

# Proj 8a: Simple EXE Hacking with Immunity (15 pts.)

## What You Need

A Windows machine, real or virtual. I used a Windows Server 2008 virtual machine.

## Purpose

To modify a Windows EXE file and save an altered version. This gives you practice with very simple features of the Immunity debugger.

---

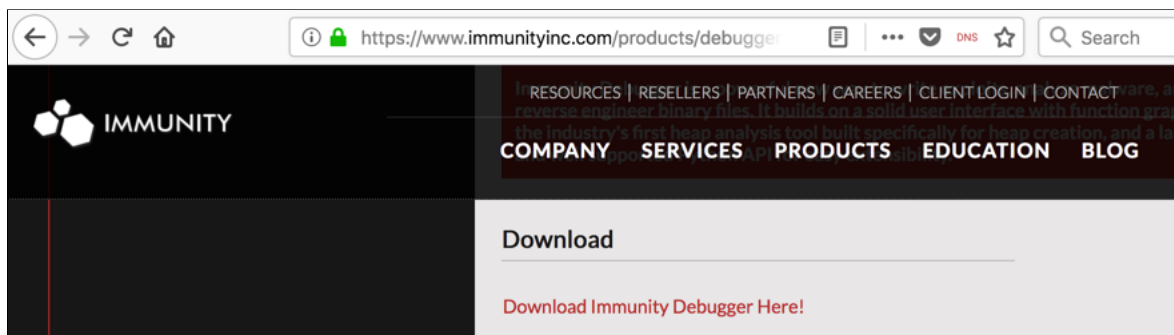
## Task 1: Target EXE Recon

### Get Immunity

On your Windows machine, in a browser, go to

<https://www.immunityinc.com/products/debugger/>

Click the "Download Immunity Debugger Here" link, as shown below.



If that link is not working, use this alternate download link: [ImmunityDebugger\\_1\\_85\\_setup.exe](#)

Install it with the default options.

### Get putty.exe

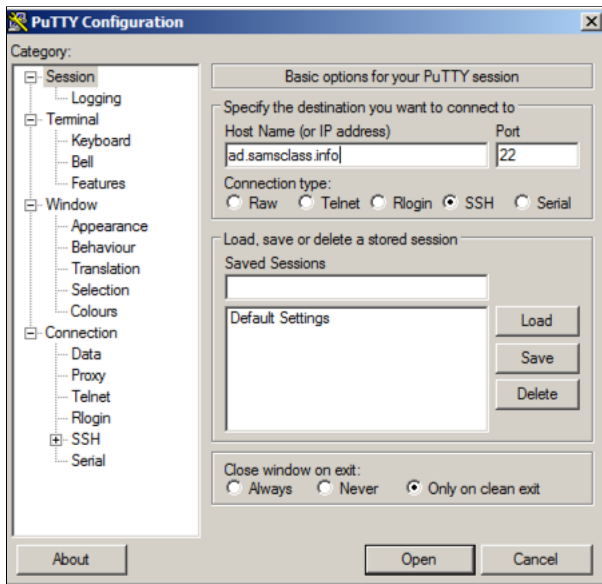
Putty is a common SSH client for Windows. It's harmless, but we will modify it to add Trojan code.

In a Web browser, right-click this link and save the putty.exe file:

<https://samsclass.info/127/proj/putty.exe>

### Running Putty

Double-click **putty.exe**. PuTTY opens, as shown below.

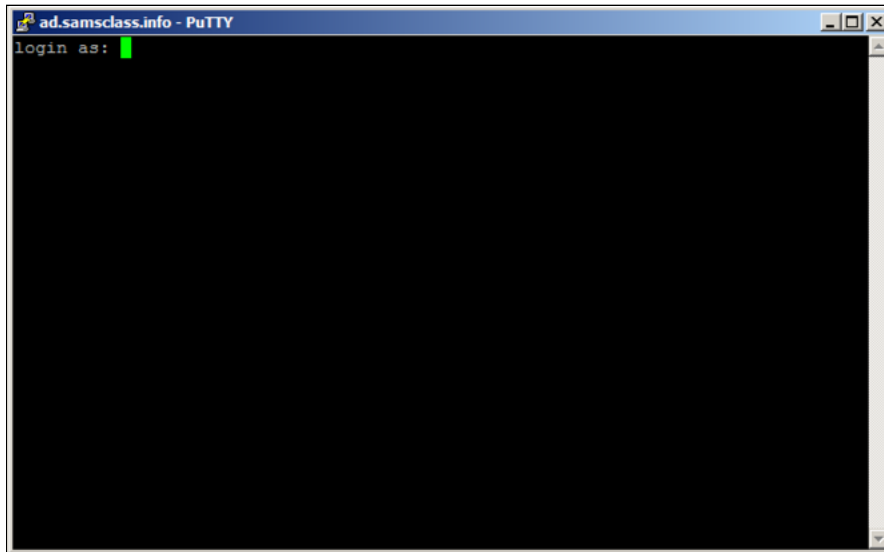


In the "Host Name (or IP address)" box, type

**ad.samsclass.info**

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

A black box opens, and shows a "**login as:**" prompt, as shown below.



You could connect to a server at this point, but that's not the point of this project. We will alter this program to do other things instead of printing "login as".

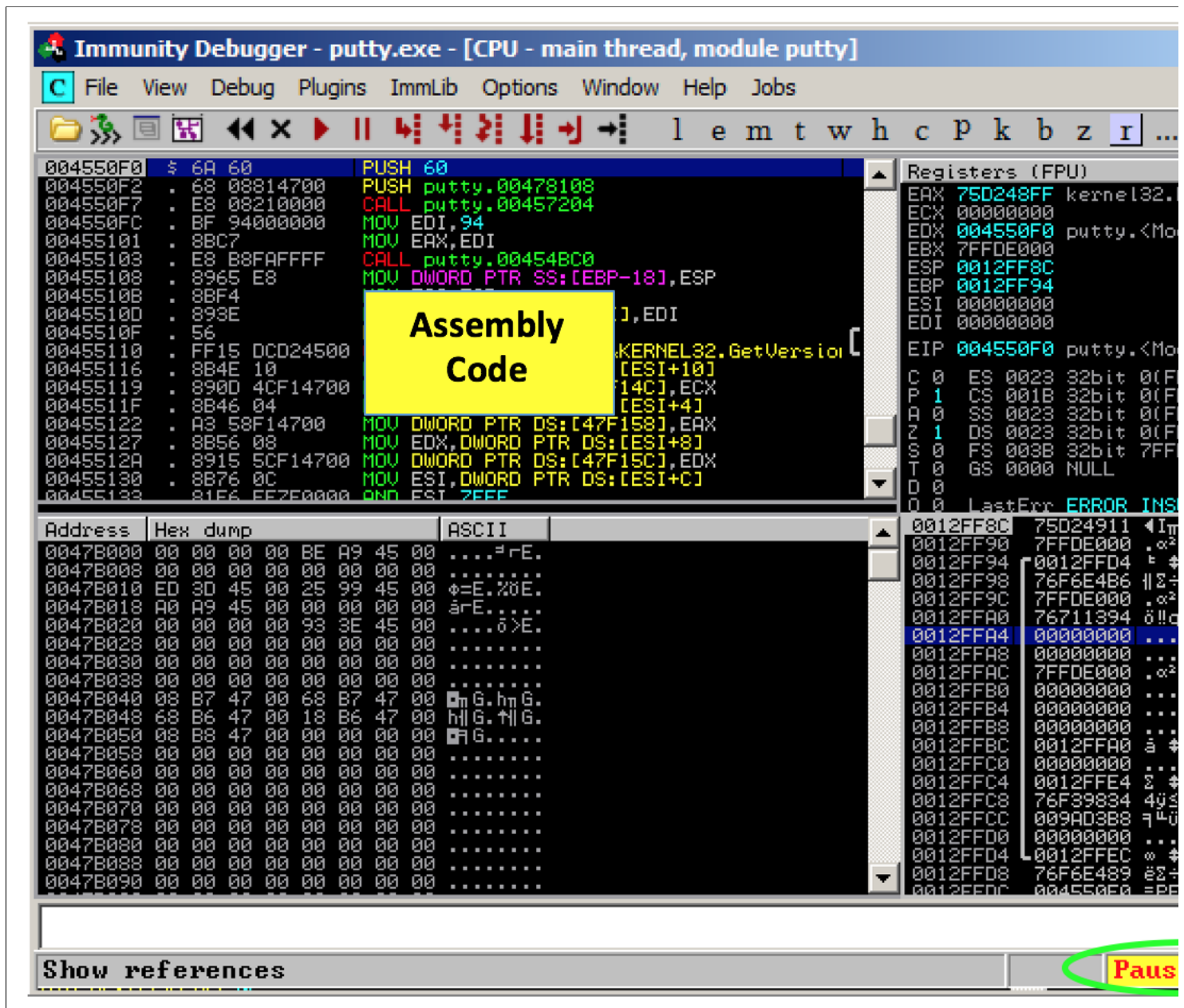
Close the Putty window.

## Starting the Immunity Debugger

Click **Start**. Search for **Immunity Debugger** and start it.

In Immunity, from the menu bar, click **File, Open**. Navigate to **putty.exe** and open it.

Immunity opens, as shown below. If your screen doesn't look like this, click **View, CPU** and maximize the CPU window.



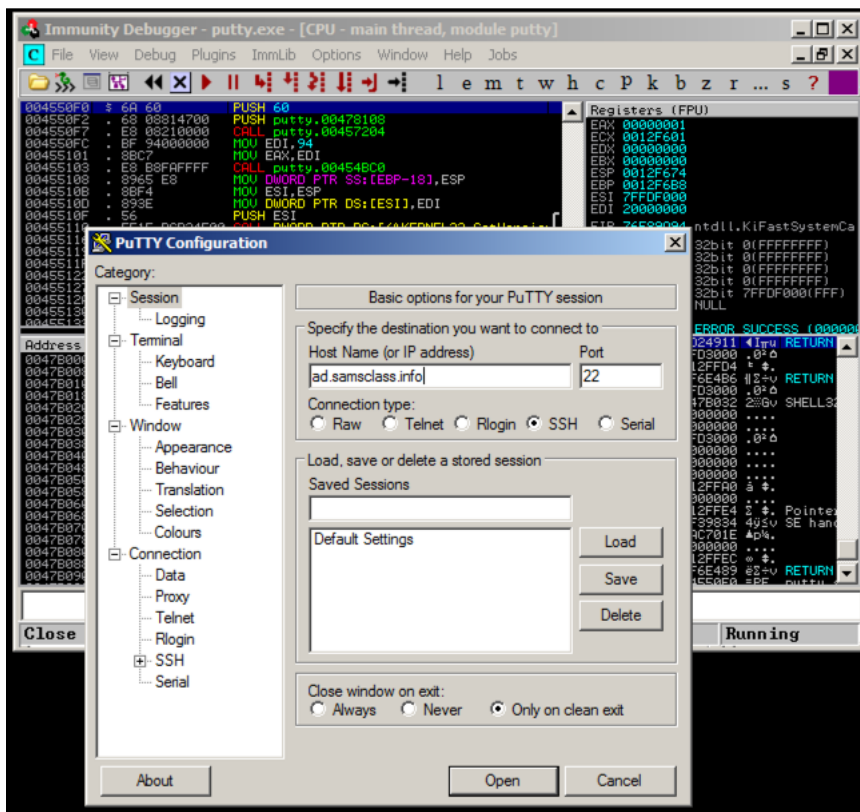
Immunity shows you a lot of data, but for now just notice the **Assembly Code** in the top left pane, and the **Paused** message in the lower right, as indicated in the figure above.

When you load a program into Immunity, it starts in a "Paused" state, with the Assembly Code window showing the first instruction.

## Running Putty in Immunity

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

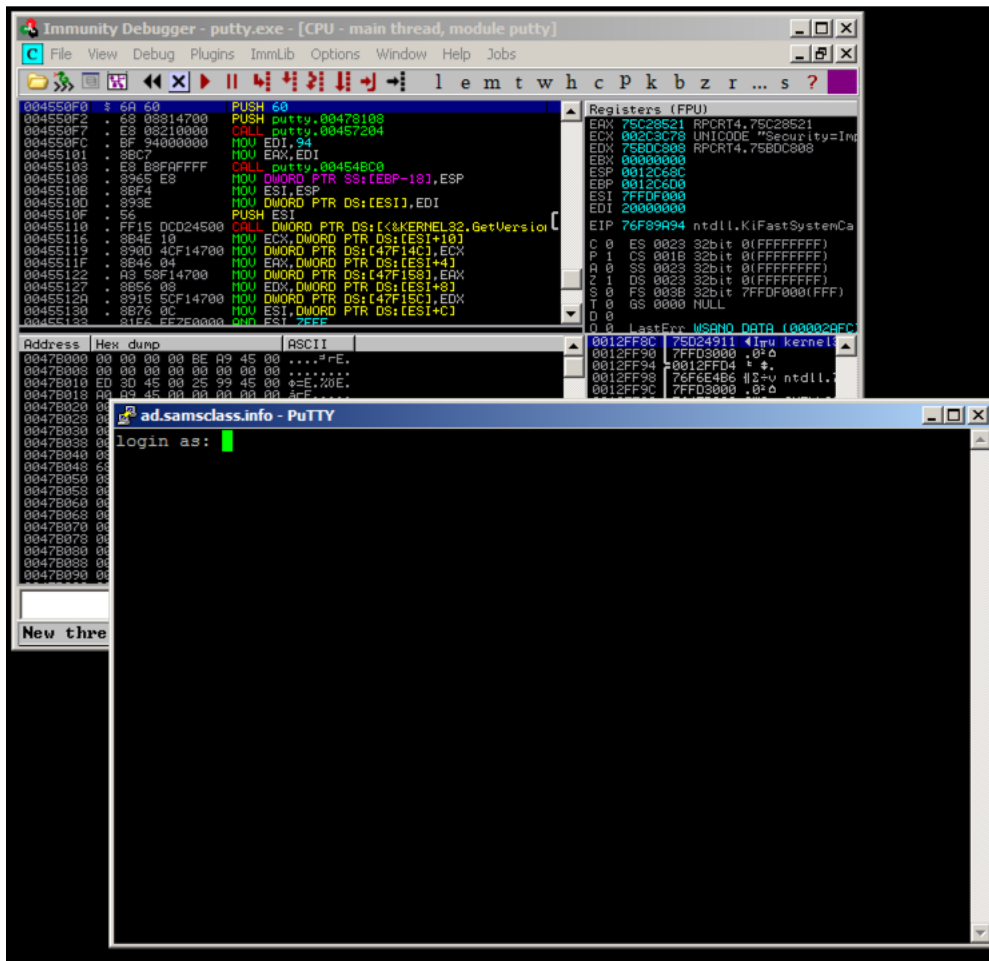
A Putty window opens, as shown below.



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

**ad.samsclass.info**

At the bottom, click the **Open** button. The "login as" message appears, as shown below.



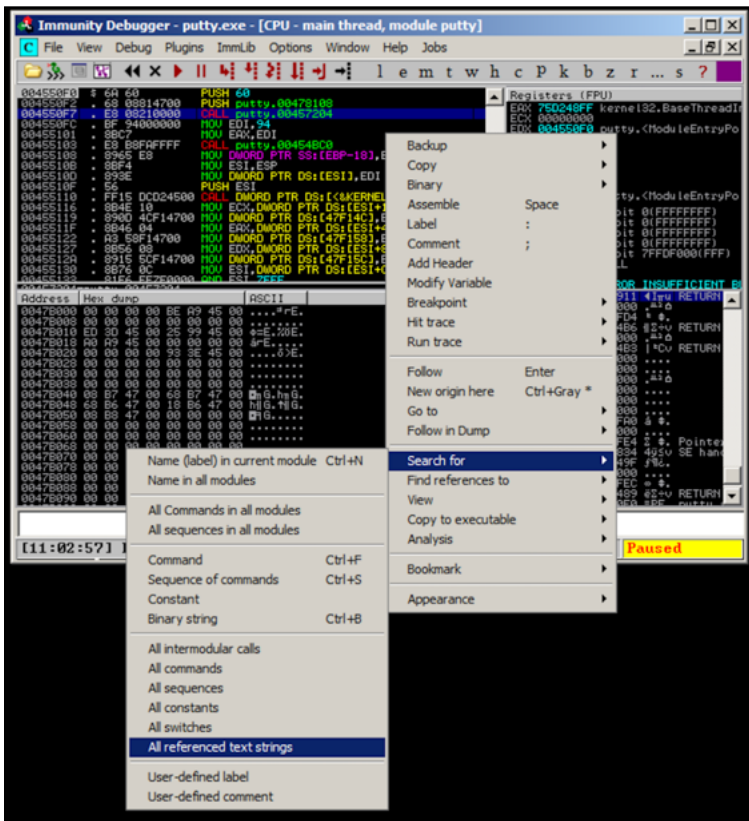
Putty is running, but it's under the control of Immunity, so we can modify its execution.

## Finding the "login as" Code

Close the Putty window.

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

In Immunity, in the "Assembly Code" pane, right-click. Point to "Search for". Click "All referenced text strings", as shown below.

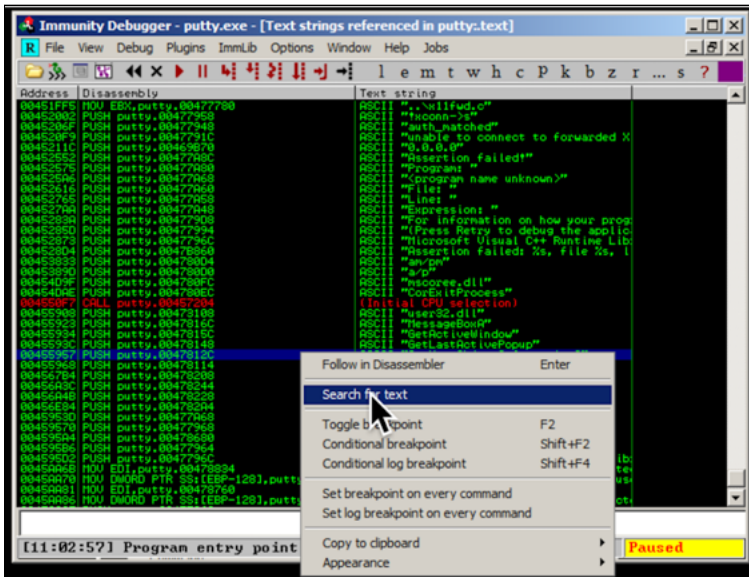


A "Text strings referenced in putty.exe" window opens, showing all the strings in the program.

Scroll to the top of the window and click on the first line, so it is highlighted.

Right-click in the window, and click "Search for text", as shown below.

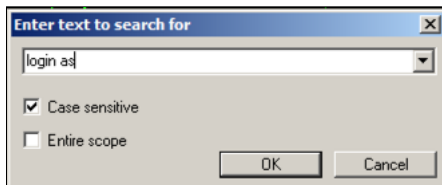
Note: the search only goes down, and does not wrap, so if you begin a search near the bottom of the window it won't find "login as".



In the "Enter text to search for" box, type

**login as**

as shown below.



Click **OK**.

Immunity finds the ASCII string "login as", and the instruction that uses it, as shown below. This instruction is at address 00417053.

R Text strings referenced in putty.exe		
Address	Disassembly	Text string
00416EA1	PUSH putty.00467DC0	ASCII "Using 3DES encryption"
00416EA8	PUSH putty.00467DA4	ASCII "Using single-DES encryption"
00416EF4	PUSH putty.00467D84	ASCII "Trying to enable encryption..."
00416F18	MOV EAX,putty.0046D3E0	ASCII "79B"
00416F47	PUSH putty.00467D68	ASCII "Initialised %s encryption"
00416F57	PUSH putty.00467D3C	ASCII "Installing CRC compensation atta
00416FC3	PUSH putty.00467D14	ASCII "SSH-1 public keys were badly for
00416FCD	PUSH putty.00467CDC	ASCII "Failed to read SSH-1 public keys
00416FEF	PUSH putty.00467CB8	ASCII "Encryption not successfully enab
00416FF9	PUSH putty.00467C98	ASCII "Successfully started encryption"
00417042	MOV DWORD PTR SS:[ESP],putty.00467C88	ASCII "SSH login name"
00417053	PUSH putty.00467C7C	ASCII "login as: "
004170B5	PUSH putty.00467C70	ASCII "Key refused"
004170CC	PUSH putty.00467C58	ASCII "Received RSA challenge"
004172A0	PUSH putty.00467C34	ASCII "Reading private key file \"%150s"
004172FD	PUSH putty.00467C14	ASCII "Unable to load private key (%s)"
0041731B	PUSH putty.00467BE4	ASCII "Unable to load private key file "
0041734B	PUSH putty.00467BC0	ASCII "Unable to use this key file (%s)"
0041736D	PUSH putty.00467B98	ASCII "Unable to use key file \"%150s"

Right-click again, and click **"Search next"**.

Immunity finds another line of code that uses this string, as shown below. This instruction is at address 0041CB6E.

R Text strings referenced in putty.exe		
Address	Disassembly	Text string
0041C703	PUSH putty.00461680	ASCII "J"
0041C733	PUSH putty.00468D58	ASCII "New SSH password"
0041C75A	PUSH putty.0045D6EC	ASCII "%.*s"
0041C771	PUSH putty.00468D1C	ASCII "Current password (blank for prev
0041C78C	PUSH putty.00468D04	ASCII "Enter new password: "
0041C7A4	PUSH putty.00468CEC	ASCII "Confirm new password: "
0041C84F	PUSH putty.00468CD0	ASCII "Passwords do not matchJ"
0041C8A1	PUSH putty.00467604	ASCII "Unable to authenticate"
0041C8D7	MOV DWORD PTR SS:[ESP],putty.004691F0	ASCII "ssh-connection"
0041C8E9	MOV DWORD PTR SS:[ESP],putty.00467394	ASCII "password"
0041C93B	PUSH putty.00468CBC	ASCII "Sent new password"
0041CA11	PUSH putty.00468CAC	ASCII "Access granted"
0041CABA	PUSH putty.00466488	ASCII "..\ssh.c"
0041CABF	PUSH putty.00468C8C	ASCII "s->type == AUTH_TYPE_PASSWORD"
0041CACF	PUSH putty.00468C6C	ASCII "Password authentication failed"
0041CADF	PUSH putty.0046786C	ASCII "Access deniedJ"
0041CB5C	MOV DWORD PTR SS:[ESP],putty.00467C88	ASCII "SSH login name"
0041CB6E	PUSH putty.00467C7C	ASCII "login as: "
0041CBCC	PUSH putty.00468CE4	ASCII "Using username \"%s\" J"

## Using Breakpoints

We'll set a breakpoint at this instruction, at address 0041CB6E.

On your keyboard, press the **F2** key. Mac users, press **fn+F2**. The address turns red, as shown below, to indicate that there's a breakpoint here.

R Text strings referenced in putty.exe		
Address	Disassembly	Text string
0041C703	PUSH putty.00461680	ASCII "␣"
0041C733	PUSH putty.00468058	ASCII "New SSH password"
0041C75A	PUSH putty.0045D6EC	ASCII "%.*s"
0041C771	PUSH putty.0046801C	ASCII "Current password (blank for prev
0041C78C	PUSH putty.00468004	ASCII "Enter new password: "
0041C7A4	PUSH putty.00468CEC	ASCII "Confirm new password: "
0041C84F	PUSH putty.00468CD0	ASCII "Passwords do not match␣"
0041C8A1	PUSH putty.00467604	ASCII "Unable to authenticate"
0041C8D7	MOV DWORD PTR SS:[ESP],putty.004691F0	ASCII "ssh-connection"
0041C8E9	MOV DWORD PTR SS:[ESP],putty.00467394	ASCII "password"
0041C93B	PUSH putty.00468CBC	ASCII "Sent new password"
0041CA11	PUSH putty.00468CAC	ASCII "Access granted"
0041CABA	PUSH putty.00466488	ASCII "..\ssh.c"
0041CABF	PUSH putty.00468C8C	ASCII "s->type == AUTH_TYPE_PASSWORD"
0041CACF	PUSH putty.00468C6C	ASCII "Password authentication failed"
0041CADF	PUSH putty.0046786C	ASCII "Access denied␣"
0041CB5C	MOV DWORD PTR SS:[ESP],putty.00467C88	ASCII "SSH login name"
0041CB65	PUSH putty.00467C7C	ASCII "login as: "
0041CBCC	PUSH putty.00468C54	ASCII "Using username \"%s\" ␣"

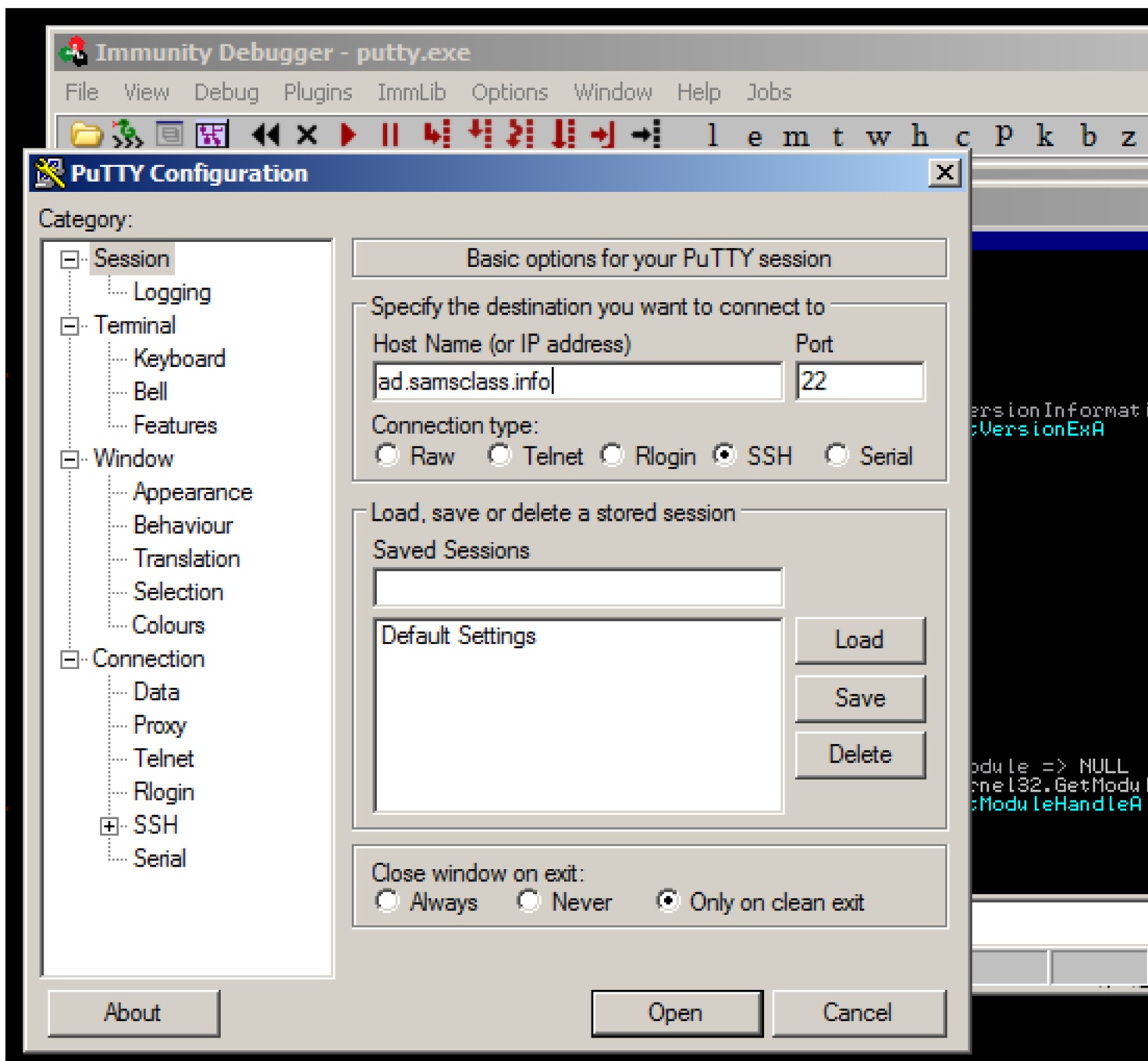
In Immunity, from the menu bar, click **Debug, Restart**.

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.



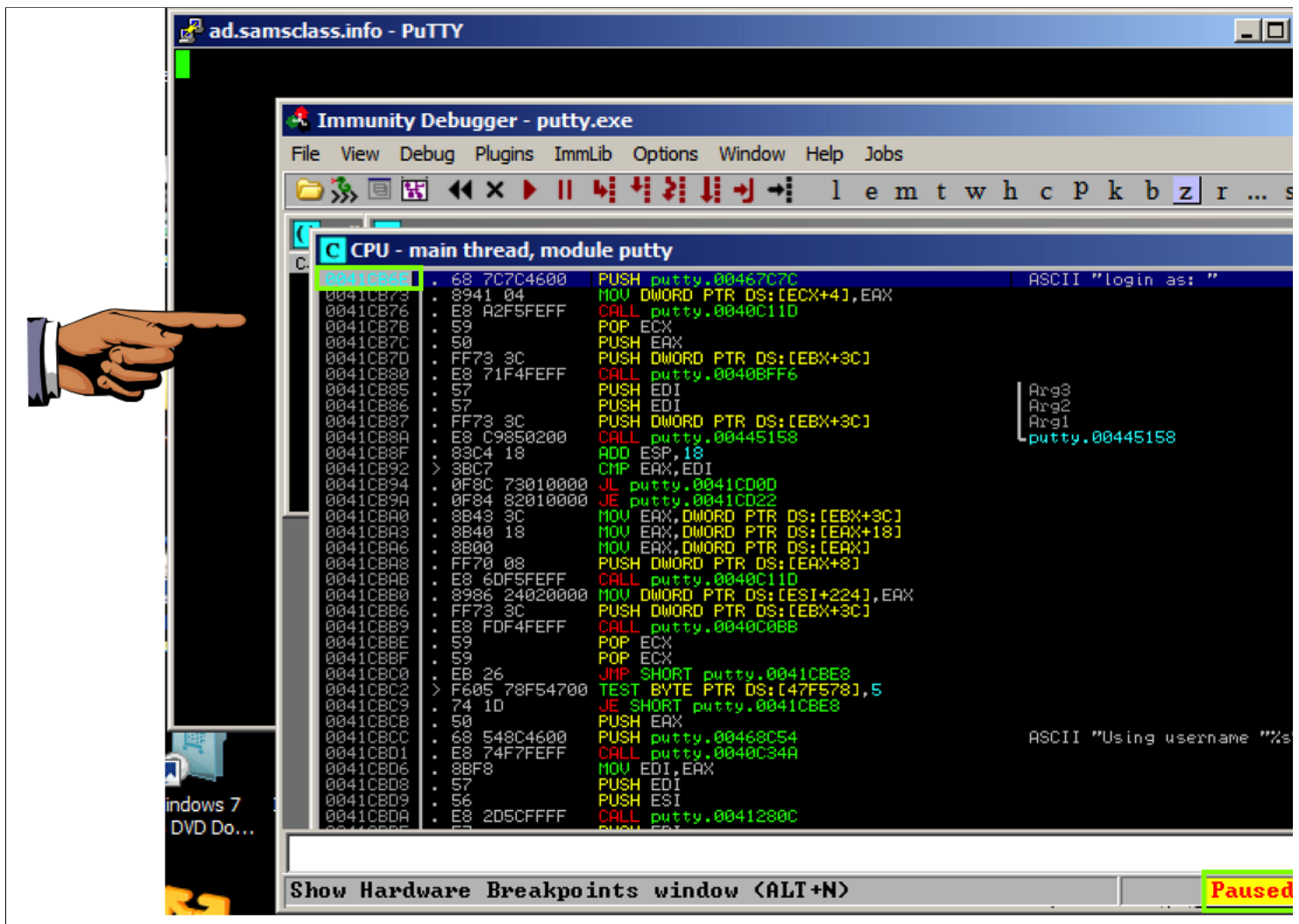


Click in the Putty window. In the "Host Name (or IP address)" box, type

**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.

## Saving a Screen Image

Make sure you can see the address **0041CB6E** in the top left of the CPU window, and **Paused** in the lower right, as shown above.

Press the **PrintScr** key to copy the whole desktop to the clipboard.

**YOU MUST SUBMIT A FULL-SCREEN IMAGE FOR FULL CREDIT!**

Paste the image into Paint.

Save the document with the filename "**YOUR NAME Proj 8a1**", replacing "YOUR NAME" with your real name.

## Task 2: Alter the Login Message

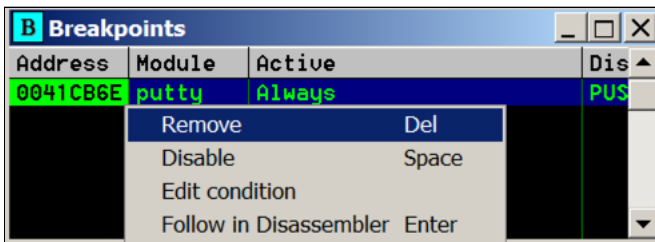
### Removing the Breakpoint

We don't need the breakpoint any more, so we'll remove it.

In Immunity, from the menu bar, click **View**, **Breakpoints**.

A "Breakpoints" window opens, showing the breakpoint.

Right-click the breakpoint and click **Remove**, as shown below.

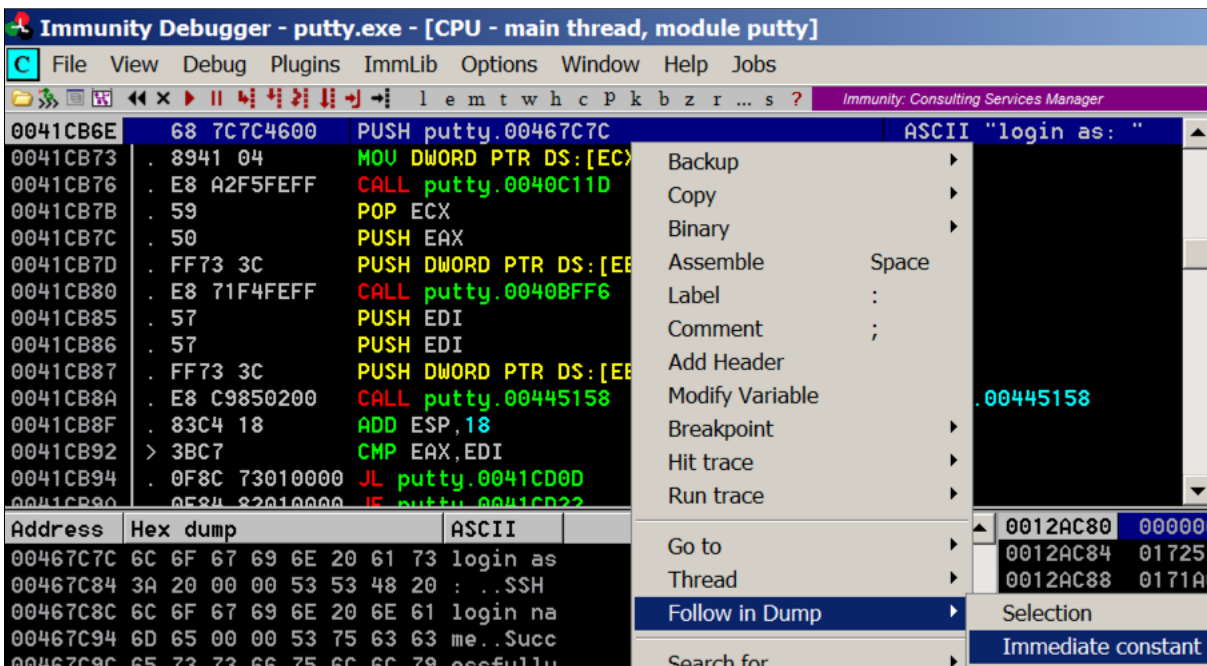


Close the "Breakpoints" window.

## Viewing the Stored Message

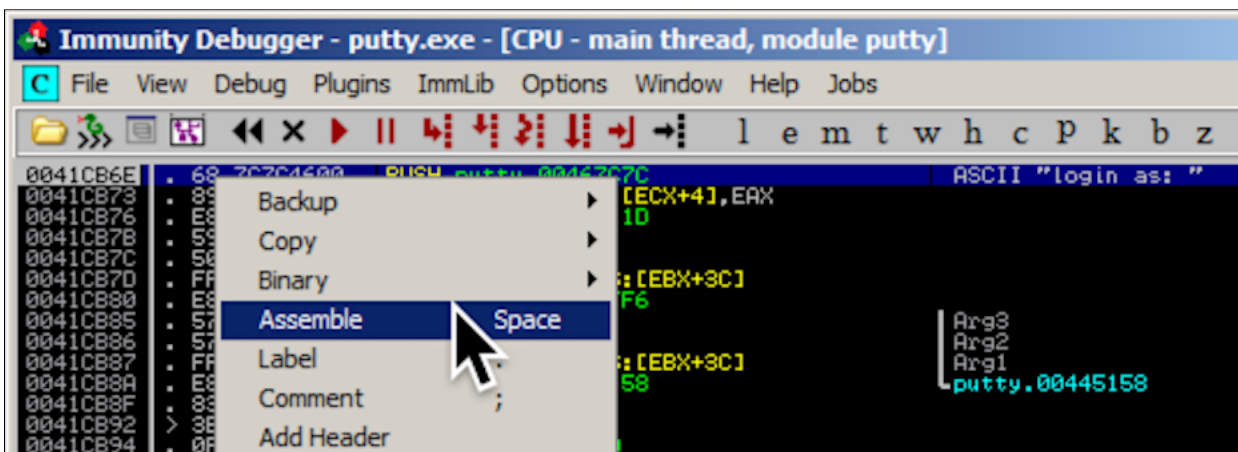
In Immunity, in the CPU window, in the Assembly Code pane, right-click the instruction at address **0041CB6E** and click **"Follow in Dump"**, **"Immediate constant"**, as shown below.

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



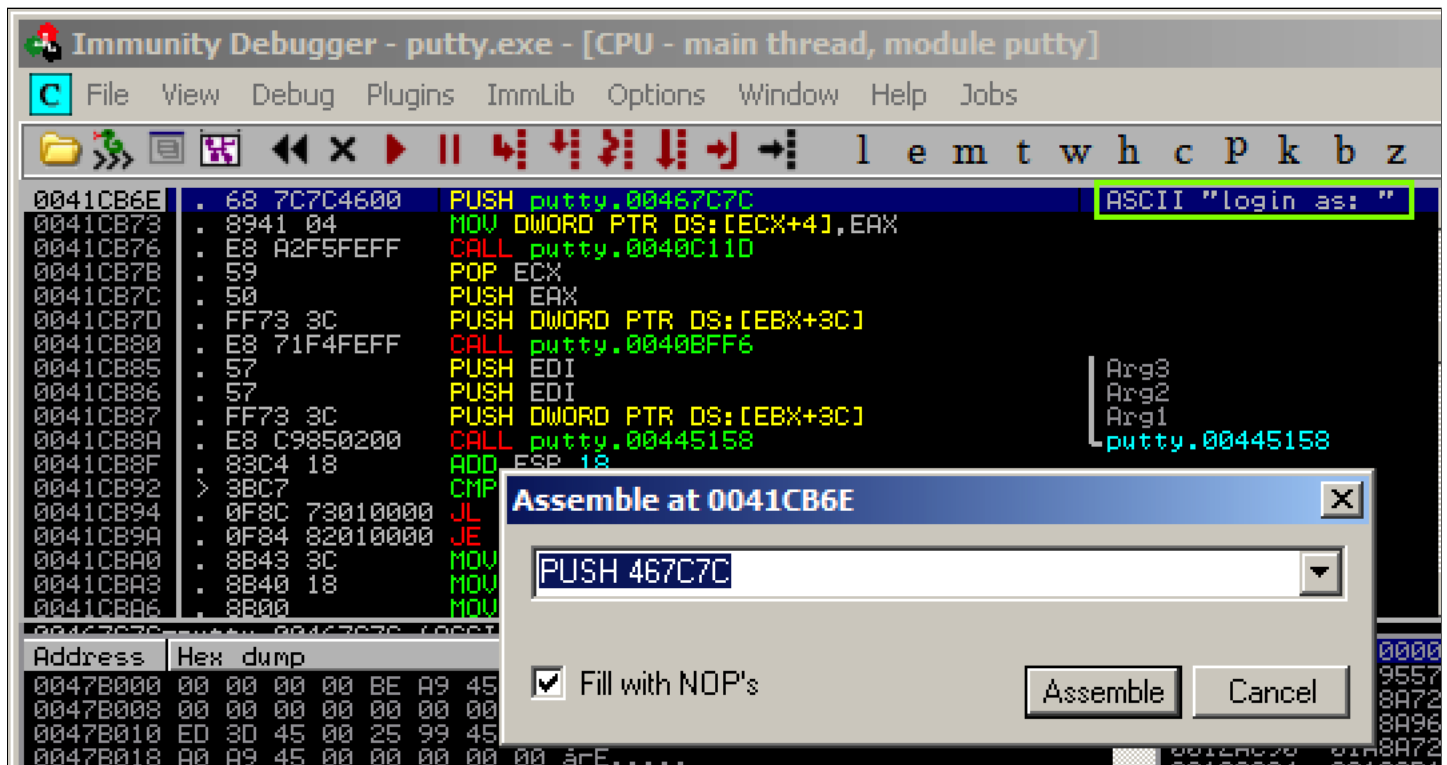
## Skipping the First Letter In the Message

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

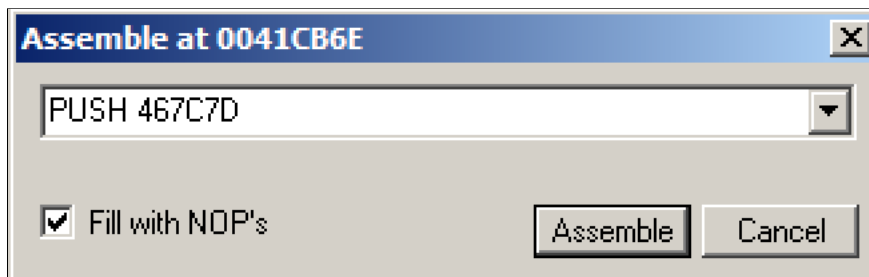


An "Assemble at 0041CB6E" box appears, as shown below.

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.



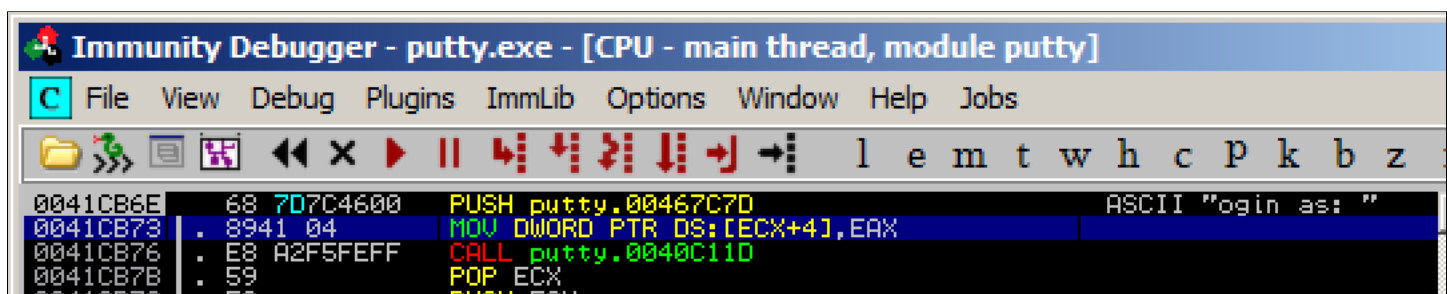
In the "Assemble at 0041CB6E" box, change the last character to **D**, as shown below. This will move the pointer from the "l" to the "o" in the string "login as: ".



Click the **Assemble** button.

Click the **Cancel** button.

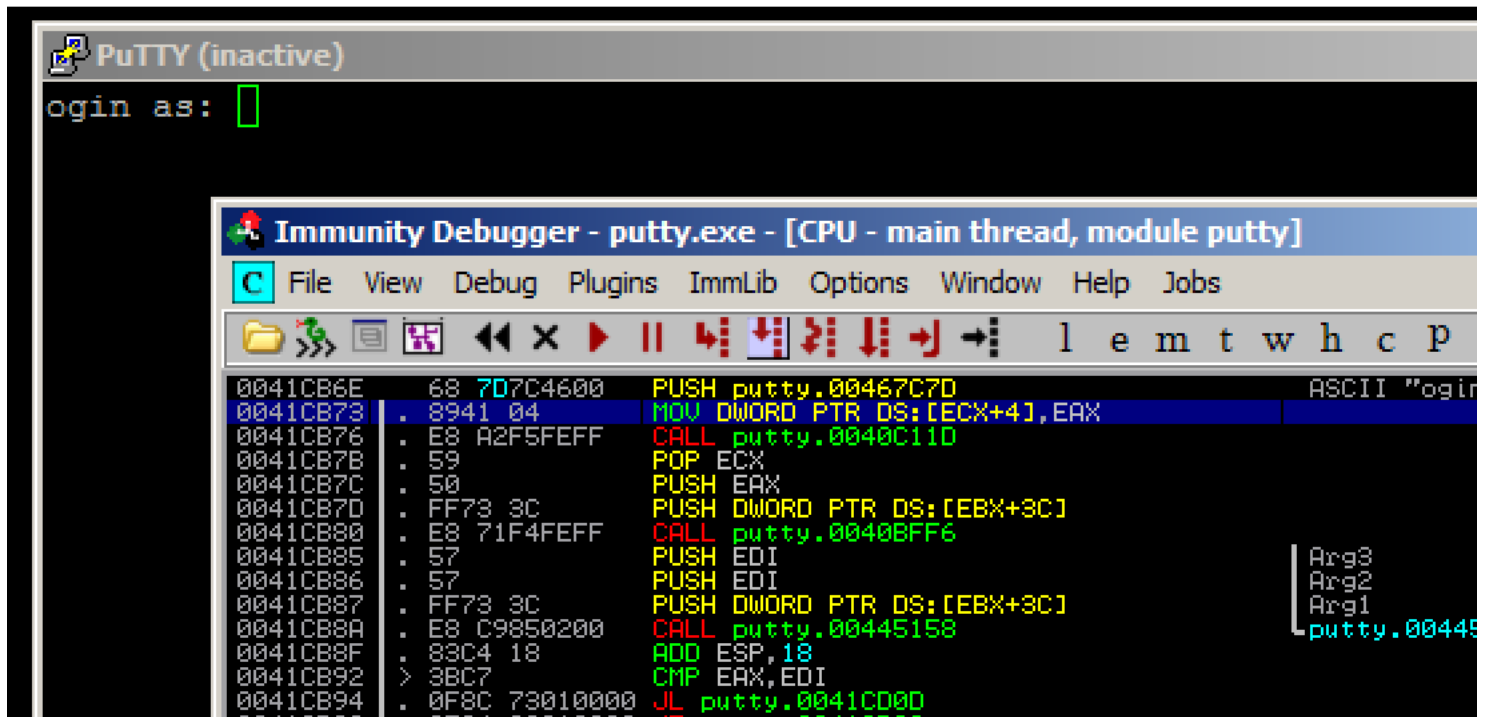
The message on the right now says "ogin as: ", as shown below.



## Running the Modified Program

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

The black login window appears, with the message "ogin as: ", as shown below.

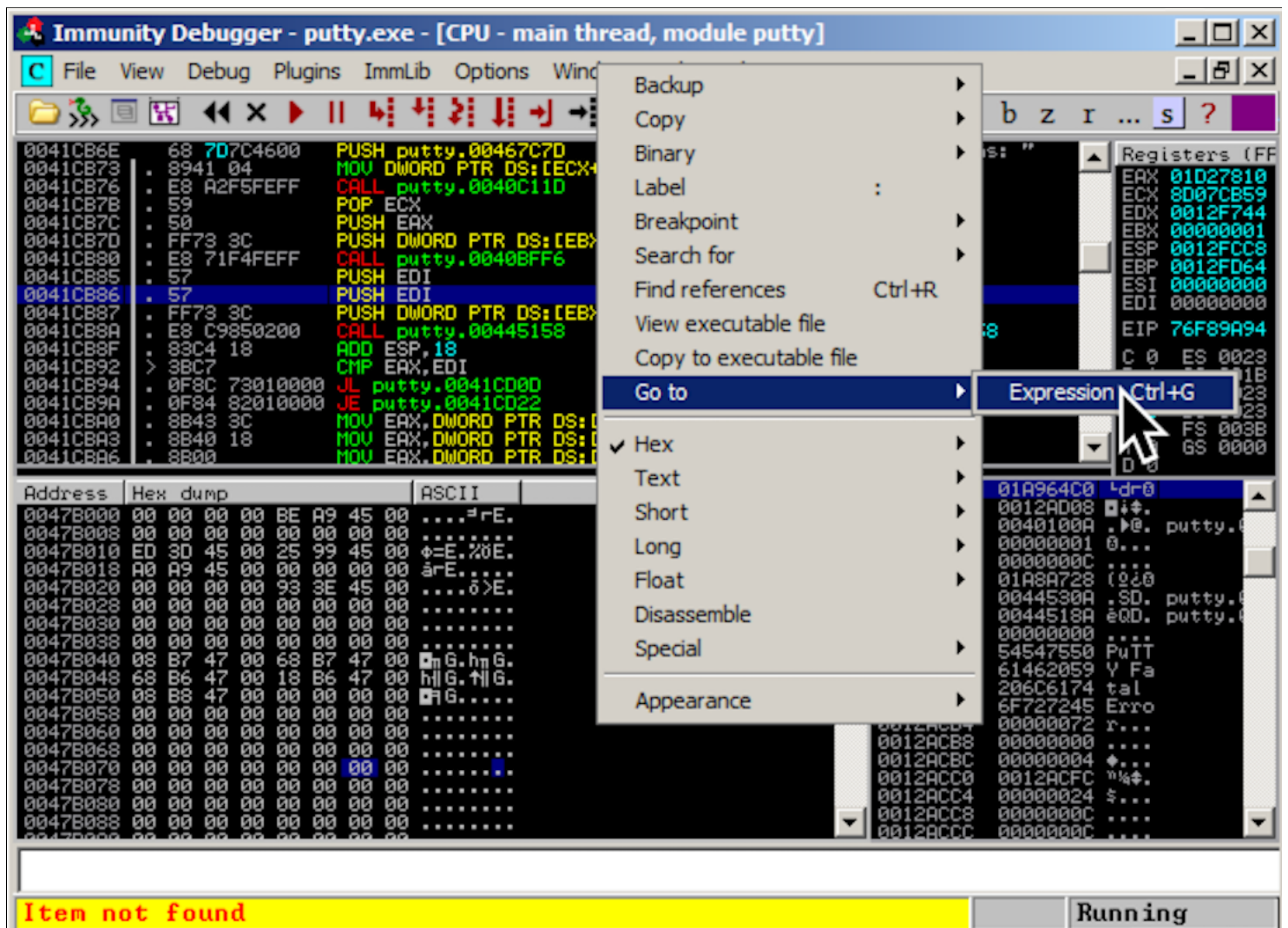


## Inserting Your Name

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

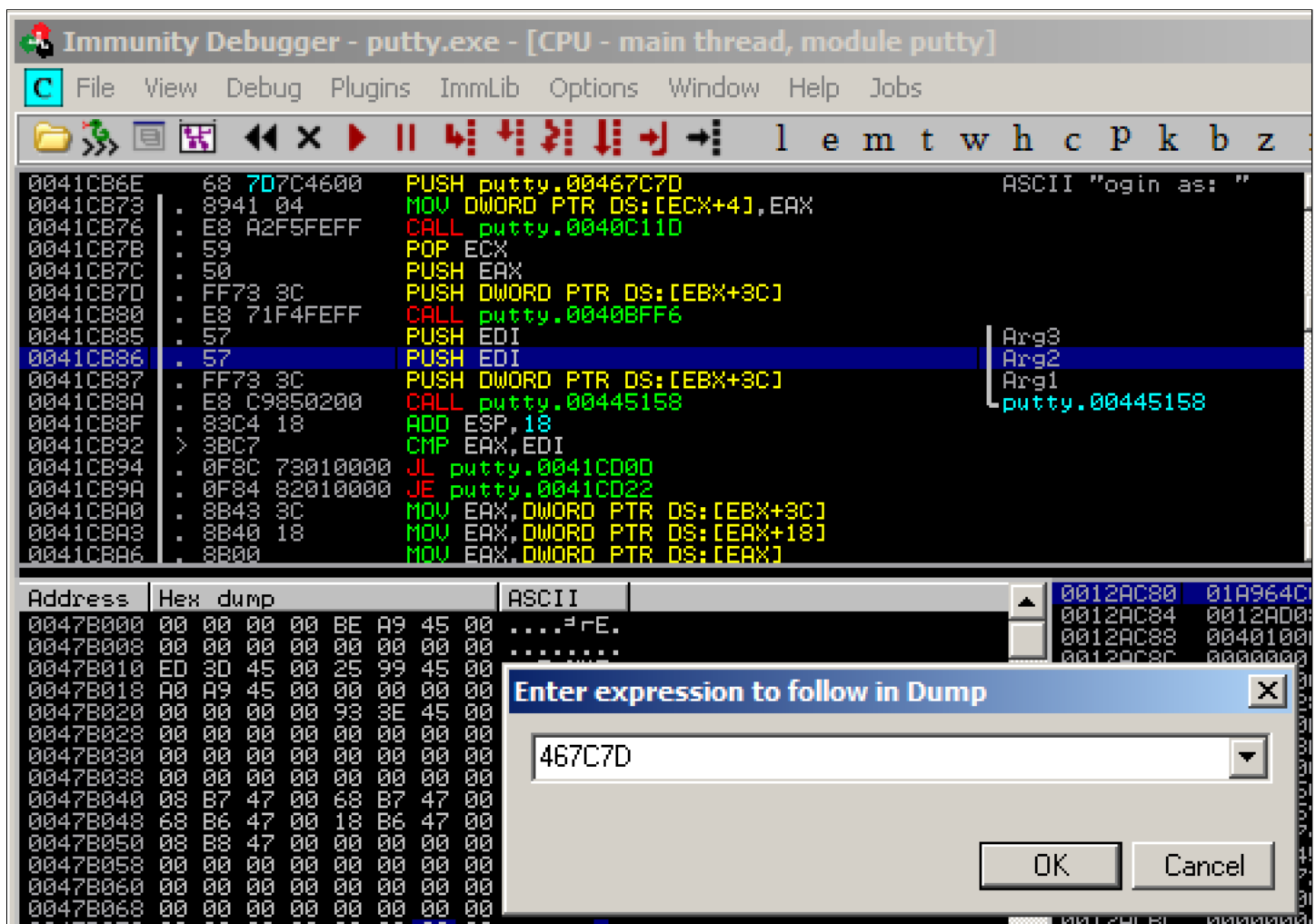
Move your mouse into the lower left pane of the CPU window, which is the "hex dump" pane.

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

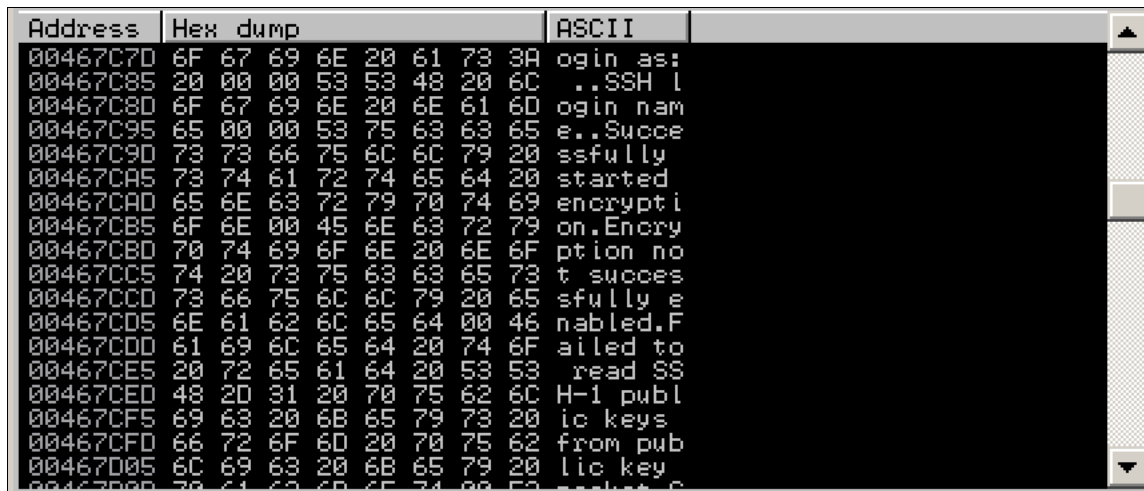


Enter 467C7D into the box, as shown below. Click OK.





The Hex Dump shows the text "ogin as: ", 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.





Click in the ASCII box, press Backspace to move back to the start, and overwrite the message with some version of your name. Make sure you insert exactly 8 letters. Don't use the literal text "YOURNAME", replace it with your own name.

Edit data at 00467C7D

ASCII

YOURNAME

UNICODE

HEX +08

59 4F 55 52 4E 41 4D 45

☒ Keep size

OK

Cancel

Click **OK**.

### Saving the Modified EXE

In Immunity, in the lower left pane of the CPU window, right-click and click **"Copy to Executable File"**, as shown below.

Address	Hex dump	ASCII	
00467C7D	59 4F 55 52 4E 41 4D 45	YOURNAME	
00467C85	20 00 00 53 53 48 20 6C	..SSH 1	
00467C8D	6F 67 69 6E 20 6E 61 6D	ogin nam	
00467C95	65 00 00 53 75 63 63 65	e..Succe	
00467C9D	73 73 66 75 6C 6C 79 20	ssfully	

Search for

Find references

View executable file

Copy to executable file

Go to

A new window pops up, with a title beginning with "File", as shown below.

Right-click in the new window and click **"Save file"**.

File C:\Users\Administrator\Desktop\putty.exe

00067C7D

59 4F 55 52 4E 41 4D 45

YOURNAME

00067C85

20 00 00 53 53 48 20 6C

..SSH 1

00067C8D

6F 67 69 6E 20 6E 61 6D

ogin nam

00067C95

65 00 00 53 75 63 63 65

e..Succe

00067C9D

73 73 66 75 6C 6C 79 20

ssfully

00067CA5

73 74 61 72 74 65 64 20

started

00067CAD

65 6E 63 72 79 70 74 69

encrypti

00067CB5

6F 6E 00 45 6E 63 72 79

on.Encry

00067CBD

70 74 69 6F 6E 20 6E 6F

ption no

00067CC5

74 20 73 75 63 63 65 73

t succes

00067CCD

73 66 75 6C 6C 79 20 65

sfully e

00067CD5

6E 61 62 6C 65 64 00 46

nabled.F

00067CDD

61 69 6C 65 64 20 74 6F

ailed to

Backup

Copy

Binary

Search for

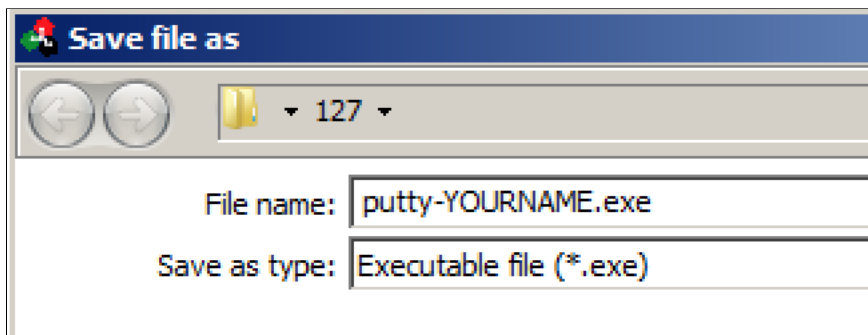
Save file

Go to offset

View image in CPU Dump

Save the file as "putty-YOURNAME.exe", replacing YOURNAME with your own name, as shown below.

Click **Save**.



## Running the Modified EXE

Close Immunity.

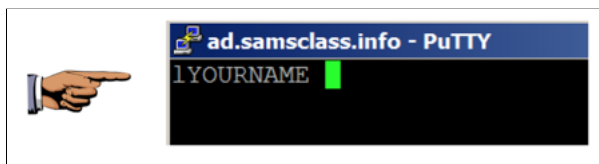
Double-click **putty-YOURNAME.exe**.

In the "Host Name (or IP address)" box, type

**ad.samsclass.info**

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

A black box opens, and shows a prompt containing **YOURNAME**, as shown below.



## Saving a Screen Image

Make sure you can see **YOURNAME** in the PuTTY window, as shown above.

Press the **PrintScrn** key to copy the whole desktop to the clipboard.

**YOU MUST SUBMIT A FULL-SCREEN IMAGE FOR FULL CREDIT!**

Paste the image into Paint.

Save the document with the filename "**YOUR NAME Proj 8a2**", replacing "YOUR NAME" with your real name.

## Turning in your Project

Email the images to **cnit.127sam@gmail.com** with the subject line: **Proj 8a from YOUR NAME**

## Sources

[Backdooring PE Files - Part 1](#)

[Art of Anti Detection 2 – PE Backdoor Manufacturing](#)

<https://github.com/EgeBalci/Cminer>

[https://en.wikipedia.org/wiki/Code\\_cave](https://en.wikipedia.org/wiki/Code_cave)

<http://stackoverflow.com/questions/787100/what-is-a-code-cave-and-is-there-any-legitimate-use-for-one>

[The Beginners Guide to Codecaves](#)

[Reversing with immunity debugger](#)

