**strops**

I completed this challenge using dynamic analysis. This might not have been the fastest route, but I could not figure out how to read the flag from the assembly code. This binary depicts a program that loops over a string the user entered and compares it to the flag of this challenge. I used GDB (with gef) to loop over the program and stop it when it compared each letter the user entered with a letter of the flag. Thus, I was able to piece together the flag letter by letter. That is how I solved this challenge.

Edit: I realize now that there was a tilde in front of the character of the flag, so I was supposed to flip the bits of the hex numbers I saw in the disassembler output and find the ascii of the hex result