**CS4238 Lab: Debugging with IDA**

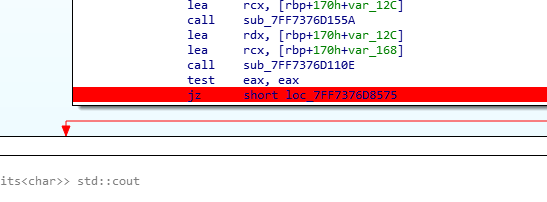
The **goal** of this lab is to get familiar with **debugging with IDA** and **program patching.**

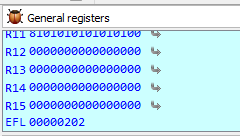
**Lab Set-up**

You will need the IDA Free for this lab (already integrated into the Flare VM). A *secret.exe* file will be used for analysis (see this lab’s attachments).   
It is a program which takes a string as input, compares the input with a secret string (after decryption), and decides whether to accept the input or reject it. In this lab, we will show three ways to pass such a password check, the first two through debugging and the third through program patching.

**A. Debugging**

The goal of this task is to (1) find the password in plain text and (2) bypass the password checking by flip one bit in *eflag*.

1. Instruction to set a breakpoint: Add breakpoint (F2) and run (F9); or run to cursor (F4), which will run the program and break at the current cursor location.
2. In *secret.exe*, set a breakpoint at *the address shown in Jnz instruction (after string compare)*. Can you read the values inside the char array?
3. Check values of variables in the current state by moving the cursor to the variable.
4. Find the registers in the right. “EFL” here stands for eflag register. Can you modify its value to change the following control flow? (Click the register, and press “E”)



**B. Patching**

The goal of this task is to bypass the password checking by patching the program permanently.

1. Select the *jnz* instruction, and switch to hex view.
2. Use an online [*assembler&disassembler*](http://shell-storm.org/online/Online-Assembler-and-Disassembler/) to get the raw bytes of *jnz* instruction with the same offset.
3. Go In Edit->Patch Program->Change Bytes…

Replace the raw bytes of *jnz* instruction with those of *jz* instruction

1. Apply your changes in Edit->Patch Program->Apply patches to input file…
2. Try to run the cracked program now!