### 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:

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.

```
C:\Users\minty\Desktop\RockPaperScissor.exe

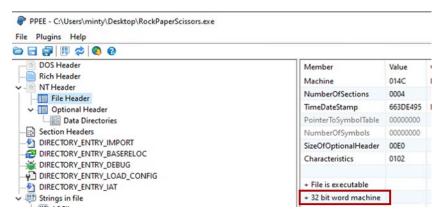
Debugger is detected. Close debugger to continue.Try again...

Press ENTER to quit...
```

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.

```
Enter 1-rock, 2-paper, or 3-scissors: 1
Congratulations! You win!
You've won 10 times.
You have more 290 times to go to get the FLAG, or reverse engereer the binary to mitigate the pain...
```

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

':	= :	= =	= :	=	=	= =	= :	=	=	= =	= :	=	:=	= =	=	=	: =	= :	=	=	: =	= :	=	=	= =	=	=	: =	= :	=	=	: =	=	=	: =	=	=	: =	= =	=	=	= =	=	=	: =	= :	=	=	: =	= :	=	=	=	=	: =	= =	= :	=	=	=	=	= =	= =	=	=	•							
=	=	=	=	: =	= :	=	=	=	= :	=	=	= =	= :	=	=	= =	=	=	:=	= =	=	=	:=	= :	=	=	= =	=	=	: =	= :	=	=	: :	=	=	: =	=	=	:=	= =	=	=	= =	=	=	: =	= :	=	=	=	=	= =	= :	=	=	=	=	: =	= =	=	=	=	=	ŀ	1(	Э	0	R	R/A	۱Y	<b>′</b> !	

#### PPEE - C:\Users\minty\Desktop\RockPaperScissors.exe File Plugins Help DOS Header Rich Header Offset Strings recognized ASCII 0001B5B4 Invalid input. Try again NT Header File Header Optional Header Data Directories 0001B5E5 You have more 295 times to go to get the FLAG... You've won 10 times. Section Headers 0001R62E You have more 290 times to go to get the FLAG, or reverse engereer the binary to mitigate the pain... DIRECTORY\_ENTRY\_IMPORT DIRECTORY\_ENTRY\_BASERELOC DIRECTORY\_ENTRY\_DEBUG DIRECTORY\_ENTRY\_LOAD\_CONFIG 0001B699 You've won 20 times. 0001B6AE You have more 280 times to go to get the FLAG, or reverse engereer the binary to mitigate the pain 0001B712 Change the FLAG revealing configuration by modifying the binary... DIRECTORY\_ENTRY\_IAT Strings in file Strings in file Strings in file UNICODE UNICODE 0001B76E You have more 270 times to go to get the FLAG, or reverse engereer the binary to mitigate the pain. Change the FLAG revealing configuration by modifying the binary. 0001B7D2 0001B813 Tools like IDA Free might be helpful for this purpose... 0001B851 You've won 30 times Registry Suspicious 0001B866 You have more 270 times to go to get the FLAG, or reverse engereer the binary to mitigate the pain 0001B8CA Change the FLAG revealing configuration by modifying the binary. 0001B90B Tools like IDA Free might be helpful for this purpose 0001B942 Using the tool, can you see the flow defining the FLAG revealing condition? 0001B98E Wishing good luck! 0001B9A9 0001B9DC ..... 0001RA0F HOORAY! YOU'VE WON over 300 TIMES! 0001BA32 The FLAG is: %s 0001BA42

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

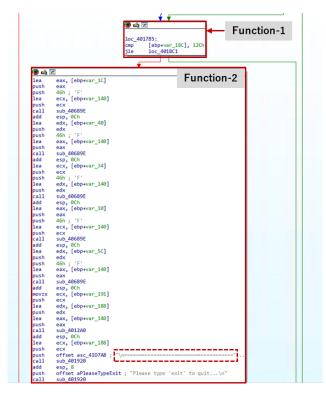
4. Now that you have the general overview on how the file works, let's use reverse engineering tool for further analysis. The tool explained here is "IDA Free", however any reversing engineering tool should work as well. In case if you use tools such as "x64dbg" which have different version for analyzing 64-bit files and 32-bit files, make sure that you launch the one for 32-bit files.

------

0001BA75

0001BAAC

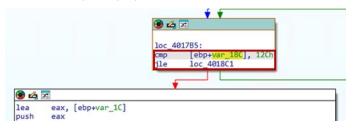
5. Open "RockPaperScissors.exe" in IDA Free. Look at the graph view and to search for the relevant code dealing with the flag revealing. Scroll down to find the functions highlighted below.



==============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.

```
Enter 1-rock, 2-paper, or 3-scissors: 2

Congratulations! You win!

HOORAY! YOU'VE WON over 300 TIMES!
The FLAG is: CSG_FLAG{rock_crushes_scissors_water_eroses_rocks}

Please type 'exit' to quit...
```

Flag is:
CSG FLAG{rock crushes scissors water erodes rocks}

### References:

# Executable file analyzing tool

PeStudio <a href="https://www.winitor.com/download">https://www.winitor.com/download</a>

PPEE (Puppy) <a href="https://mzrst.com/">https://mzrst.com/</a>

Detect It Easy (DIE) <a href="https://github.com/horsicq/Detect-It-Easy">https://github.com/horsicq/Detect-It-Easy</a>

# Reverse engineering tool

X64dbg (x32dbg) <a href="https://x64dbg.com/">https://x64dbg.com/</a>

IDA Free <a href="https://hex-rays.com/ida-free/">https://hex-rays.com/ida-free/</a>

Ghidra <a href="https://ghidra-sre.org/">https://ghidra-sre.org/</a>

Rizin <a href="https://rizin.re/">https://rizin.re/</a>

Radare2 <a href="https://github.com/radareorg/radare2">https://github.com/radareorg/radare2</a>