Name of challenge: **Fantaxotic Fledgling**
Category: binex
Difficulty: medium
Brief overview: Fully protected 64 bit binary. We create a fake 2 byte canary and place it on the stack, the program asks a user to send a message to the server, after they send the message the program will exit and claim that the user's message has been sent. We will use a vulnerable gets when the user sends this message so they can overflow the message buffer and overwrite the canary. Below the canary lies a locking byte with the digit 1 in its place, if set to 0, before the program closes the user gets redirected to a win function. But they need to properly brute force the 2 byte canary, if the canary is detected changed in any way the program immediately exits. The reason we chose 2 bytes is because there are 65,536 different possible port values you can connect to so theoretically if we opened all the ports on a web server someone can one shot the solution with a simple python program. But the idea of the challenge is still difficult because most people have never learned what a stack canary is and their not going to get the source code in C, just the assembly of a localized version running.

Vulnerability
The function gets() is used to read input without bounds checking, allowing a buffer overflow. Attackers can overwrite the canary and lock variable, but must restore the canary's original value to avoid detection. If they successfully flip the lock to 0, the program jumps to the win() function.

Exploitation Strategy
To win, an attacker must:
-Overwrite the lock variable (one byte) to 0.
-Preserve the original canary value.
-Brute-force the 2-byte canary, with 65,536 possibilities.
-Because gets() does not check input length, they can overwrite both values, but they need to correctly guess the canary. If they control all possible TCP ports, they can attempt all 65,536 values simultaneously.