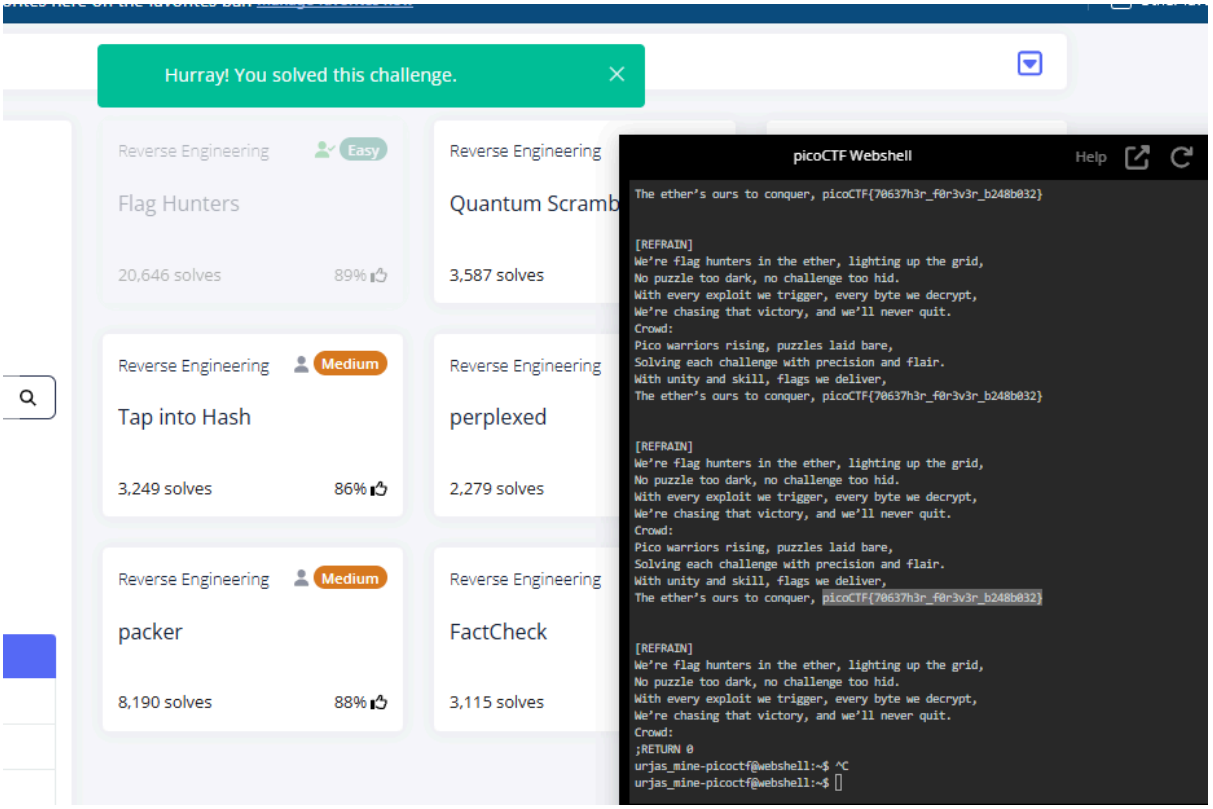
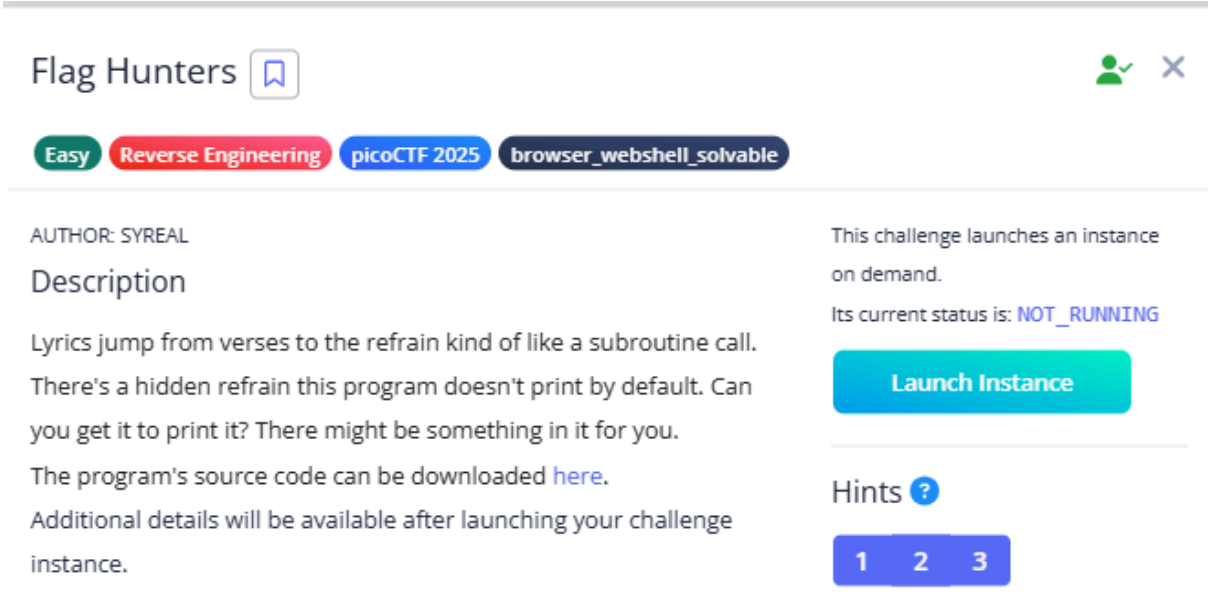
JASMINE C. OMANDAM





When I first opened the challenge *Flag Hunters* on picoCTF, I noticed the description compared verses and refrains to subroutine calls. That immediately made me think: there's a hidden function or output that isn't printed by default, and my job is to coax it out.

I launched the instance and connected with netcat:

bash
nc verbal-sleep.picoctf.net <port>

The program greeted me with poetic verses about hacking, exploits, and flag hunting. At the end of each stanza, it prompted me with:

Code
Crowd:

That prompt was my entry point. The hints reinforced my suspicion:

- "Unsanitized user input is always good, right?" → the program was evaluating my input directly.
- "Is there any syntax ripe for subversion?" → I should try calling functions or injecting code.
- "Lyrics jump from verses to the refrain…" → there must be a hidden function named something like `refrain`.

So I typed:

refrain()

Suddenly, the program printed new verses — different themes like forensics, binary reversing, cryptography, and web exploitation. Each time I called `refrain()`, it revealed another hidden stanza. I kept repeating the call, cycling through these hidden refrains.

Finally, after enough iterations, the program printed a special line that stood out from the lyrical text:

picoCTF{70637h3r_f0r3v3r_b248b032}

That was the flag. I copied it directly from the netcat output and submitted it on the picoCTF site.

Reflection

The challenge was a clever play on the idea of subroutine calls. By recognizing that my input was unsanitized and directly evaluated, I realized I could invoke hidden functions. Persistence repeatedly calling **`refrain()`** — eventually revealed the flag embedded in the poetic output. It was a simple but elegant exercise in spotting hidden functionality and exploiting unsanitized input.