Category:

Reversing

Name:

Rock Paper Scissors

Message:

You are provided with an executable file named “RockPaperScissors.exe”. Once executed, it starts rock, paper, scissors game and asks your choice to compete against the computer. The flag for this challenge is displayed if you beat the computer. However, one win is not enough! You are required to win it more than 300 times to get the flag, which is so time consuming!! Can you think of a way to shorten this process?

Hints:

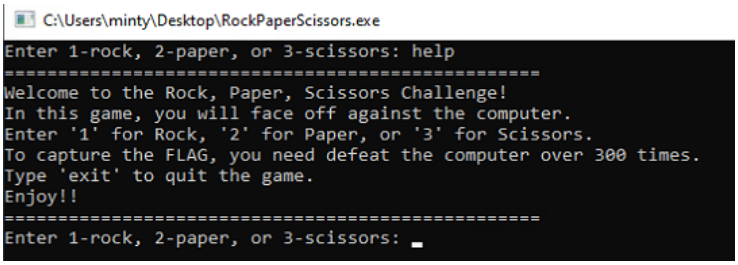
- Trying to win 300 times for the flag? Oh no, you’re thinking way too inside the box! It’s time to break the rules with some good old reverse engineering.
- With a little binary magic, you can tweak the flag reveal condition. Why wait for 300 wins when you can set it to something more manageable? Find that crucial function, patch program, and change byte to rewrite your destiny!

Objective:

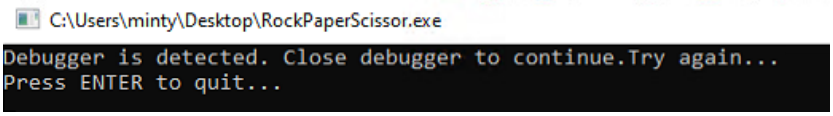
Your task is to reveal the flag from “RockPaperScissors.exe” either by winning the game more than 300 times legitimately or by modifying the binary to allow the flag to be revealed with fewer wins. This requires basic binary analysis and modification technique, especially the understanding of how software logic can be altered after the compilation at the machine code level.

Instructions:

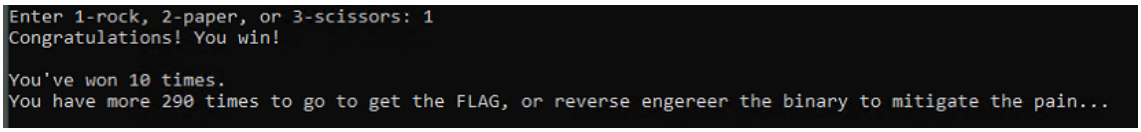
1. Download the zip file (“RockPaperScissors.zip”) and extract it to get the executable file “RockPaperScissors.exe”. No password is required to extract. Once executed, it asks for your choice for rock paper, scissors game. If necessary, enter the help command (help/-h/--h) to display additional information.



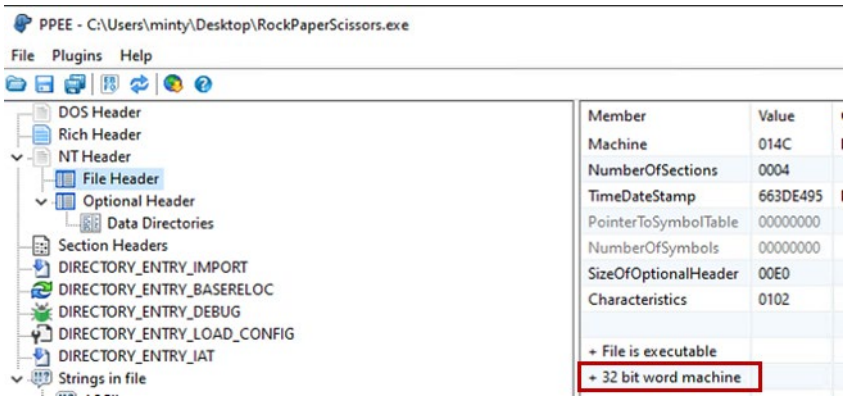
Run the program without using the debugger as it is programed to quit its operation if the debugger is detected.



2. If you enter either option (“1” for rock, “2” for paper or “3” for scissors), it returns the game result and continue asking you for the choice again. As you continue, it displays messages after winning 5, 10, 50, 100 and 200 times. The messages suggest performing reverse engineering to modify the binary instead of continuing the game straight forward.

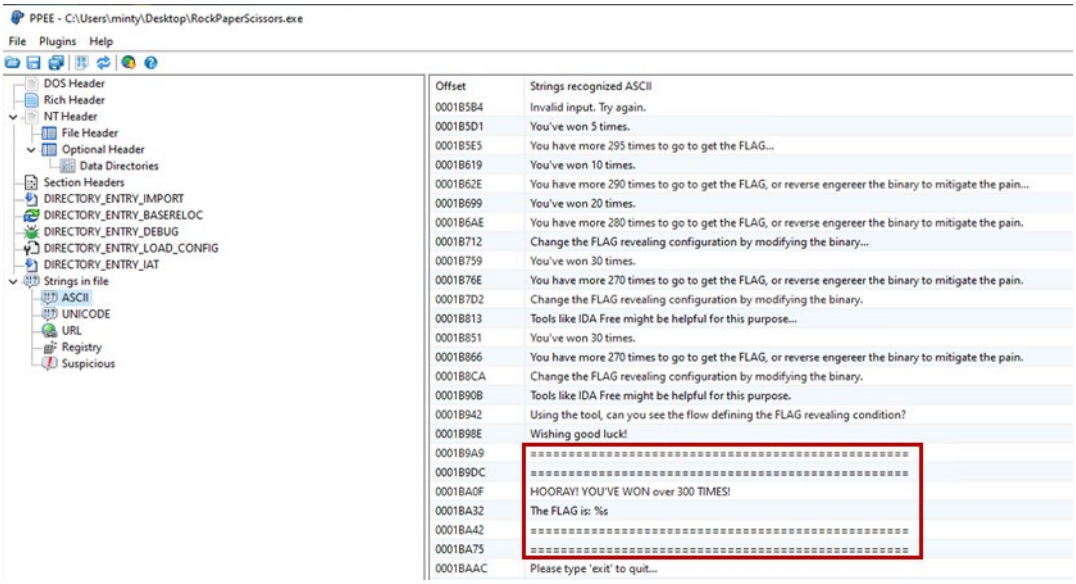


3. As it takes too much time to win 300 times, let’s try to modify the binary. First thing fist. Start from checking the file by using executable file analysis tool. The tool explained here is “PeStudio”, however any executable file analyzing tool should work as well. Open the file on PeStudio and note that the file is 32-bit word machine, indicating that it is intended for 32-bit environment.

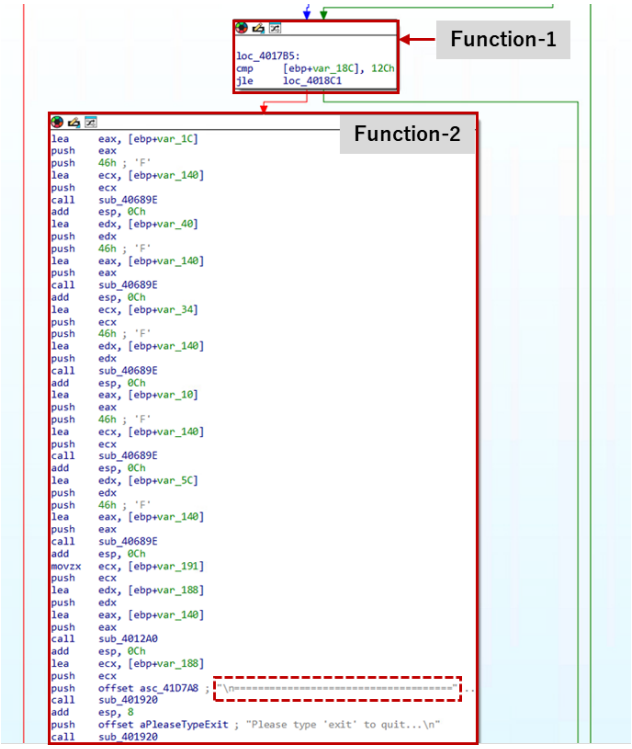


Then, looking into the list of strings, you may see phrases related to flag. For example, the image below shows the messages displayed after certain numbers of wins and that the flag is displayed with the phrase “==========
==========HOORAY!

YOU'VE WON 300 TIMES! The FLAG is:", which could be a lead during the reverse engineering.

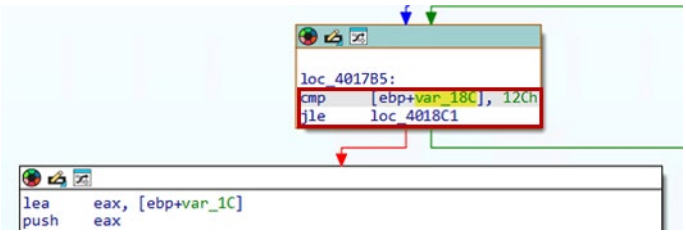


4. Now that you have the general overview on how the file works, let's use reverse engineering tool for further analysis. The tool explained here is "IDA Free", however any reversing engineering tool should work as well. In case if you use tools such as "x64dbg" which have different version for analyzing 64-bit files and 32-bit files, make sure that you launch the one for 32-bit files.
5. Open "RockPaperScissors.exe" in IDA Free. Look at the graph view and to search for the relevant code dealing with the flag revealing. Scroll down to find the functions highlighted below.

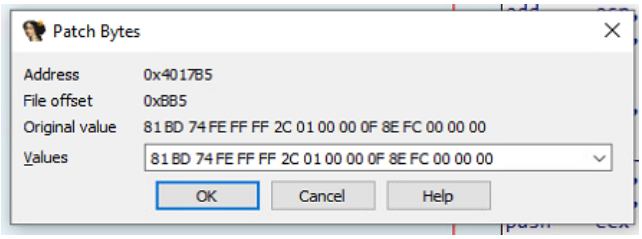


The Function-2 is used to display the flag, as you can see the first part of the phrase "==========HOORAY! YOU'VE WON 300 TIMES! The FLAG is:". This suggests that the function before Function-2, which is Function-1, defines the flag revealing condition. Function-1 compares the stored value in "ebp" with "300" (represented in hexadecimal as "12Ch") to check if the player won more than 300 times. Therefore, Function-1 is the exact code to be modified to get the flag with fewer wins. If you modify the comparison condition from "300" to a fewer number, the flag can be easily displayed. Let's target to decrease it to "1" instead of "300" using binary patching.

6. Select the instruction that compares player's win with "300" as shown below.

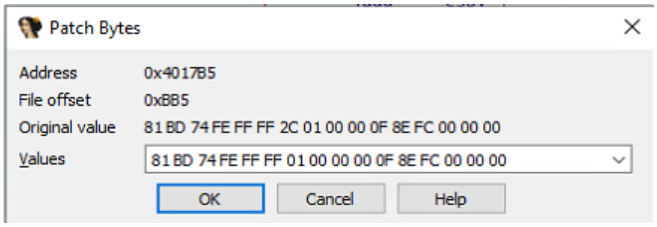


From the top bar, select **Edit > Patch program > Change byte**. The original bytes are shown as "81 BD 74 FE FF FF 2C 01 00 00 0F 8E FC 00 00 00", which is sequence of machine code instructions which are explained earlier.



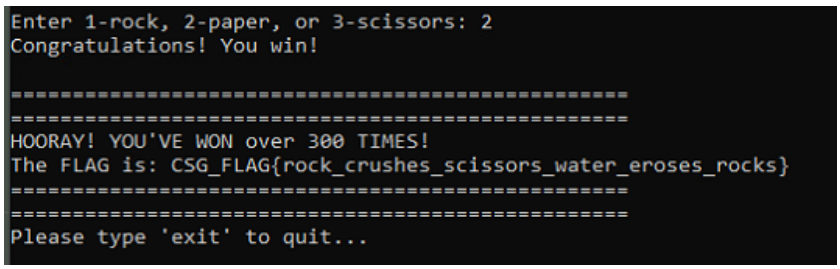
Using CyberCheff, the first part “81 BD 74 FE FF FF 2C 01 00 00” translates to “CMP DWORD PTR [RBP-0000018C],0000012C”, while “0F 8E FC 00 00 00” translates to “JLE 0000000000000102” in assembly instruction (Disassemble x86).

To change the condition so that it checks if the stored value in “ebp” is greater than “1” instead of “300”, modify “2C 01 00 00” (“300” in little-endian hexadecimal format) to “01 00 00 00” (“1” in little-endian hexadecimal format).



After editing, save the changes from top bar **Edit > Patch program > Apply patches to input file**.

7. Finally, run the modified “RockPaperScissors.exe” to ensure it changes the behaviors as expected. The flag should appear after winning more than 1 time.



Flag is:

CSG_FLAG{rock_crushes_scissors_water_erodes_rocks}

References:

- **Executable file analyzing tool**

PeStudio https://www.winitor.com/download

PPEE (Puppy) https://mzrst.com/

Detect It Easy (DIE) https://github.com/horsicq/Detect-It-Easy

- **Reverse engineering tool**

X64dbg (x32dbg) https://x64dbg.com/

IDA Free https://hex-rays.com/ida-free/

Ghidra https://ghidra-sre.org/

Rizin https://rizin.re/

Radare2 https://github.com/radareorg/radare2