Stack and Stack Frame

CSC 471 - 02

Tyler Prehl

3/11/2022

Introduction

The purpose of this lab is to better understand the stack and stack frame. Using my knowledge of IA32 CPU register and X86 ASM basics, I will be able to use OllyDbg to modify a binary file and prevent a “nagging” piece of malware.

Analysis and Results

1) Which CPU register is used to store the return value (1) of the function rtcMsgBox()? Why?

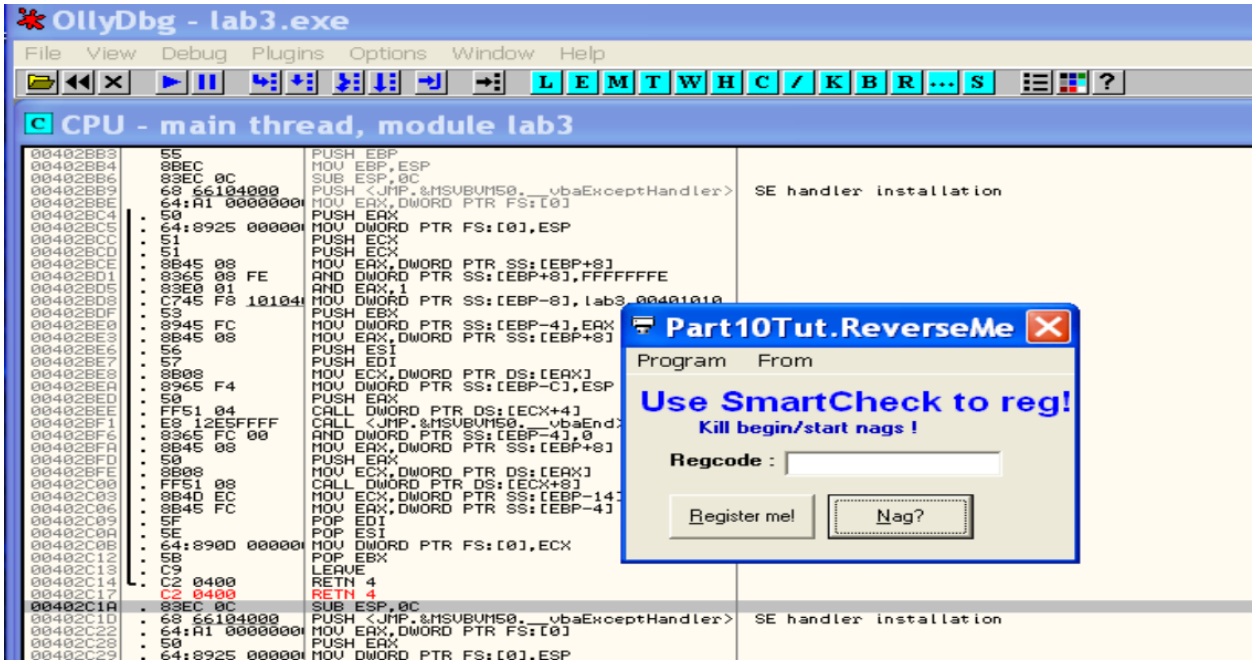
The EAX CPU register is used to store the return value (1) of the function rtcMsgBox(). This is because the purpose of the EAX register is to store and return values from the stack.

2) What’s the meaning of ”PUSH EBP, MOV EBP, ESP”?

Before I answer the question, it is important to note that the EBP register is the base stack pointer register and ESP is the current stack pointer. “PUSH EBP, MOV EBP, ESP” represents the start of a new stack frame. PUSH EBP pushes whatever is currently in the EBP register onto the top of the stack so that the program can access data that was pushed onto the stack in reference to the EBP register, and then MOV EBP, ESP moves the ESP value (current stack pointer) into EBP to save the base of the new stack frame to the EBP register.

3) Please explain why changing the instruction on 0x402C17 from ”PUSH EBP” to ”RETN 4” removes the Nag screen.

0x402C17 is the beginning of the nag routine, so instead of calling the nag routing, RETN 4 simply returns the value that ends the process and removes the most recent frame, preventing any nags from occurring.



Discussion and Conclusion

After seeing the successful completion of this lab through, I now understand how to easily manipulate assembly code in OllyDbg to change functions to prevent certain stack frames from executing, thereby preventing a nagging action from occurring. In the future, this will be helpful for preventing specific functions from running by modifying the executable file of any given malware.