I first opened up question 6 in Ghidra and went to main to see what I was working with. There was only a print statement saying “sorry, not this time” so I knew I would have to find the flag elsewhere.

A screenshot of a computer program

Description automatically generated

I then decided to search in strings to see if I could find any interesting strings I could follow that could possibly take me to the flag and I found a string that said “decrypted message” so I clicked and followed that and found the function called “nowayhome”. I also noticed the jumbled up letters below it that possibly looked like an encrypted flag.

A screenshot of a computer

Description automatically generated

I found what looked like an encrypted/scrambled flag in “nowayhome”. I then focused in on the function called “garbledygoop” because it resembles the format of a encrypt\_decrypt function. I also noticed the print statement (“Decrypted Message”)

A screen shot of a computer

Description automatically generated

When I clicked on the garbeledygoop function to look closer at it, I got the same conclusion that it was most likely a decryption method. I noticed the type of parameters that garbledygoop took in. The parameters were char, int and long. I thought it was interesting that it took in an int.

A screenshot of a computer program

Description automatically generated

Now I had to think about how to go about actually getting the decrypted flag. I knew that I first needed to have the “nowayhome” to be called since main was just instantly printing “sorry, not this time”. To edit this, I opened up Cutter. I changed the printf to instead call nowayhome.

A computer screen shot of text

Description automatically generated

A computer screen shot of a program code

Description automatically generated

I then tired to run it in qemu. I of course opened up two terminals so I could run qemu. When I just tried to run it, I got a segmentation fault

A black background with white text

Description automatically generated

Because of this segmentation fault, I realized I needed an argument. I then went back and looked at garbledygoop and tried a couple of different things and this is when I really focused on the types of parameters and noticed the int more clearly and that it needed an integer argument. I started with 1 and just kept adding 1 each time until I was able to get the flag at integer 9.

Here is the flag below:

A screen shot of a computer

Description automatically generated