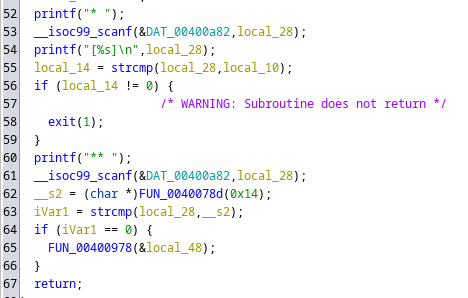
Rand strings against the executable and found the following thing:

SuperSeKretKey

If i answer the first prompt with that string, it will give me another prompt. Answering wrong for either the first or the second one will end the program. The first prompt has only one asterisk (\*), while the second has two.

Given the above information, I assume that I now have to find the answer to the second prompt in order to progress through the challenge.

Looking through the functions in ghidra, I found an interesting one: FUN\_0040085d, in which there was the above mentioned string and the two prompts:



Figure

There is a comparison between the answer to the second prompt (local\_28) and \_s2 which would probably be the correct answer.

Unfortunately, the function that generates the correct answer is pseudorandom:



Figure

( by that I mean the use of the random variable iVar1 for generating the answer )

This probably suggests the need of dynamic analysis by debugging it with gdb.

I was a little bit confused by the 9th line but I think that should be equivalent with “pvVar3[param1] = 0”, basically putting the null terminator to the end of the string.

param1 should be the length of the answer so If we look back at the function call from Figure 1, on line 62, 0x14H ( 20 decimal ) is the length of the answer in our case.

Before trying to dynamically analyze the executable, I first wanted to see where does the correct answer take us, so I opened FUN\_00400978 from line 65 in Figure 1.