VCU RE Software challenge

Rough Notes

Initial Notes

- I know that the seed is not static and will require dynamic analysis to solve
- hex chars and ascii will be needed
- write a python script with decription methods to open it and send correct input.

Static Analysis

- we see that it is a windows .exe file
- size_t is a special unsigned int type in c and c++
- going through ghidra and trying to name all the functions to see what is happening, will go back and do this in ida as well.
- I see similar things to what I see. I made better names this time around since I had more forsight.
- I see the same strings
- It seems that my dynamic analysis will be what changes this time around.
- I see that the seeded string gets stored into an undefine variable, I think this could be useful.
- The seeded value is created with time()
-

Dynamic Analysis

Ideas

- I will need a python script to perform the decryption in real time.
- I will start be doing the low langing fruit. I will try and do it all in hex to keep solid fundamental.
- At the end I may need to turn certain characters into ascii and leave some as hex characters.
- I need to see how I will generated the time value. I think I might have to stop at the debugger to do this. I think what I will do is store them all in a list to do this.
- To automate it I might want to see what the **time()** does.
- `time(0)` is getting the current system time in the unix format. There is a way to do this in python. I may need to try and run the .exe with a password in python.
- I also need to make sure I get the c equivalent version of the c commands.
- I think I should be good to just use the system time for the python script.
- My first goal is to run and make sure I have the seeded string correct.

- I had to use some manipulation of the xor function to rewrite the algorithm in my code.
- I am looking into using (ctypes)[https://stackoverflow.com/questions/75535673/how-to-mimic-rand-function-of-c-in-python] module to see if I can auto generate the same results.

Ida analysis

- I am `rand()` will store the value in eax, I am making sure it lines up with what I get in python. if that is correct than my code should be accurate to generate the correct code. I just need the script to run the .exe at the same time as the python script..

d the string that I am looking for.

- This will be useful for the (hex-characters)[https://stackoverflow.com/questions/41559398/bash-type-characters-in-hexadecimal-notation-to-standard-input]
  -Run the script with a text file, and then see if we get the same times
- The time is seeded once the user enters in the password.
- If I get the matching time variable I should then be able to calculate the password in python to enter.
- It seems that I may not be able to generate a password in real time. I may need to take it from the program some how.

Questions

- How do I synchronize the python script with the executable?
- can I access variables from the executable?
- It seems the seed is generated once the user inputs the code. Am I suppose to run c code to get the value for the seed in the future so I then can go back and type in what I found to generate the code.
- Even when I try to rewrite in C code I am having the issue of getting the correct seed.
- Do I need to compile with the same compiler? Should I go back in and see how the rand() works.
-

Summary 5/13

- I am facing the issue of trying to figure out how to generate the password.
- I have found out that the seed is set with time.
- My issue is getting the correct rand() values from the C. I have tried using python and C to generate the code.
- I have my algorithm ready to reverse it is just a matter of getting the correct seed so I can generate constant correct inputs.