disco_dance

"I heard disco parties can be pretty random and chaotic. Let’s see just how chaotic. Flag format: CTF{sha256}"

To solve this challenge first i needed to understand the way the encryption worked first. Luckily it wasn't anything too complicated:

- The server gets the last 5 messages from a Discord channel.
- It concatenates their contents (newest first) to make a "seed."
- The AES key is SHA256(seed).
- The flag is encrypted with AES-CBC using a random IV.
- The output is base64(iv + ciphertext).
  So if you know those 5 messages, you can reconstruct the key and decrypt the flag; the system is only as "random" as the Discord channel’s latest messages

Said and done, I managed to insert 5 messages manually and immediately after, I connected to the challenge and got an otput.

From here all I had to do was run a script with the info I gathered:

```
import base64, hashlib
from Crypto.Cipher import AES
from Crypto.Util.Padding import unpad

# Messages (oldest -> newest)
M1, M2, M3, M4, M5 = "a1", "a2", "a3", "a4", "a5"

seed = M5 + M4 + M3 + M2 + M1   # "a5a4a3a2a1"
print("Seed:", seed)
print("SHA256(seed):", hashlib.sha256(seed.encode()).hexdigest())

ciphertext_b64 =
"KwLvXKeNjS9Rzsdj4bwYJIfaavb7w+kKXA2uzWzHVxRRIKykbhQyb5tV0KggRtYkq3iw6LCKam7
LINiXloV5op9hzNwZPIYtf3raCDJFPu+Hv7VJp5MwqTZf7KPudx1v"

raw = base64.b64decode(ciphertext_b64)
iv, ct = raw[:16], raw[16:]
cipher = AES.new(hashlib.sha256(seed.encode()).digest(), AES.MODE_CBC, iv)
pt_padded = cipher.decrypt(ct)

try:
```

```
    pt = unpad(pt_padded, AES.block_size).decode('utf-8', errors='replace')
    print("PLAINTEXT:", pt)
except ValueError:
    print("BAD PADDING (seed probably not the real last 5 messages).")
```

```
└─$ python3 dance.py
Seed: a5a4a3a2a1
SHA256(seed): e90494640352c6b76f07c550423d508d18e64379e278f5b3b6e3030b051779a5
PLAINTEXT: CTF{f55ba4939edd5611a7ab797529b51dae47989b3c5a99f2ffc82e4b2c74d03e56}
```

flag: CTF{f55ba4939edd5611a7ab797529b51dae47989b3c5a99f2ffc82e4b2c74d03e56}