



Challenge: Feed (100 pts) (Cryptography)

Challenge 8 Solves X

Feed

100

verified

Just give the flag!!!!!!!.

```
11111111100100010101011101011111101011111  
111110110110110110000110100100100010100101  
11111100101000000000000000000000000000000000000  
0000010101111110010101111111111111111111100  
1010011111111111111100101001000000000000000000  
0000000000000000000000000101001101111111111111  
1111110010100010100100100000000010100111111  
111111100101011111110010100111110010100010  
1001000000000101011111001010010001010011000  
00000101001011111111111111001010
```

2/5 attempts

Flag Submit

We were given a long binary string and asked to find the flag.

How I solved it

At first, I was not sure how to approach the binary-looking code since it did not resemble ordinary binary or common encodings. So, I used dCode's Cipher Identifier tool, which is designed to automatically analyze encrypted messages and suggest possible ciphers. This tool uses frequency analysis, coincidence index, and signature detection to identify the type of encoding or cipher applied. I input the code into the tool and tried many of the suggested cipher options from its extensive list, but none worked at first.

The tool lists many possible ciphers including ASCII code, XOR cipher, Gray code, Bacon cipher, and Spoon, among others. Since the challenge name was "Feed," it made me think of "spoon feeding." So I focused on the Spoon cipher option. Using the Spoon decoder on dCode, I finally got the correct decoded message which was the flag. This process shows how the Cipher Identifier helps narrow down the possibilities, but some reasoning and guesswork based on clues is still important to solve such challenges.

Flag: Flag{Sp00n_f33ding}

Summary:

The challenge name hinted at the Spoon language. By connecting the dots from the name Feed → Spoon, and using resources like the esolang wiki and dCode, I decoded the binary string to retrieve the flag.

~By Team justahacker