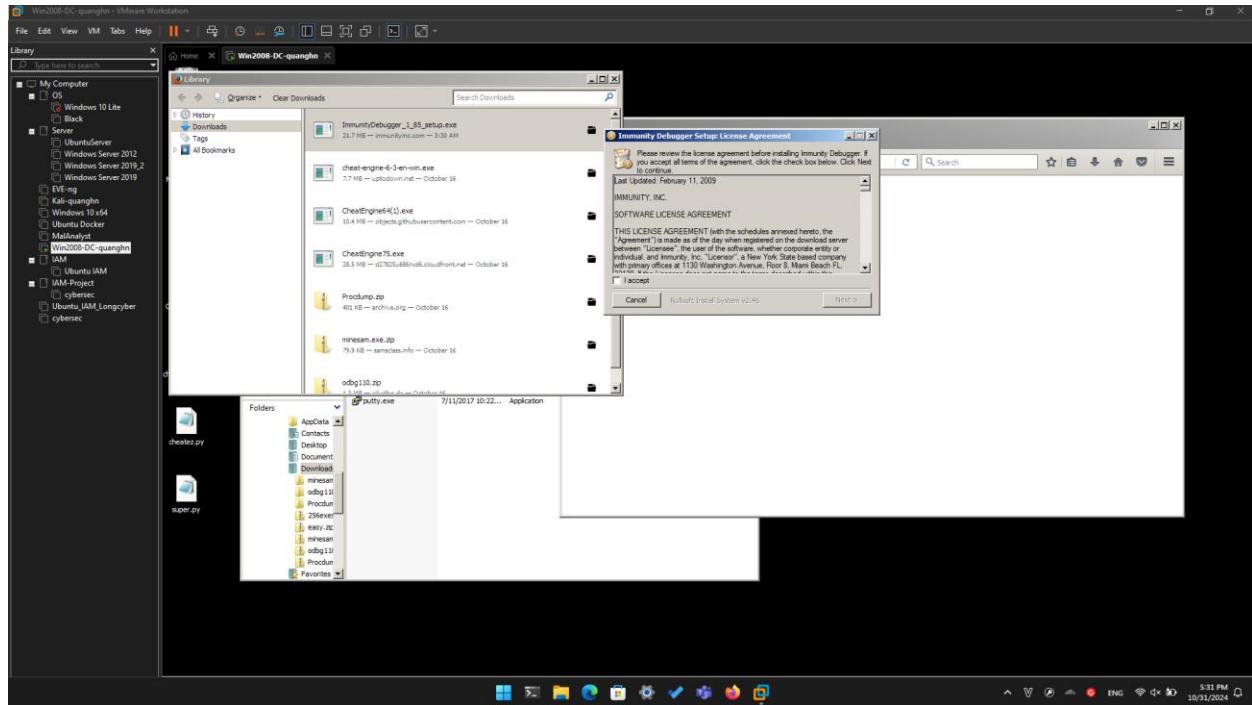


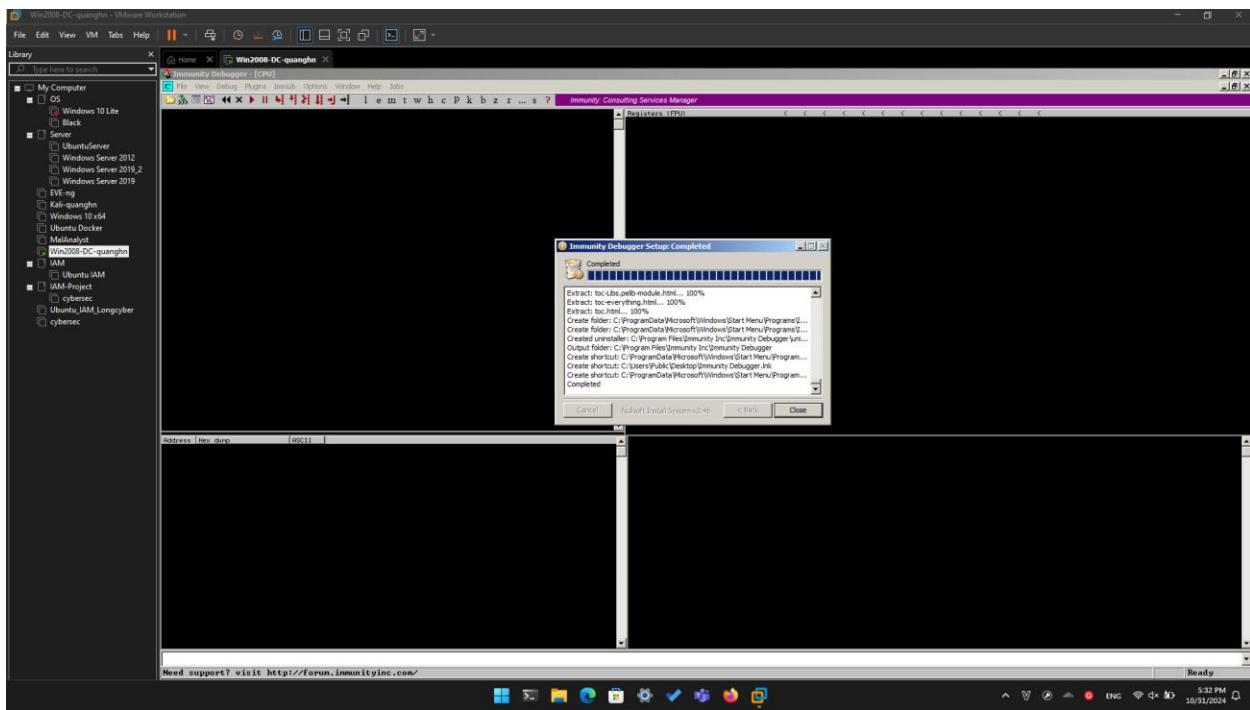
# Lab 9: Simple EXE Hacking with Immunity

Huynh Ngoc Quang – SE181838

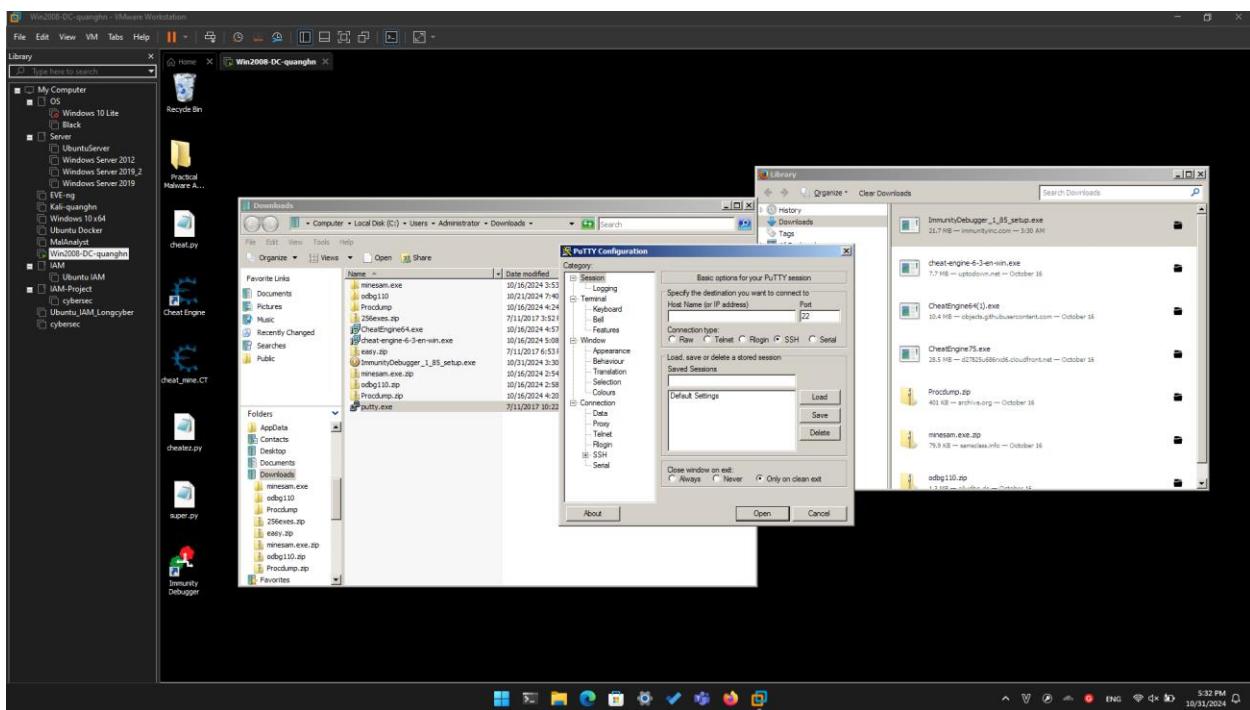
## Task 1: Target EXE Recon

### Get Immunity

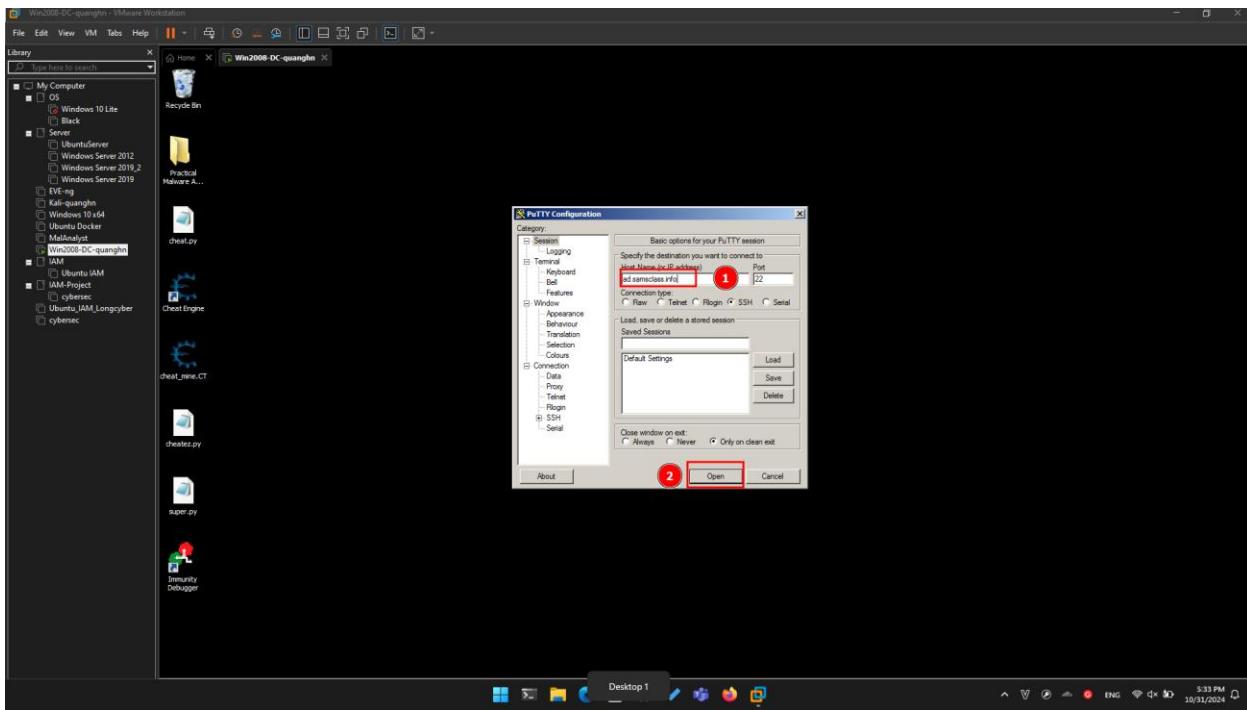
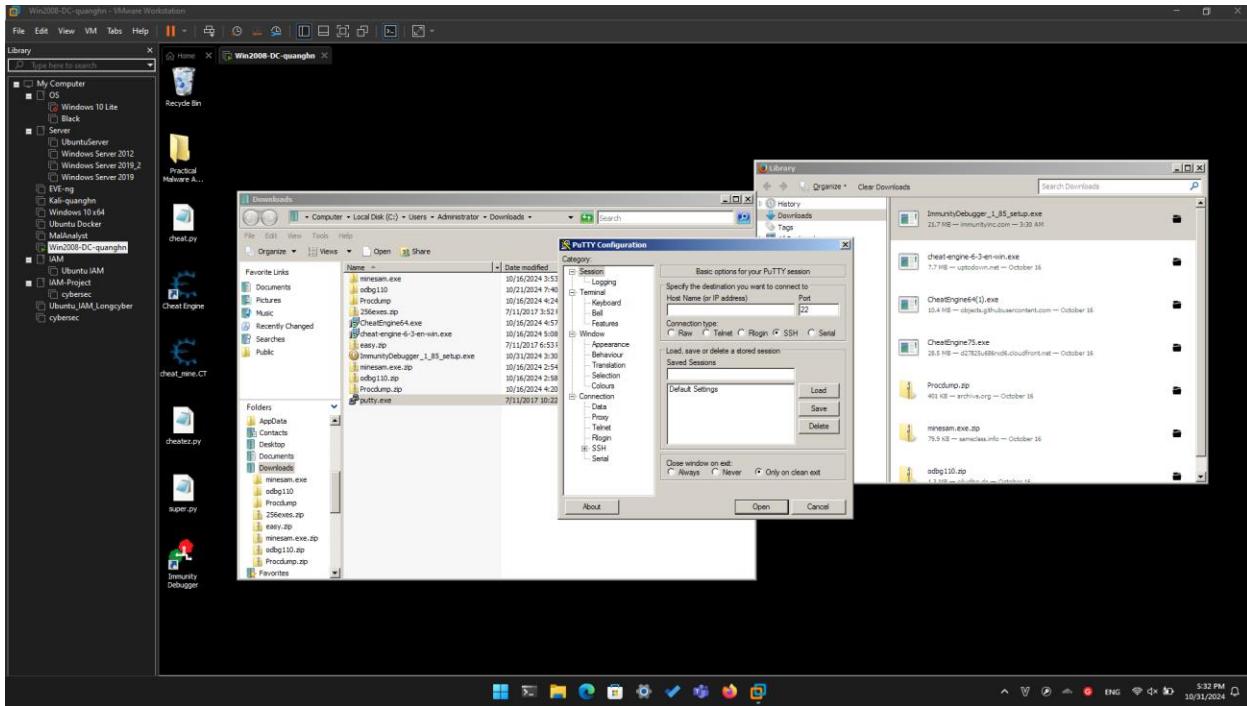


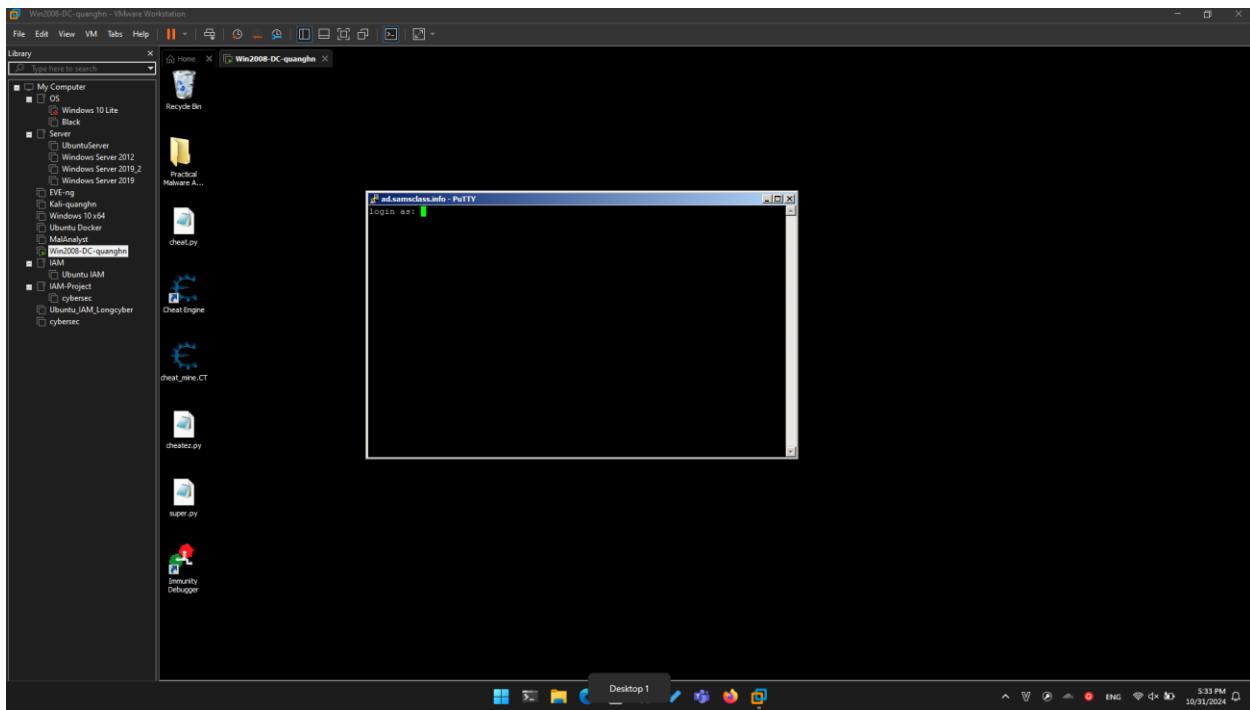


## Get putty.exe

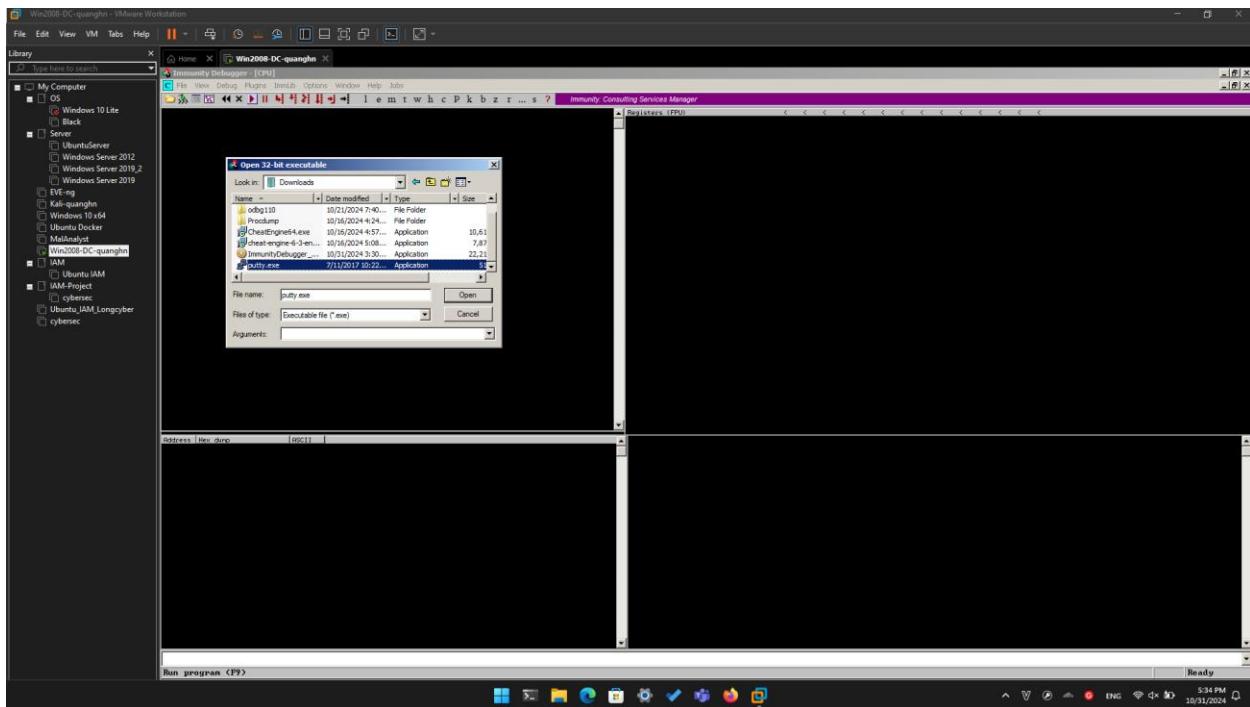


## Running Putty





## Starting the Immunity Debugger



```

00455100 6A 48 8B 14 PUSH EDI
00455104 6B 00 00 00 ADD EDI,00000000
00455108 48 8D 3C 00 LEA EDI,[EDI+00000000]
0045510C 8B C7 MOV EBX,EAX
0045510E 8B D9 MOV ECX,EBX
00455110 8B F0 MOV ECX,ECX
00455112 8B C7 MOV EBX,EAX
00455114 8B D9 MOV ECX,EBX
00455116 8B F0 MOV ECX,ECX
00455118 8B C7 MOV EBX,EAX
0045511A 8B D9 MOV ECX,EBX
0045511C 8B F0 MOV ECX,ECX
0045511E 8B C7 MOV EBX,EAX
00455120 8B D9 MOV ECX,EBX
00455122 8B F0 MOV ECX,ECX
00455124 8B C7 MOV EBX,EAX
00455126 8B D9 MOV ECX,EBX
00455128 8B F0 MOV ECX,ECX
0045512A 8B C7 MOV EBX,EAX
0045512C 8B D9 MOV ECX,EBX
0045512E 8B F0 MOV ECX,ECX
00455130 8B C7 MOV EBX,EAX
00455132 8B D9 MOV ECX,EBX
00455134 8B F0 MOV ECX,ECX
00455136 8B C7 MOV EBX,EAX
00455138 8B D9 MOV ECX,EBX
0045513A 8B F0 MOV ECX,ECX
0045513C 8B C7 MOV EBX,EAX
0045513E 8B D9 MOV ECX,EBX
00455140 8B F0 MOV ECX,ECX

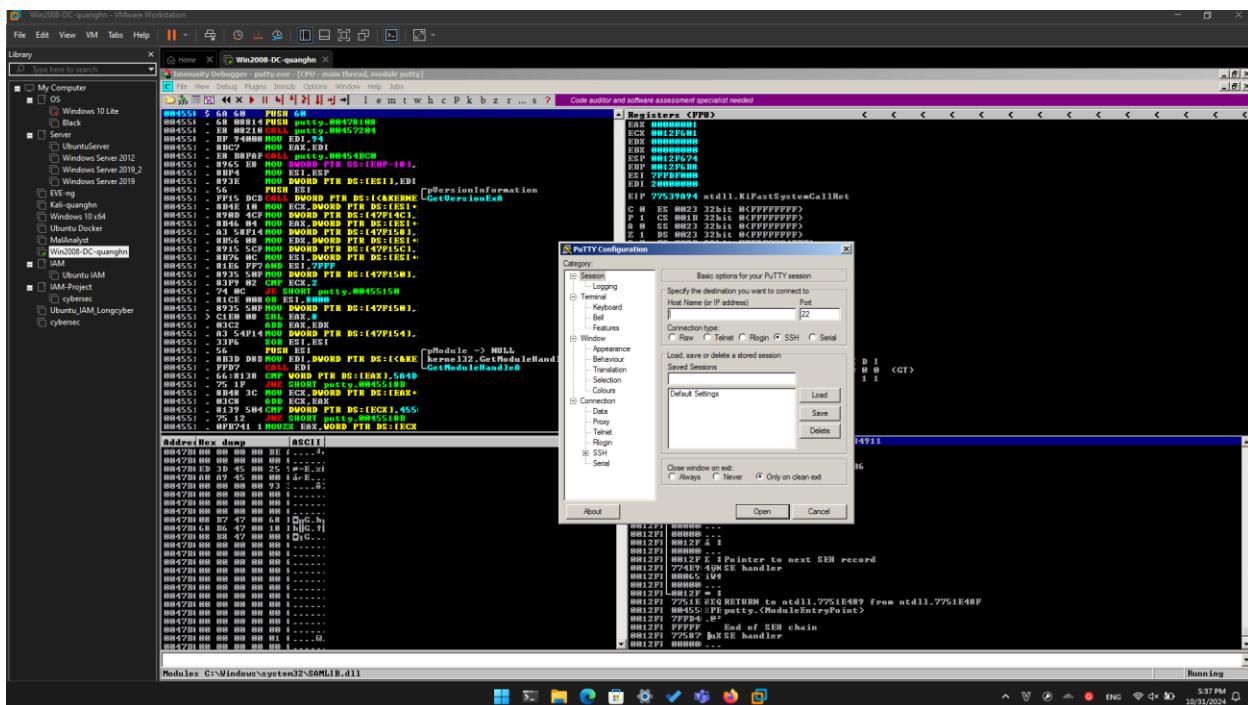
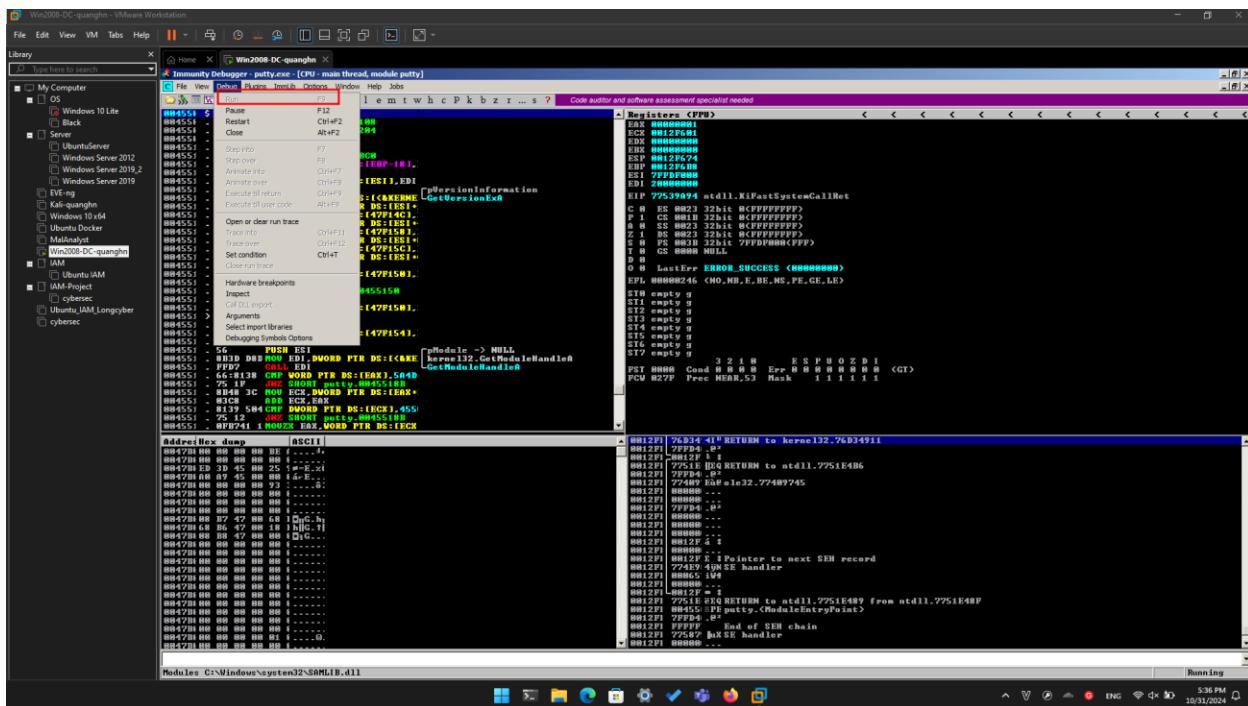
```

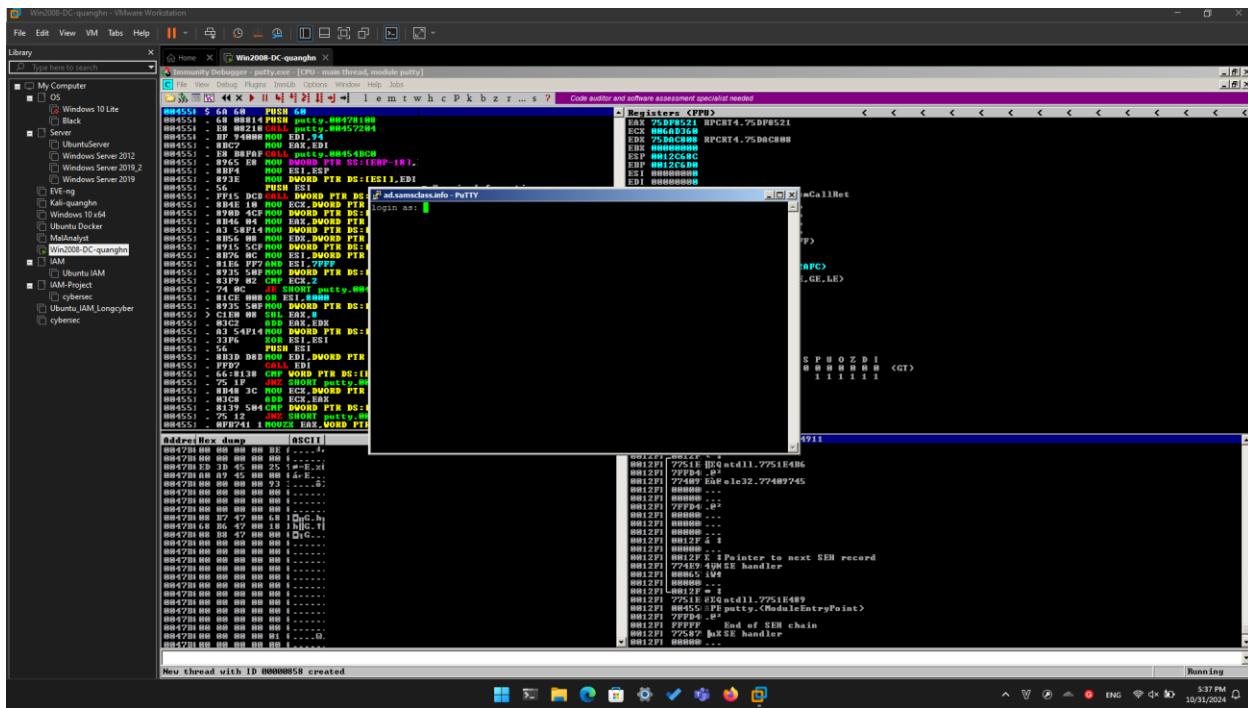
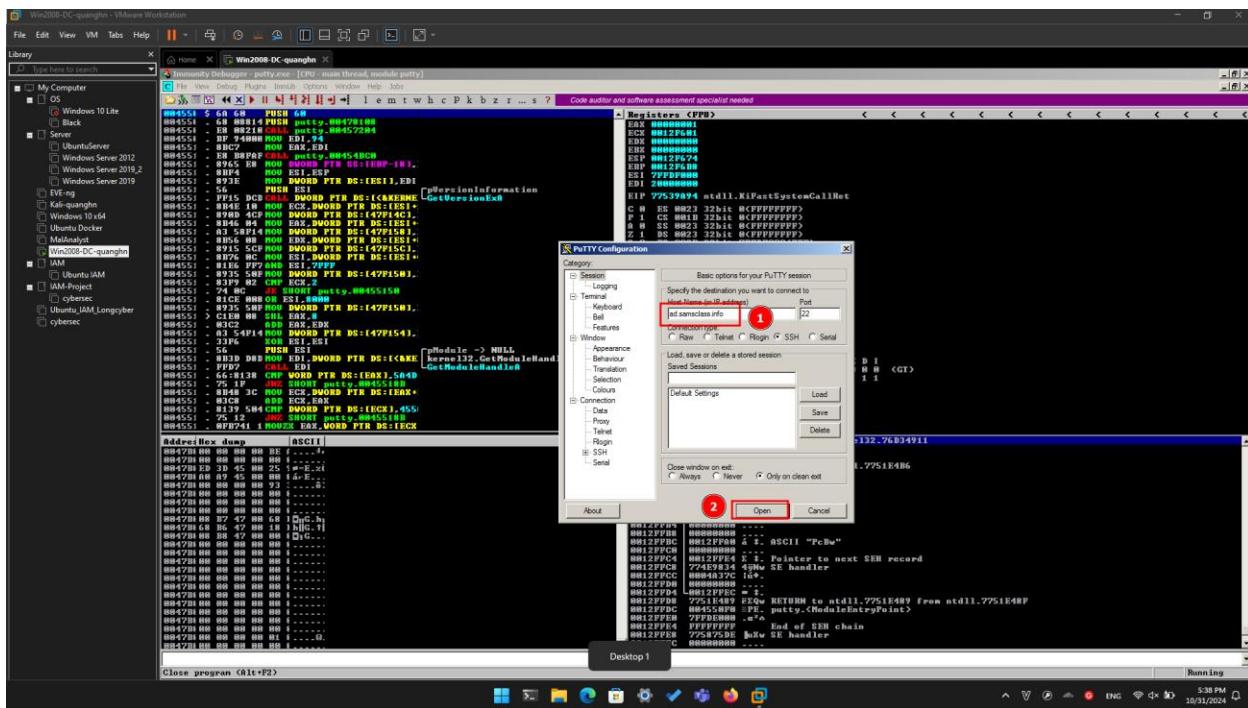
```

00455100 6A 48 8B 14 PUSH EDI
00455104 6B 00 00 00 ADD EDI,00000000
00455108 48 8D 3C 00 LEA EDI,[EDI+00000000]
0045510C 8B C7 MOV EBX,EAX
0045510E 8B D9 MOV ECX,EBX
00455110 8B F0 MOV ECX,ECX
00455112 8B C7 MOV EBX,EAX
00455114 8B D9 MOV ECX,EBX
00455116 8B F0 MOV ECX,ECX
00455118 8B C7 MOV EBX,EAX
0045511A 8B D9 MOV ECX,EBX
0045511C 8B F0 MOV ECX,ECX
0045511E 8B C7 MOV EBX,EAX
00455120 8B D9 MOV ECX,EBX
00455122 8B F0 MOV ECX,ECX
00455124 8B C7 MOV EBX,EAX
00455126 8B D9 MOV ECX,EBX
00455128 8B F0 MOV ECX,ECX
0045512A 8B C7 MOV EBX,EAX
0045512C 8B D9 MOV ECX,EBX
0045512E 8B F0 MOV ECX,ECX
00455130 8B C7 MOV EBX,EAX
00455132 8B D9 MOV ECX,EBX
00455134 8B F0 MOV ECX,ECX
00455136 8B C7 MOV EBX,EAX
00455138 8B D9 MOV ECX,EBX
0045513A 8B F0 MOV ECX,ECX
0045513C 8B C7 MOV EBX,EAX
0045513E 8B D9 MOV ECX,EBX
00455140 8B F0 MOV ECX,ECX

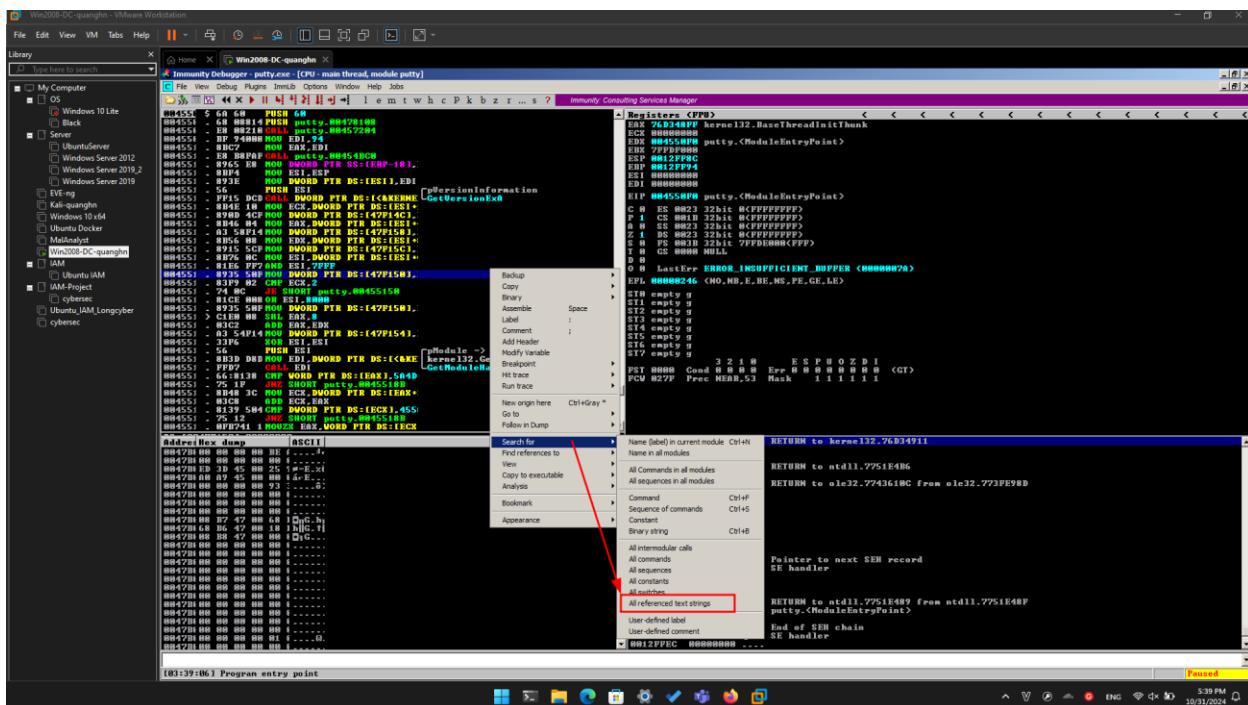
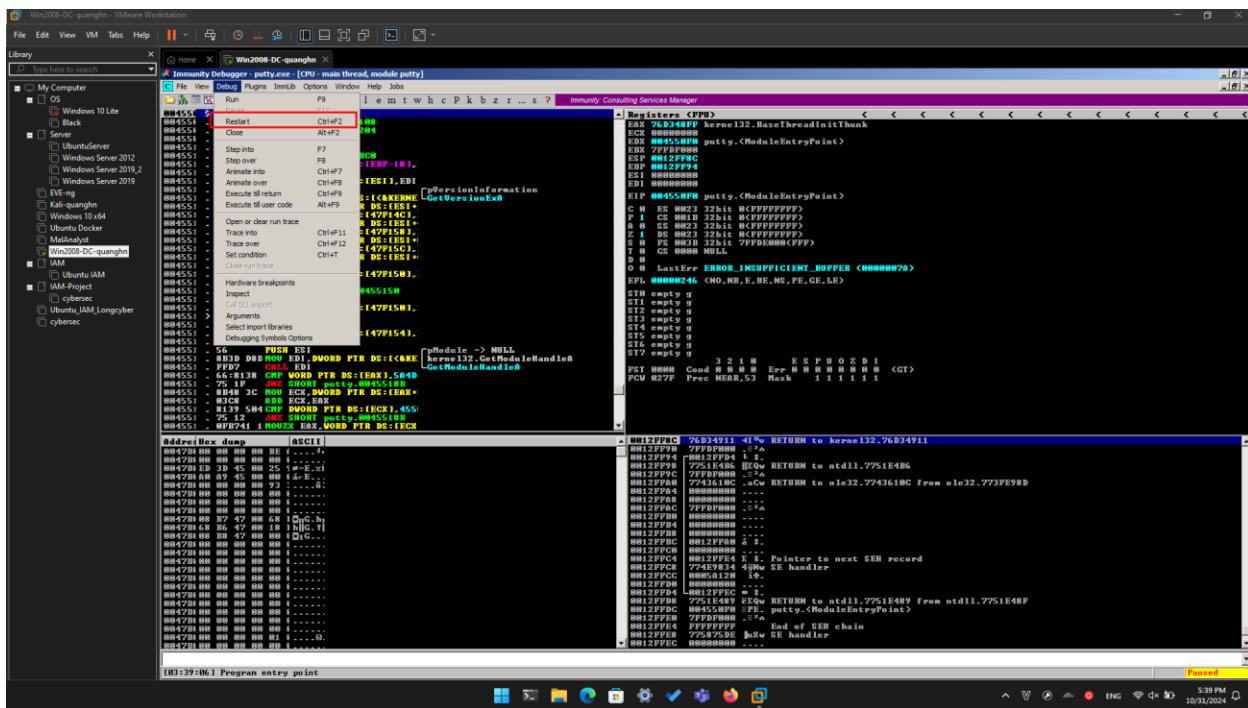
```

## Running Putty in Immunity

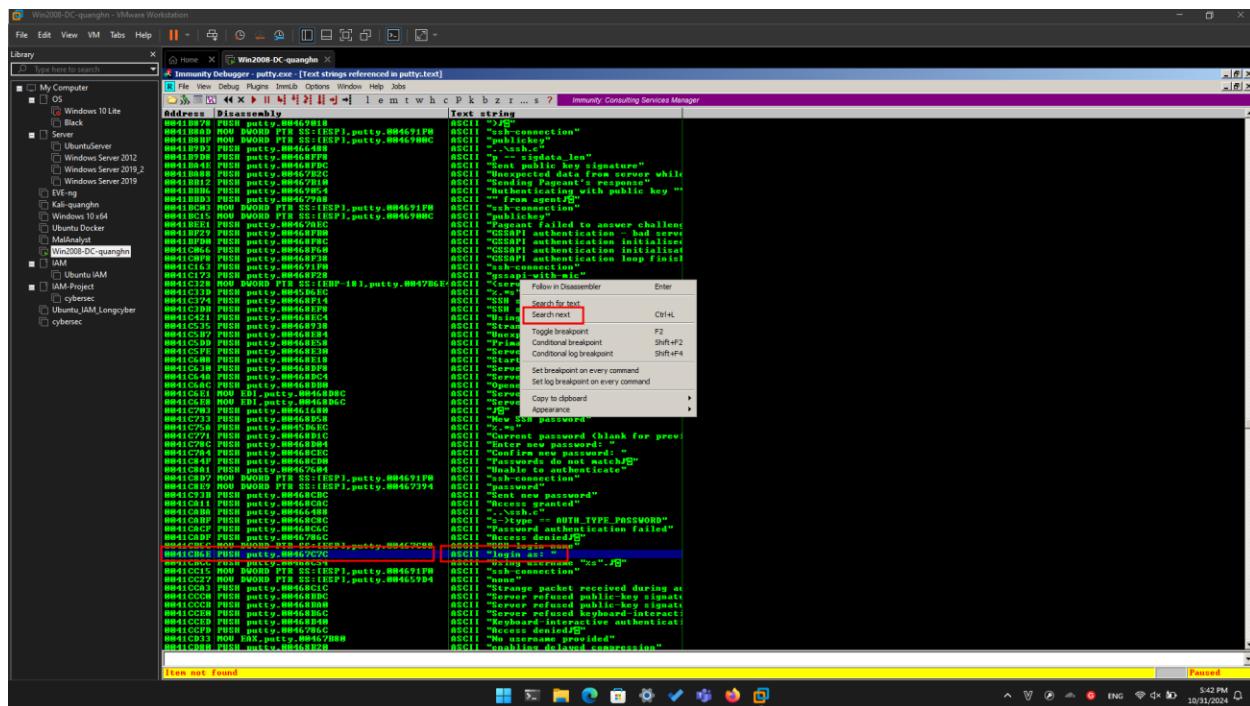




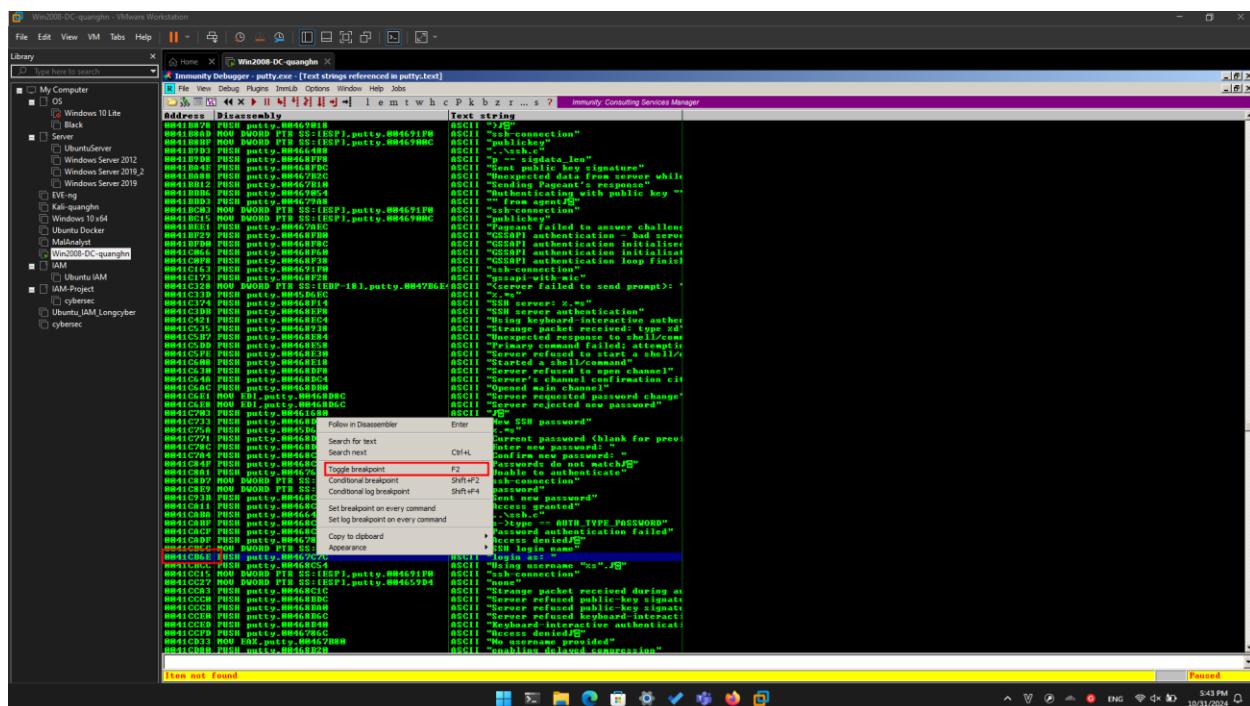
## Finding the "login as" Code



The screenshot shows the Immunity Debugger interface with the file 'putty.exe' loaded. The assembly window displays assembly code, and the text strings window shows various ASCII strings. A context menu is open over the text strings, with options like 'Search for text', 'Format for a', 'Toggle breakpoint', 'Conditional breakpoint', 'Conditional log breakpoint', 'Set breakpoint on every command', 'Set log breakpoint on every command', 'Copy to clipboard', and 'Find in file "putty.exe"'. The status bar at the bottom indicates 'Paused'.



## Using Breakpoints



In Immunity, from the menu bar, click **Debug**, **Restart**.

A box pops up warning you that "Process 'putty' is active". Click **Yes**.

In Immunity, from the menu bar, click **Debug**, **Run**.

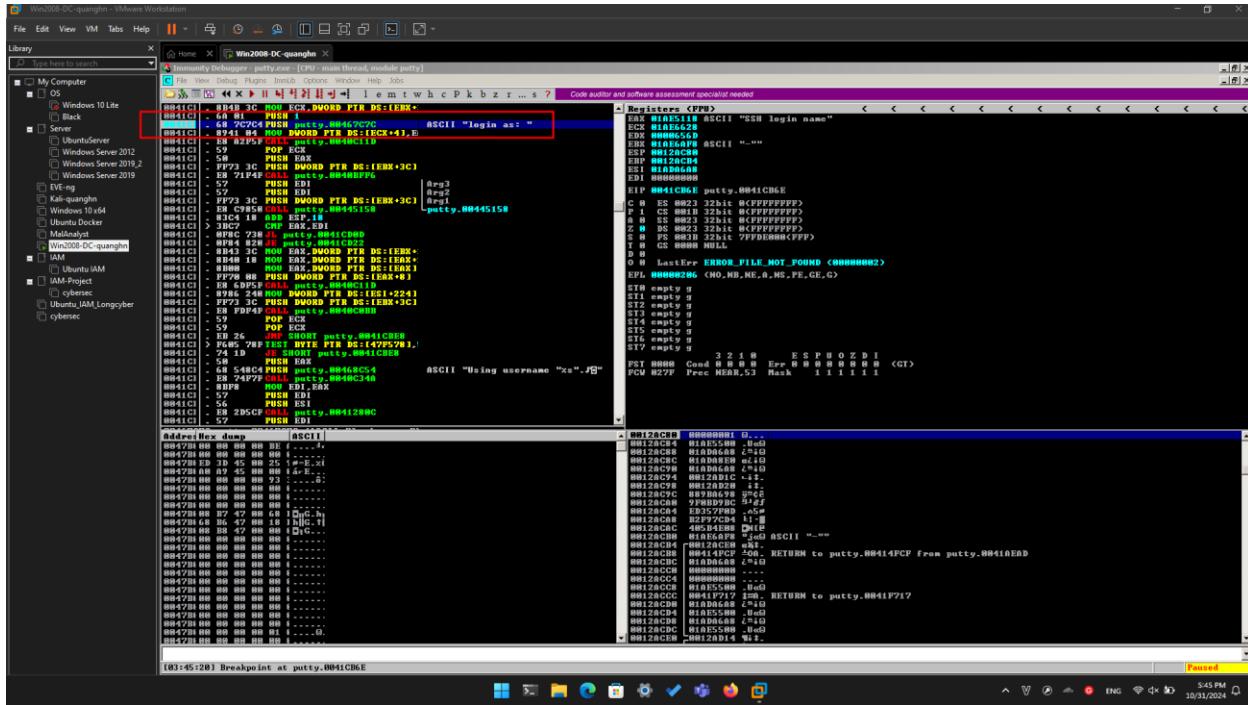
A Putty window opens, as shown below.

The screenshot shows a debugger session on Win2008-DC-quangnh. The assembly window displays code from `putty.exe`, specifically the `main thread, module putty`. The assembly code includes instructions like `PUSH ECX`, `CALL QWORD PTR DS:[EBP+00000000]`, and `MOV ECX, EDI`. A tooltip for `GetVersionExA` is visible. The memory dump window shows the address `0012F9B8` containing the value `00000000`. The Registers window shows CPU registers with values such as `ECX 0012F9B8`, `EDX 00000000`, and `ESP 00000000`. The Stack window shows the stack contents starting with `00000000`. A `Putty Configuration` dialog box is overlaid on the interface, showing session details for a connection to `132.76.34.911` on port `22`.

Click in the Putty window. In the "Host Name (or IP address)" box, type  
**ad.samsclass.info**

At the bottom, click the **Open** button.

A black window opens, but before the "login as" message appears, the program stops, as shown below.

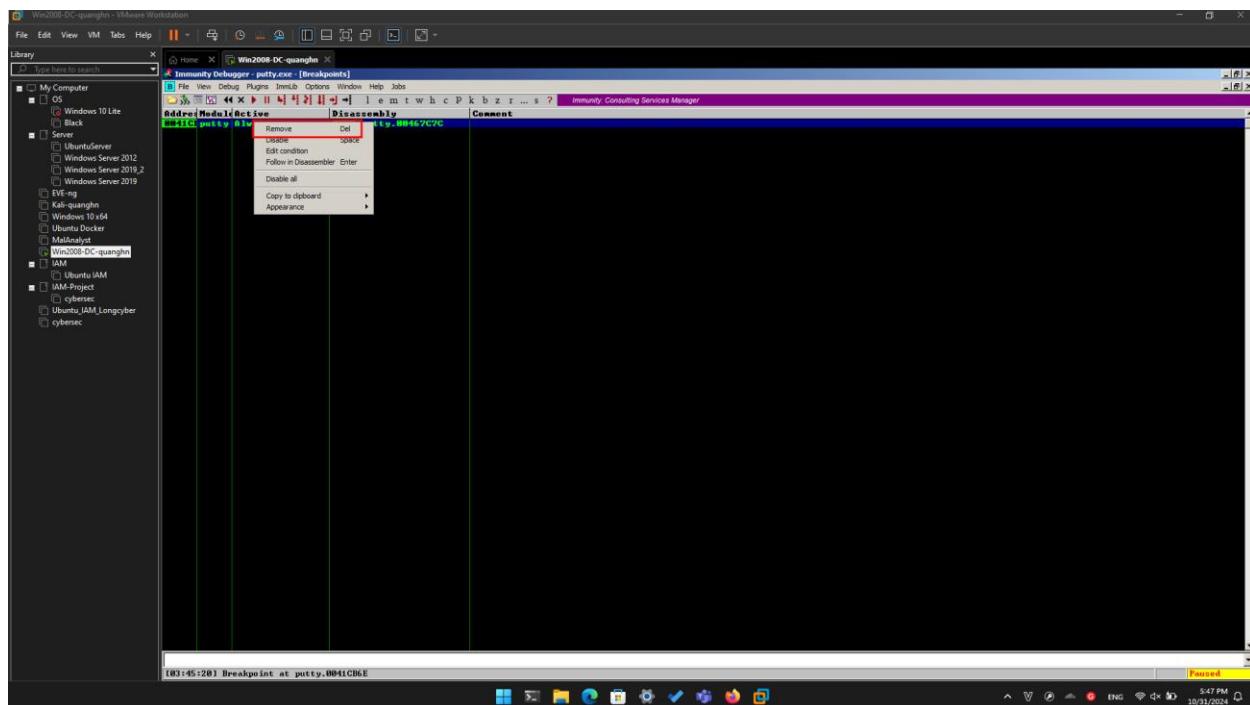
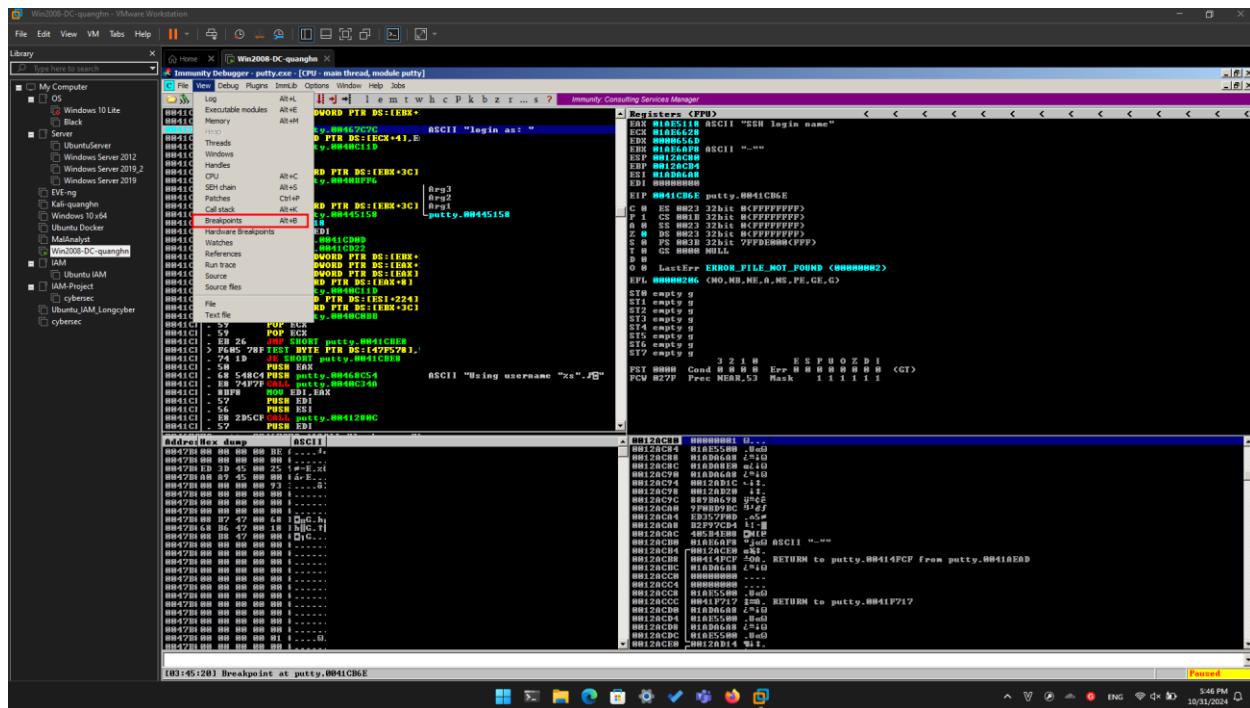


The program stopped at instruction 0041CB6E, as shown in the image above.

We'll use this instruction to hijack the program's execution.

## Task 2: Alter the Login Message

### Removing the Breakpoint



## Viewing the Stored Message

In Immunity, in the CPU window, in the Assembly Code pane, right-click the instruction at address **0041CB6E** and click "**Follow in Dump**", "**Immediate constant**", as shown below.

The lower left pane shows the stored "login as" message, in hexadecimal and ASCII text.

Screenshot of Immunity Debugger showing assembly and registers for the Win2008-DC-quangnh VM. A breakpoint is set at putty.0041C96E. The assembly window shows the code for the putty process, including the password check logic. The registers window shows various CPU registers. The memory dump window shows the memory dump starting at address 0041C96E.

Screenshot of Immunity Debugger showing assembly and registers for the Win2008-DC-quangnh VM. A breakpoint is set at putty.0041C96E. The assembly window shows the code for the putty process, including the password check logic. The registers window shows various CPU registers. The memory dump window shows the memory dump starting at address 0041C96E.

## Skipping the First Letter In the Message

An "Assemble at 0041CB6E" box appears, as shown below.

This shows the command at this location. It's a PUSH instruction, placing the address 467C7C onto the stack. That address points to the letter "l" in the ASCII string "login as: ", as shown on the right side of the instruction line, outlined in green in the image below.

In the "Assemble at 0041CB6E" box, change the last character to **D**, as shown below. This will move the pointer from the "l" to the "o" in the string "login as: ".

```

0041CB6E: push    edi
0041CB6F: mov     ecx,0040C91D
0041CB74: call    putty,0040C91D
0041CB79: add     esp,4
0041CB7C: pop     edi
0041CB7D: ret

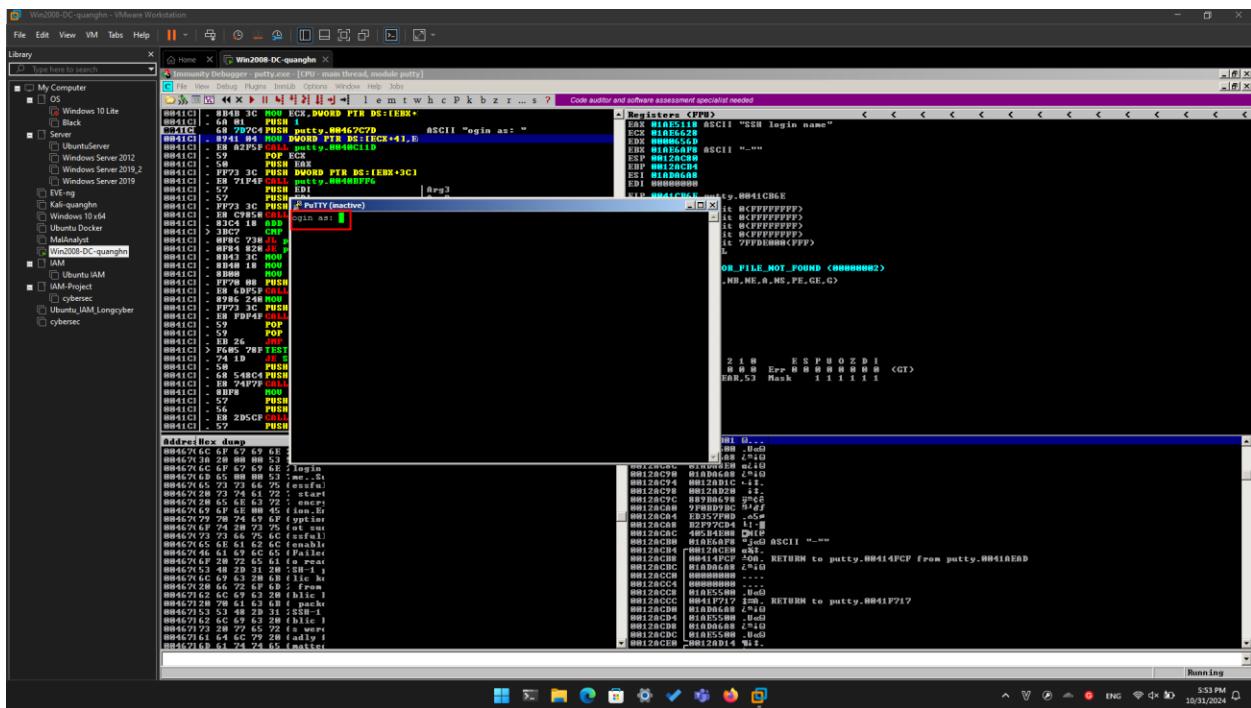
```

```

0041CB6E: push    edi
0041CB6F: mov     ecx,0040C91D
0041CB74: call    putty,0040C91D
0041CB79: add     esp,4
0041CB7C: pop     edi
0041CB7D: ret

```

## Running the Modified Program

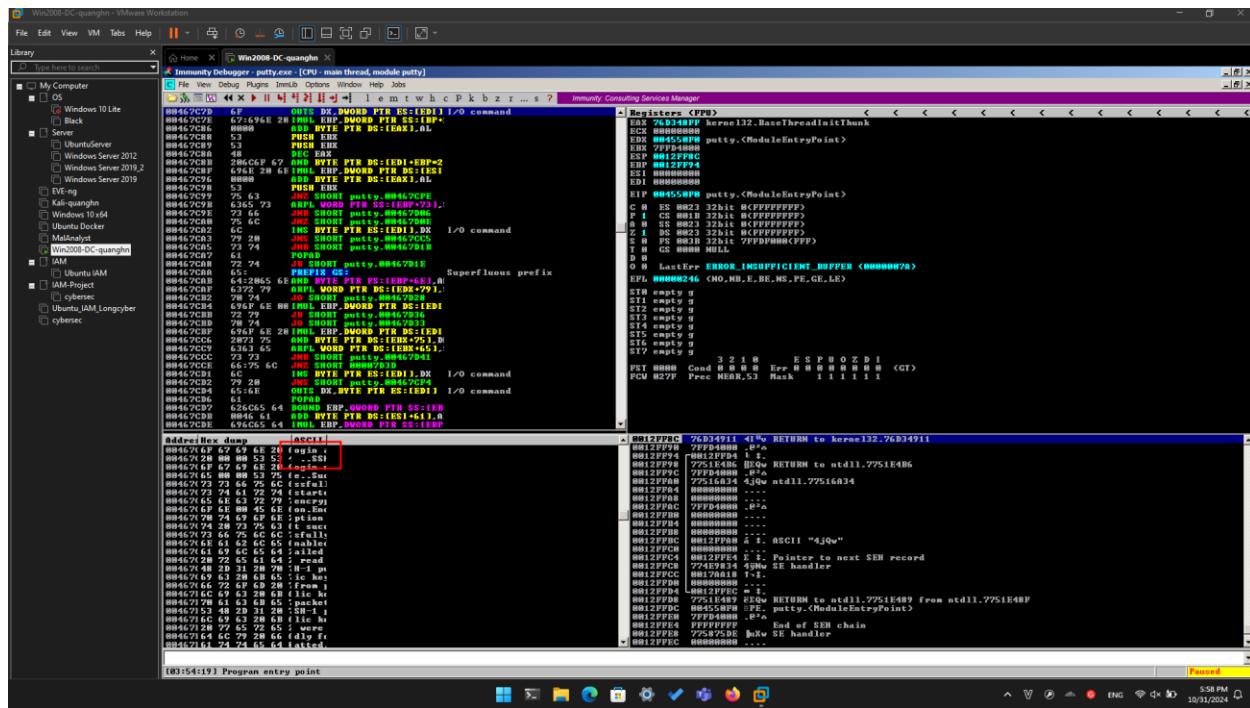
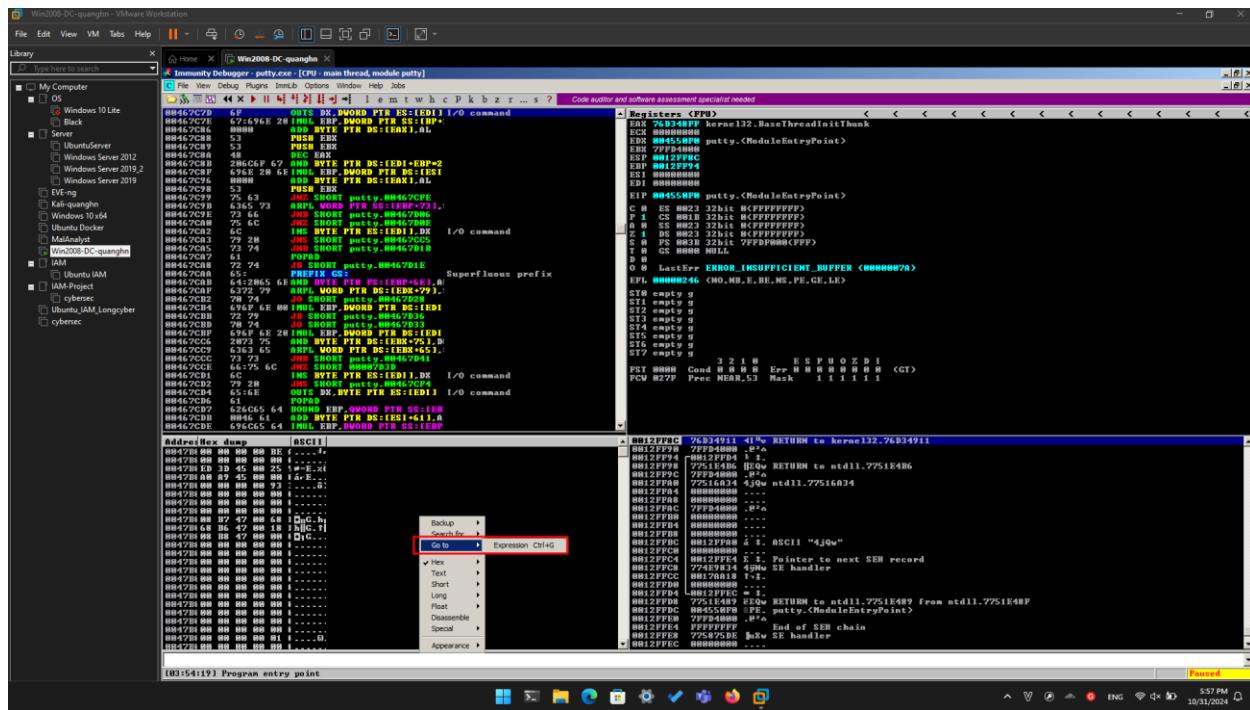


## Inserting Your Name

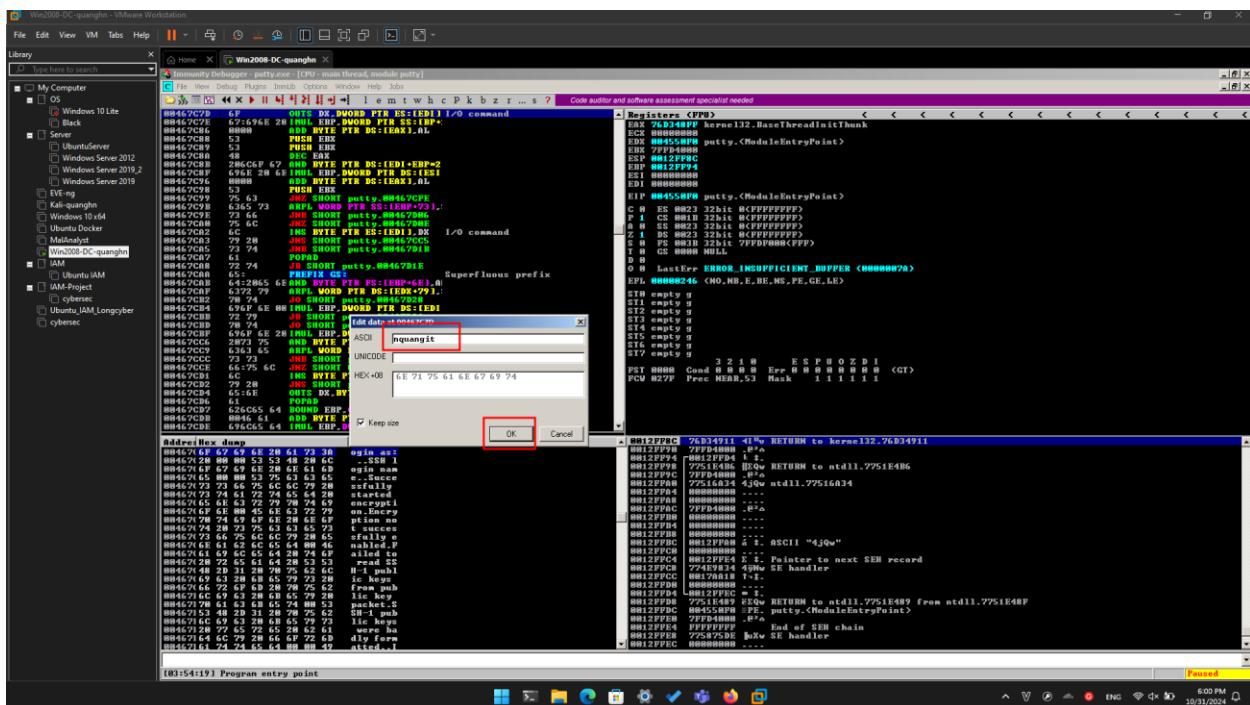
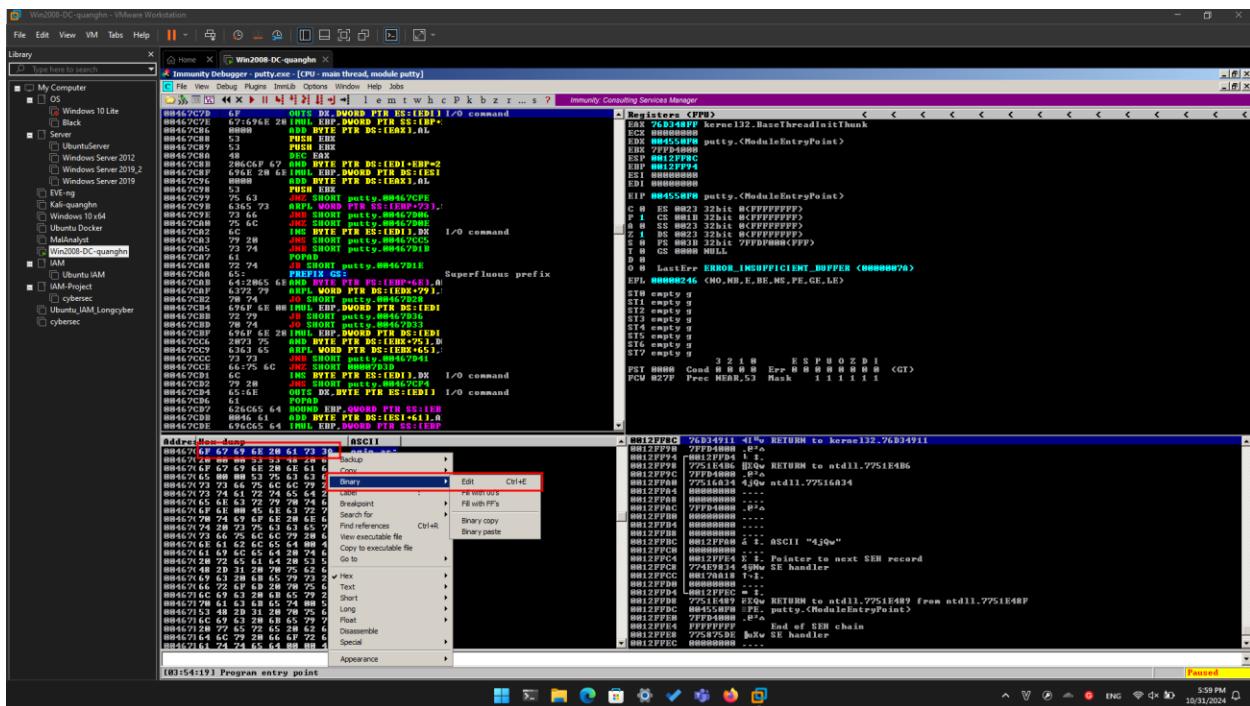
Now we want to change the text from "ogin as:" to your name.

Move your mouse into the lower left pane of the CPU window, which is the "hex dump" pane.

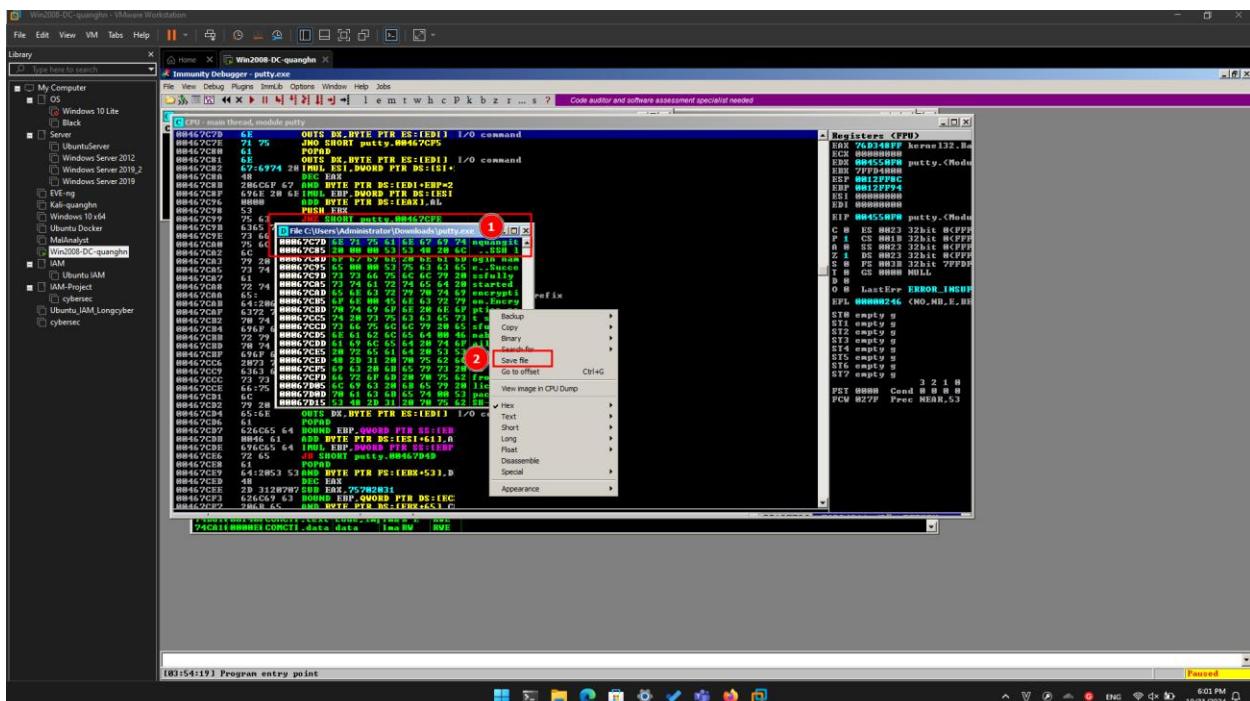
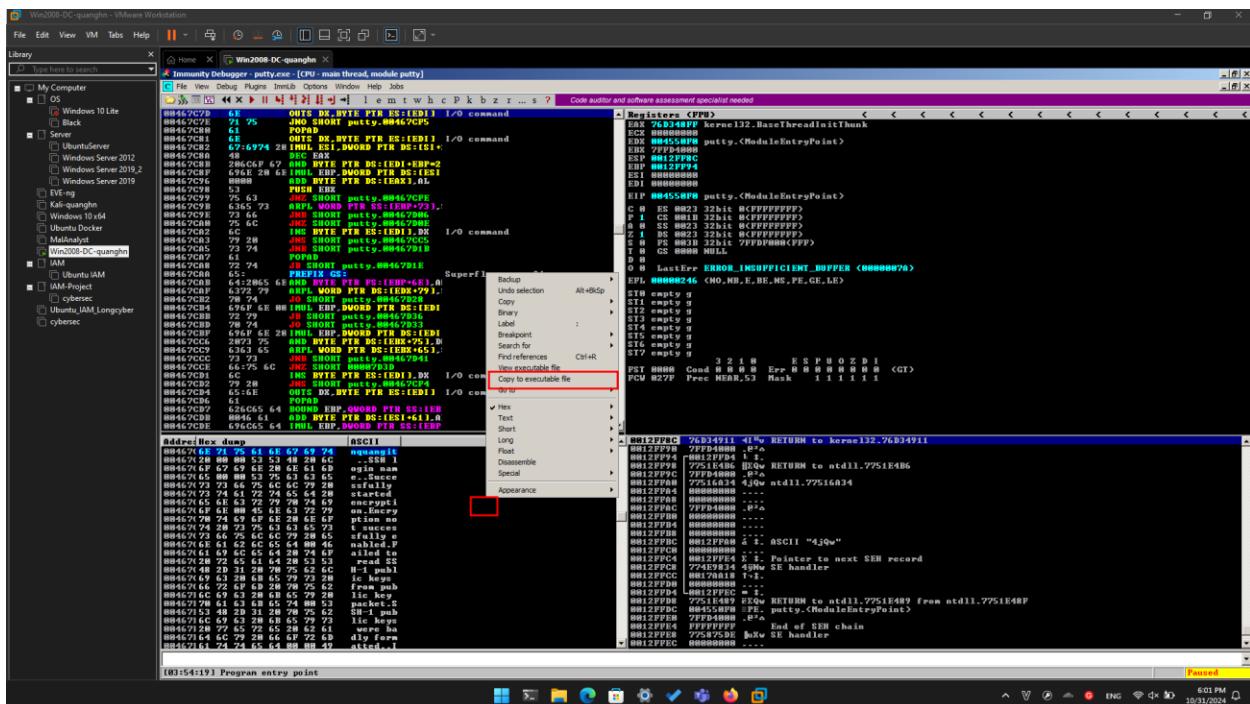
Right click, point to "**Go to**", and click **Expression**, as shown below.

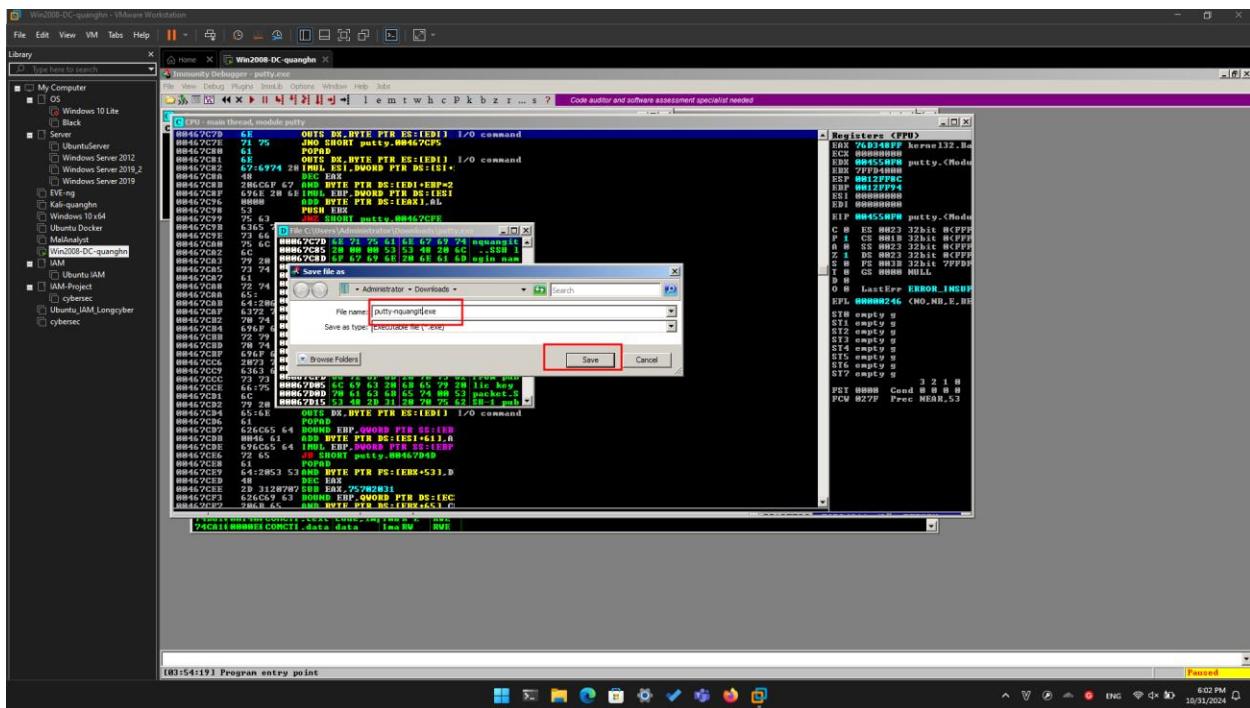


In the top left of the Hex Dump pane, point to **6F**, hold down the left mouse button, and select the entire row of 8 bytes. Then release the left button, point to **Binary**, and click **Edit**, as shown below.



## Saving the Modified EXE





## Running the Modified EXE

