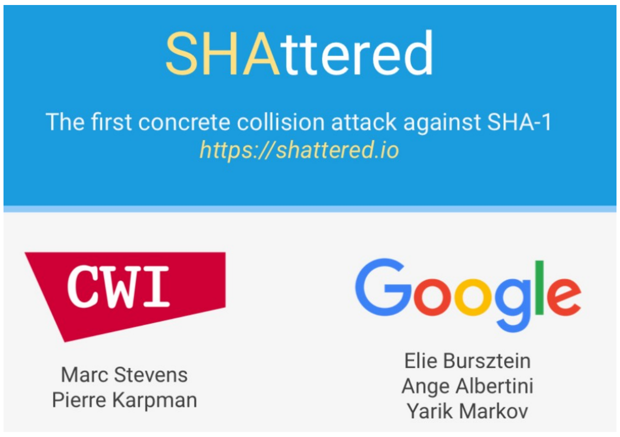
And now we can crack the three pdf with this wordlist and the tool called pdfcrack.

Crack 1Hola.pdf :

```
root@kali:~/Téléchargements/main# pdfcrack -f 1Hola.pdf -w
/root/Bureau/custom_wordlist.txt
PDF version 1.3
Security Handler: Standard
V: 2
R: 3
P: -3904
Length: 128
Encrypted Metadata: True
FileID: 34ef3f1f94c5a1a642014ddf22af7900
U: 5233de370d2758db5857a2a21592631c00000000000000000000000000000000
O: fc81cb565ad34c0f8d431ec8772e44d7ee7da867715f50294bfe5b23116564c4

found user-password: 'x2n1z'
```

1Hola.pdf content :

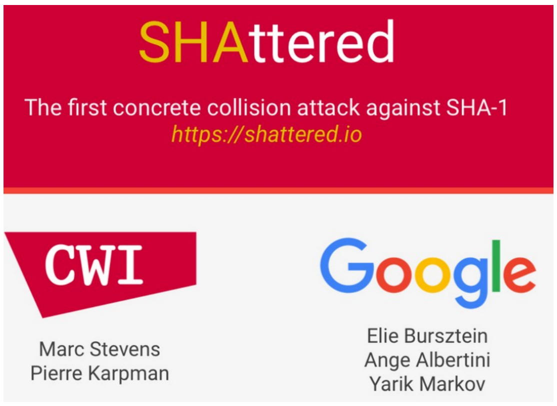


Crack 2Mi.pdf

```
root@kali:~/Téléchargements/main# pdfcrack -f 2Mi.pdf -w
/root/Bureau/custom_wordlist.txt
PDF version 1.3
Security Handler: Standard
V: 2
R: 3
P: -3904
Length: 128
Encrypted Metadata: True
FileID: 4d97190be1720380d8d767624b7fc47d
U: f957b7fc930d9df6711936428571ca4c00000000000000000000000000000000
O: b78ac4035ad34c0f8d431ec8772e44d7ee7da867715f50294bfe5b23116564c4

found user-password: '39adz'
```

2Mi.pdf content :

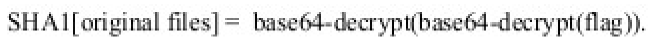


Crack 3Amigo.pdf :

```
root@kali:~/Téléchargements/main# pdfcrack -f 3Amigo.pdf -w
/root/Bureau/custom_wordlist.txt
PDF version 1.0
Security Handler: Standard
V: 2
R: 3
P: -3904
Length: 128
Encrypted Metadata: True
FileID: cc995aa02345d11dffa5c238757d18b7
U: 23e1ecefcd7290ebf2bd93ee473b907500000000000000000000000000000000
O: bccac30612d34c0f8d431ec8772e44d7ee7da867715f50294bfe5b23116564c4

found user-password: '8yfa2'
```

3Amigo.pdf content :


SHA1[original files] = base64-decrypt(base64-decrypt(flag)).

The content of **1Hola.pdf** and **2Mi.pdf** show a pdf of a website talking about collision attack.

Source : https://shattered.io/

This attack show the possiblity to have same SHA1 signature in two pdf, so we can easilly signe with a valid SHA1 a malicious pdf. (Read the blog for more)

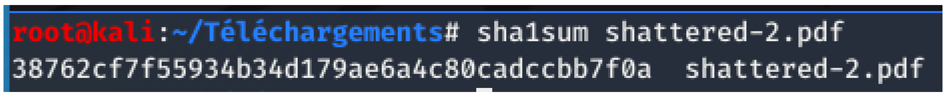
The content of **3Amigo.pdf** show the flag format, it’s the SHA1 of original files, then encoded in base64.

As our pdf **1Hola.pdf** and **2Mi.pdf** is exactly the same than the PoC used in the shattered.io website, and the SHA1 of those original pdf is exactly the same due to the collision attack, i assume that this is our pdf original.

Download one of them.

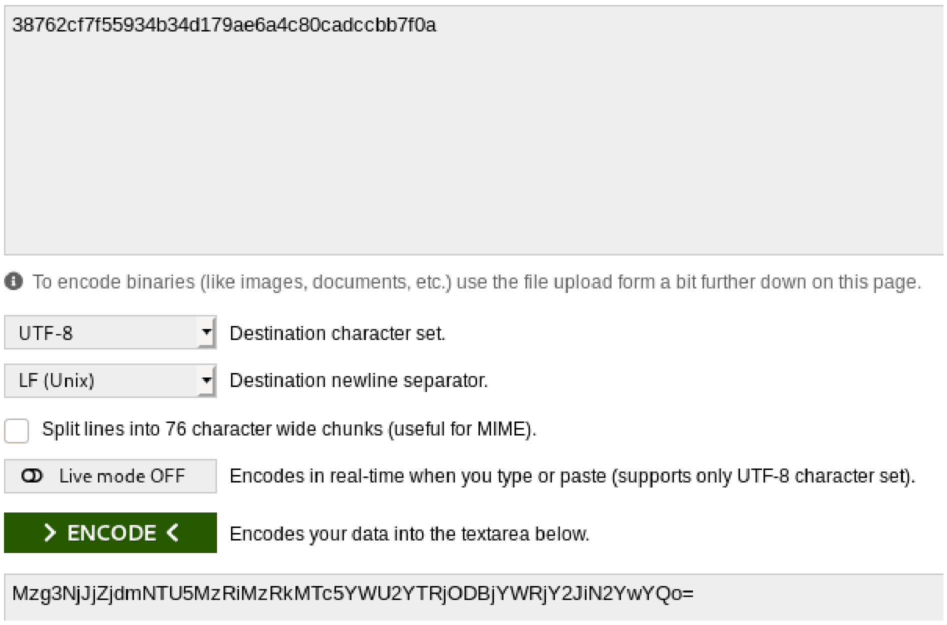
Source : https://shattered.io/static/shattered-1.pdf
or
Source : https://shattered.io/static/shattered-2.pdf

Then retrieve the SHA1 of the file.



```
root@kali:~/Téléchargements# sha1sum shattered-2.pdf
38762cf7f55934b34d179ae6a4c80cadccbb7f0a  shattered-2.pdf
```

Now encode the SHA1 in base64.



```
38762cf7f55934b34d179ae6a4c80cadccbb7f0a
```

To encode binaries (like images, documents, etc.) use the file upload form a bit further down on this page.

UTF-8 — Destination character set.

LF (Unix) — Destination newline separator.

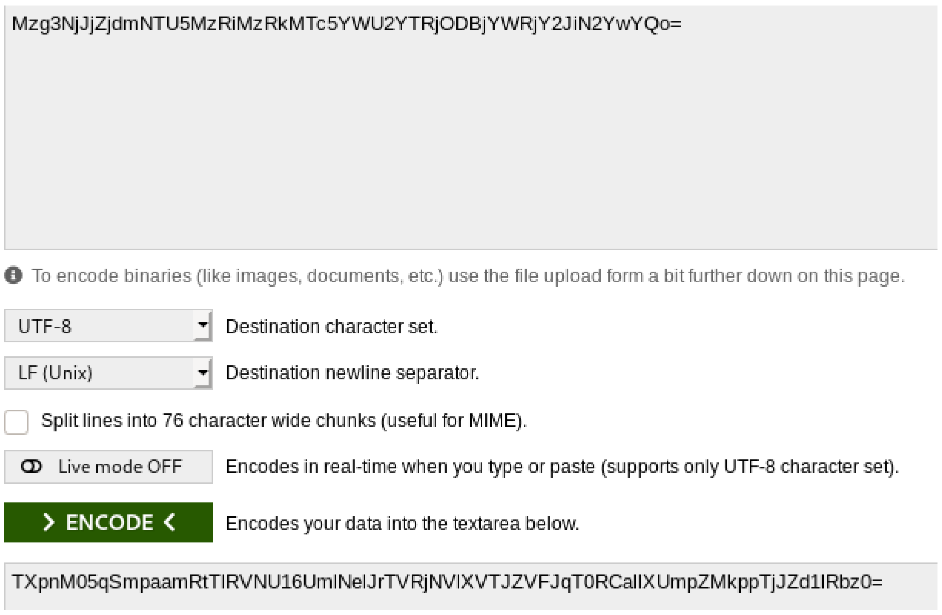
☐ Split lines into 76 character wide chunks (useful for MIME).

Live mode OFF — Encodes in real-time when you type or paste (supports only UTF-8 character set).

Encodes your data into the textarea below.

```
Mzg3NjJjZjdmNTU5MzRiMzRkMTc5YWU2YTRjODBjYWRjY2JiN2YwYQo=
```

And then encode the base64 to base64 again.



```
Mzg3NjJjZjdmNTU5MzRiMzRkMTc5YWU2YTRjODBjYWRjY2JiN2YwYQo=
```

To encode binaries (like images, documents, etc.) use the file upload form a bit further down on this page.

UTF-8 — Destination character set.

LF (Unix) — Destination newline separator.

☐ Split lines into 76 character wide chunks (useful for MIME).

Live mode OFF — Encodes in real-time when you type or paste (supports only UTF-8 character set).

Encodes your data into the textarea below.

```
TXpnM05qSmpaamRtTlRVNU16UmlNelJrTVRjNVlXVTJZVFJqT0RCallXUmpZMkppTjJZd1lRbz0=
```

And we get our flag !

Flag :

p_ctf{TXpnM05qSmpaamRtTlRVNU16UmlNelJrTVRjNVlXVTJZVFJqT0RCallXUmpZMkppTjJZd1lRbz0=}