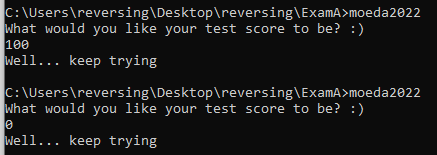
**MoedA2022 writeup**

Author: Shoham Zeharya

To get an idea what we should expect to look for in the dissassembly, i run the program in cmd and this is the output i got:

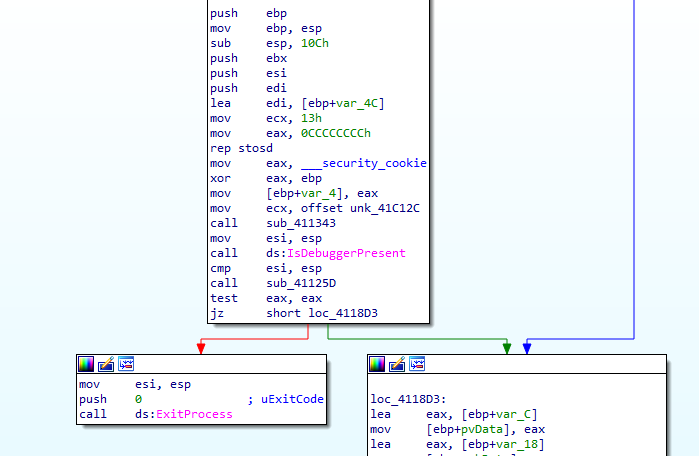


So i assume the input is not gonna be anything we can think about.

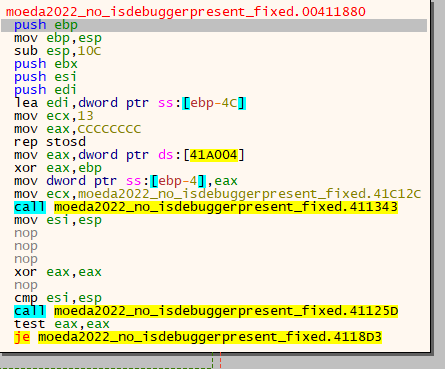
Before opening any program, I’ll open the executable in cff explorer, go to **Optional Header** > **DllCharacteristics** > And disable the option "**DLL can move**". This will cause the executable to load at the preset ImageBase, even in the case of dynamic debugging.

No use running the program in a debugger without disabling peb+2, It will just kick us out.

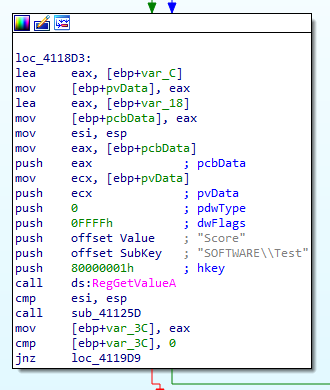
To hide the debbuger, in the menu up top in x32 i open **Debug**, then **advanced options**, then **hide debugger.**



In the entry point, something quite strange happened. In x32 the entry point was at 41105a and in ida it was at the address cff explorer said it would. What might have happened is that the program changed the entry point on the fly. Anyway in x32 The is a call for a function the gets the current thread id, and right after there is a call to is debugger present. Lets patch over this call, and insert xor eax,eax command.

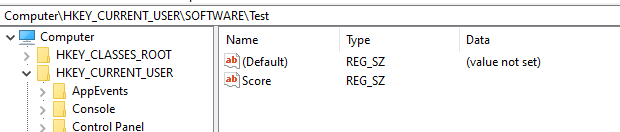


However, when i press run in x32, the preogram still crashes. A little after the IsDebuggerPresent call there is a call for RefGetValueA:

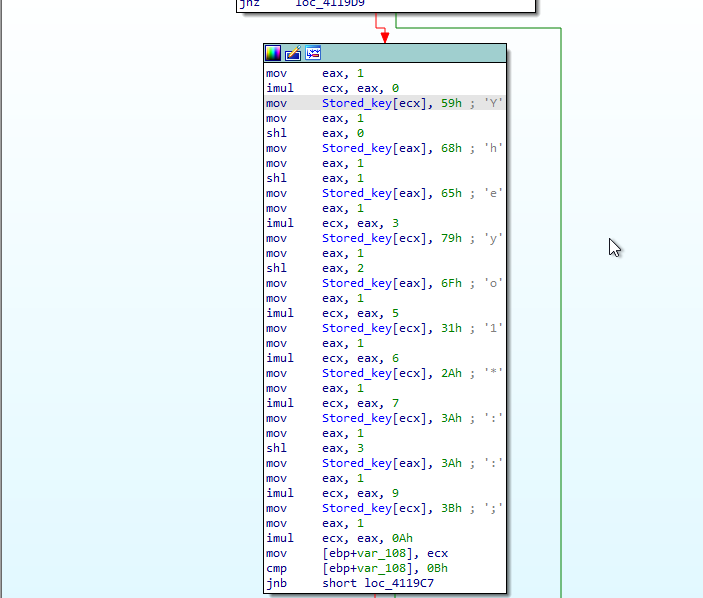


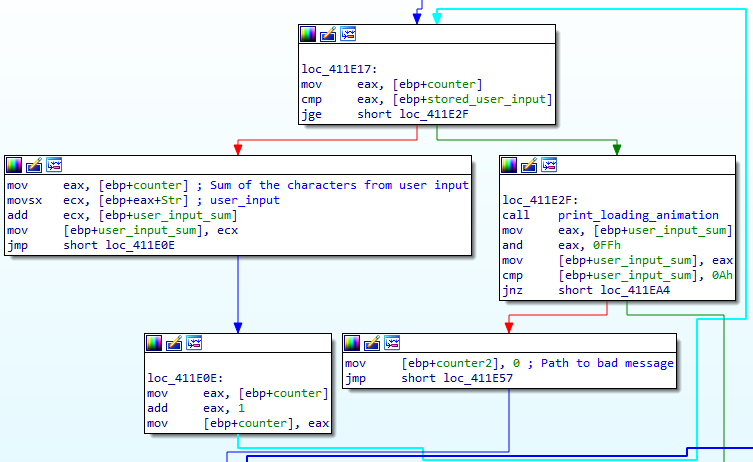
So what i might have to do is create a registery value at SOFTWARE\Test names Score,

and the value doesn’t have to be anything restricted. I assume the value needs to be in "HKEY\_CURRENT\_USER\SOFTWARE\Test", and turns out it was correct, since now I am able to advance more.



Right after the registery check, there is loading of the string "Yheyo1\*::;" to memory, this is the encrypted password:

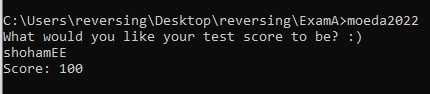


Now to the main part of the program. After the program asks the user for input and stores it, some interesting stuff are happening. First (1) the sum of the characters is calculated, then, if the sum modulu 0xAA is not equal 0xA (2) the user gets the bad message.

1

2

In case the sum was right – for example "shohamEE" was given as input, the program decrypts the string "Yheyo1\*::;" to the string "Score: 100".



snippet of the code in main – in test.py.