

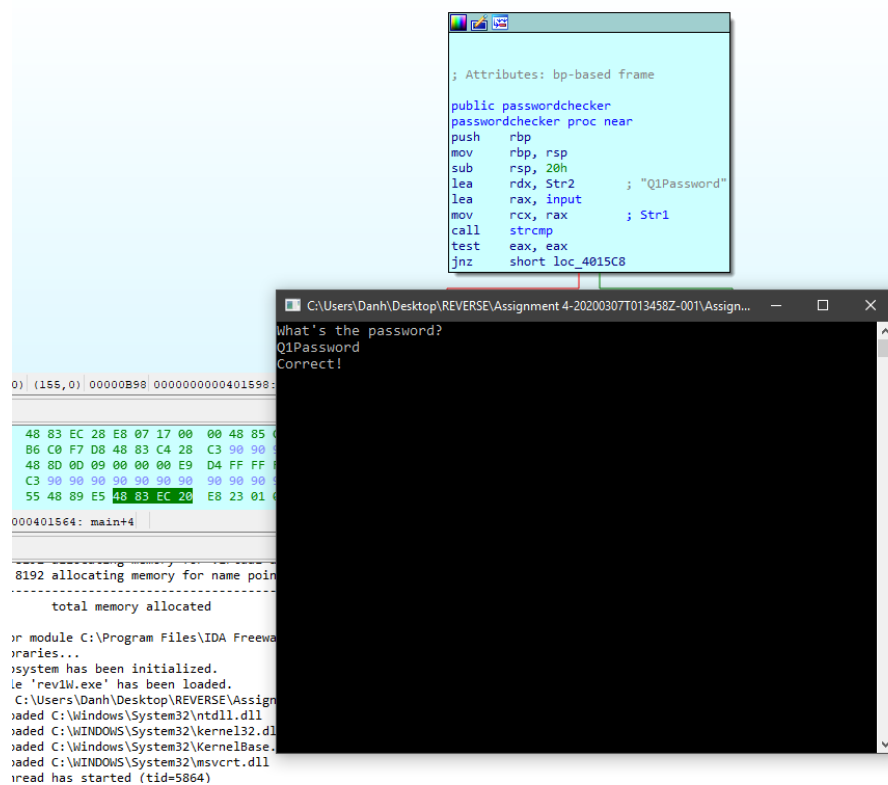
Reverse Engineering

Spring-2020 Semester

Danh Pham

Task1:

Q1Password



Task2:

Hello

The screenshot displays the IDA Pro interface for a function named `passwordcheck`. The function's assembly code is shown in the main window, with callouts highlighting specific instructions:

```

; Attributes: bp-based frame
public passwordcheck
passwordcheck proc near
push    rbp
mov     rbp, rsp
sub     rsp, 20h
lea     rax, input
movzx   eax, byte ptr [rax+1]
cmp     al, 65h
jnz     short loc_40163C

lea     rax, input
movzx   eax, byte ptr [rax+2]
cmp     al, 6Ch
jnz     short loc_40162E

```

The Hex View window shows the memory layout of the function, with the password string "Hello" stored in memory. The Output window shows the program's execution, displaying the prompt "Whats the password?", the input "Hello", and the response "Correct!".

Task3:

849

The screenshot displays a debugger interface with three main panels:

- Assembly View:** Shows a block of assembly code. The instruction `mov eax, cs:input` is highlighted in red. Below it, a conditional jump `jnz short loc_4015F9` is shown. Further down, two paths are visible: one leading to `lea rcx, aCorrect ; "Correct!"` and `call printf`, and another leading to `loc_4015F9: lea rcx, aWrong` and `call printf`. A third block shows `loc_401605: nop`, `add rsp, 20h`, `pop rbp`, and `retn`.
- General registers:** Lists registers RAX, RBX, RCX, RDX, and RSI with their current values. RAX is 0000000000000000, RBX is 0000000000000000, RCX is 00000000FFFFFFFF, RDX is 000077FD0E97FA30, and RSI is 0000000000000057.
- Console Window:** Shows the output of the program. It displays the prompt "What number am I thinking of?" followed by the input "849" and the response "Correct!".

At the bottom, a memory dump shows hex values and their ASCII representation: `20 E8 53 01 00 00 48 00 00 UH&Hfi·&5...H...`, `00 00 48 8D 15 80 5A 00 00 @*..è...H..*Z...`, `E8 74 16 00 00 E8 00 00 00 H..*...èt...è...`, and `83 C4 20 5D C3 55 48 09 E5HFÄ·lÄUH&ä`.

Task4:

change jz to jnz

