

PMA 401. Simple EXE Hacking with Ollydbg

What You Need

A Windows 2016 Server machine, real or virtual. Other Windows versions should also work.

Purpose

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

Downloading OllyDbg

Download OllyDbg 1.10 here:

<http://www.ollydbg.de/download.htm>

Right-click the file and click **Extract**, "Extract All...".

Double-click the red icon to launch it.

Downloading HashCalc

Download HashCalc here:

<https://www.slavasoft.com/hashcalc/>

Unzip it and double-click the **setup** to install it. Install it with the default options.

Task 1: Target EXE Recon

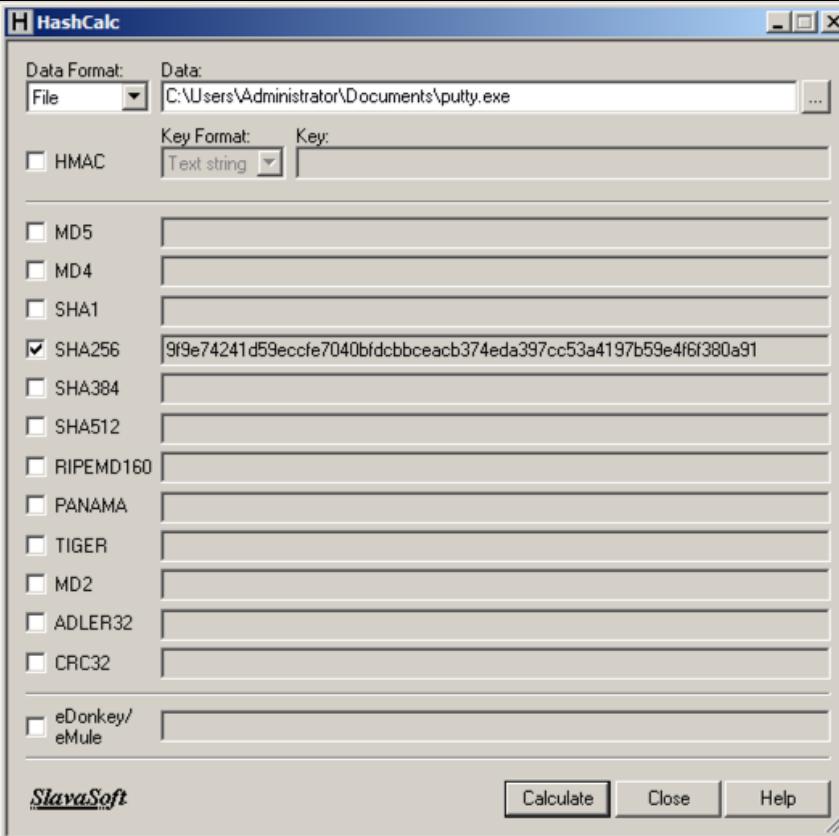
Get putty.exe

Download putty here:

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

Verifying the SHA256 Hash

Run Hashcalc on **putty.exe** and confirm that the SHA256 value matches the value shown below.



Running Putty

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

If PuTTY won't start, right-click it, click **Properties**, and click **Unblock**.



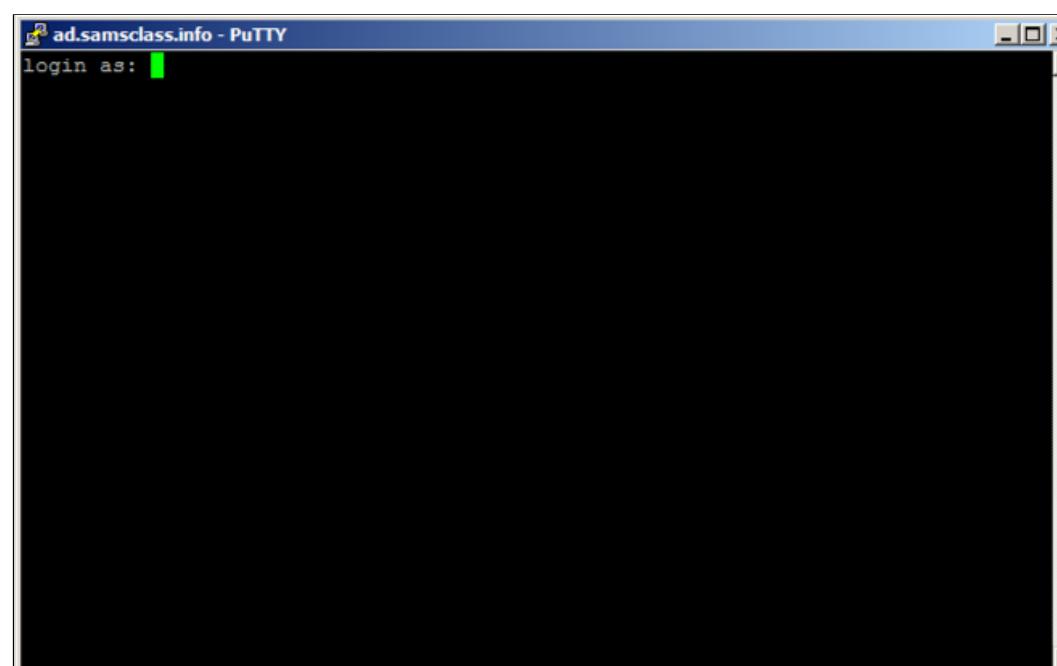
In the "Host Name (or IP address)" box, type

ad.samsclass.info

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

If a "PuTTY Security Alert" box pops up, click **Yes**.

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 Ollydbg

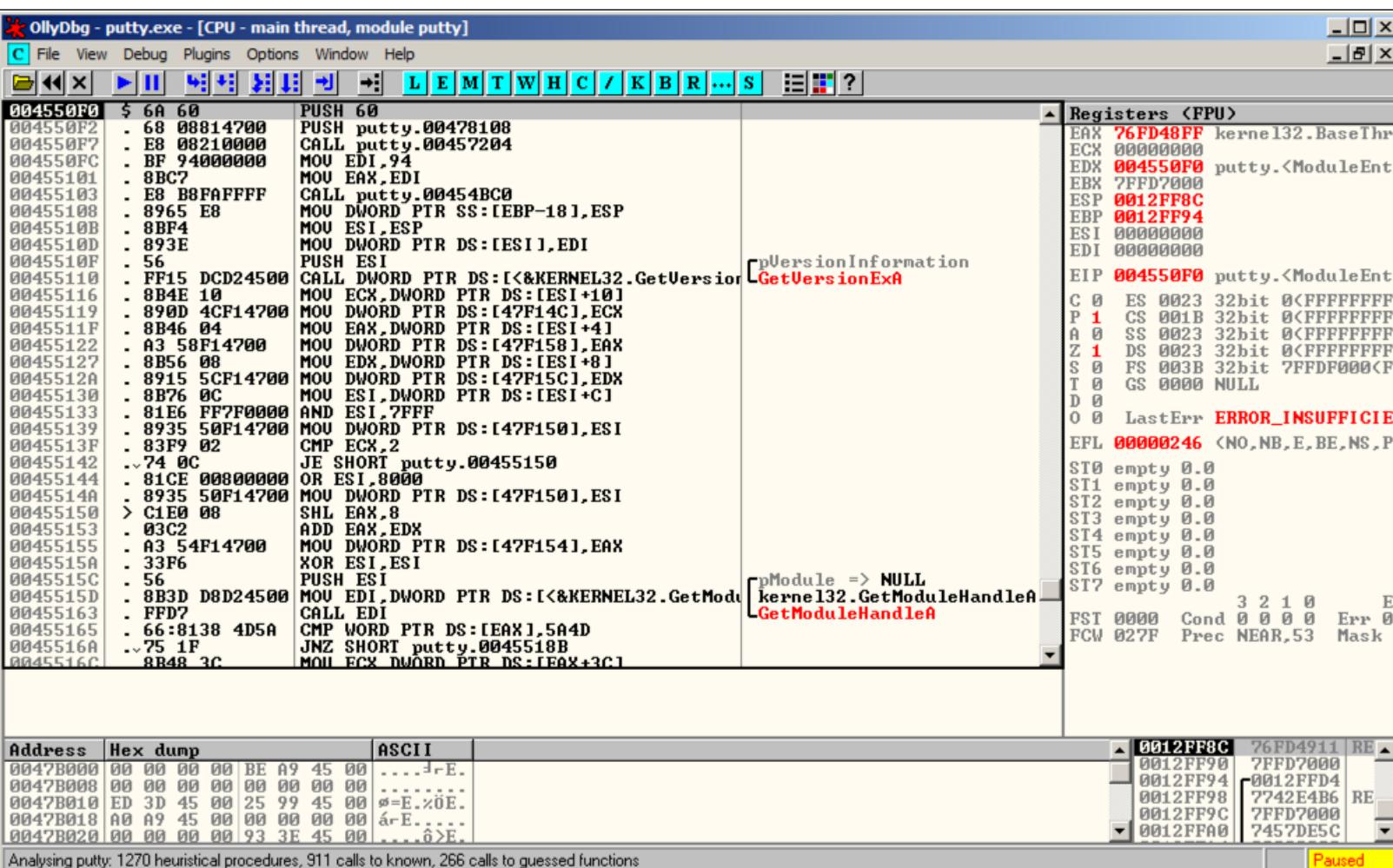
Launch OllyDbg.

If messages ask about deleting old DLLs click **Yes**.

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

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

(If you are using a 64-bit Windows system, the assembly code will look different from the image below until you execute the first Run command. Don't worry about that now.)



Ollydbg 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.

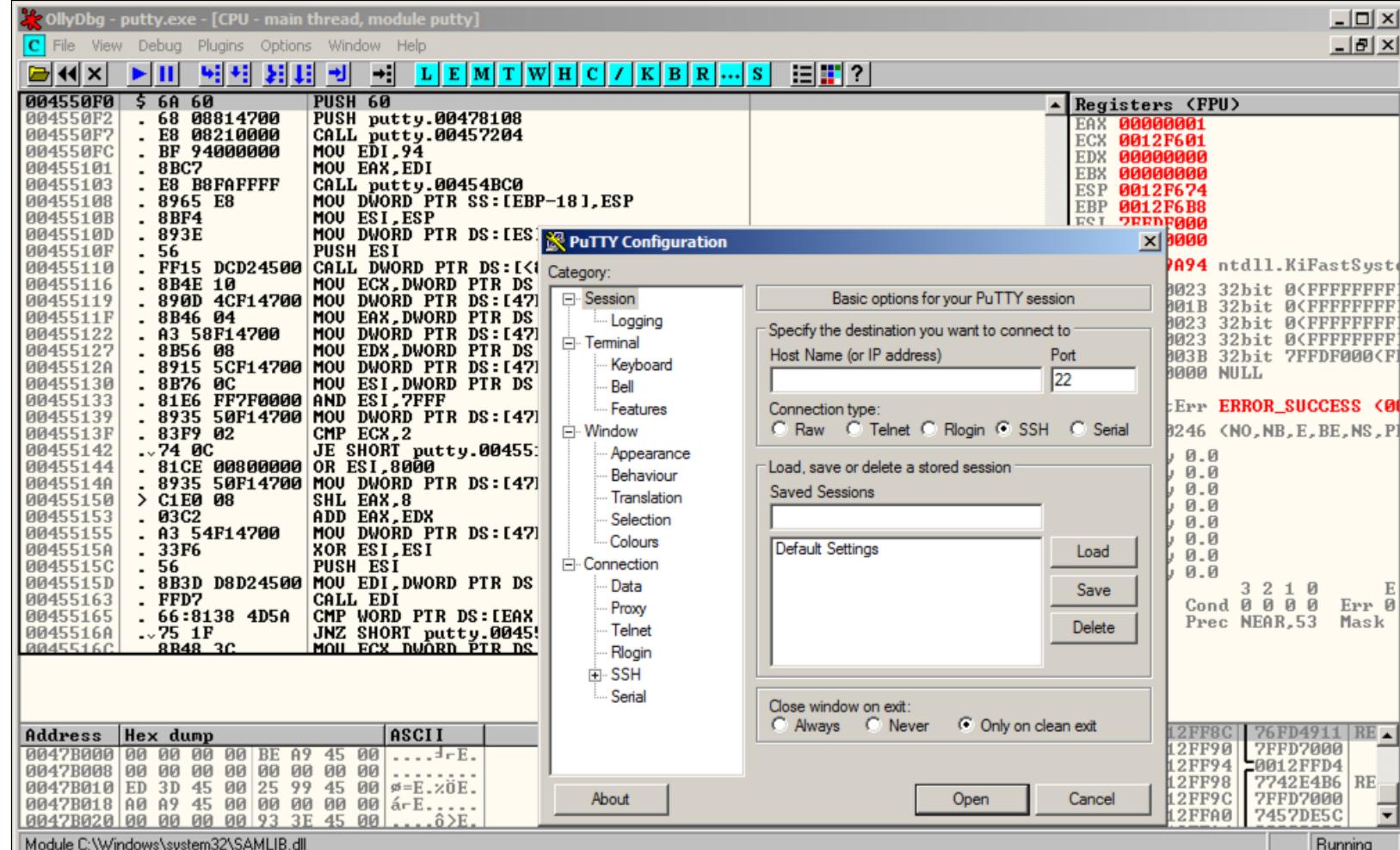
When you load a program into Ollydbg, it starts in a "Paused" state, with the Assembly Code window showing the first instruction.

Running Putty in Ollydbg

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

If the lower-right corner of OllyDbg still shows a "Paused" message, click **Debug, Run** again.

A Putty window opens, but it's behind the Olly window. At the bottom of the screen, in the taskbar, click the "**PuTTY Configuration**" button to bring the PuTTY window to the front, as shown below.



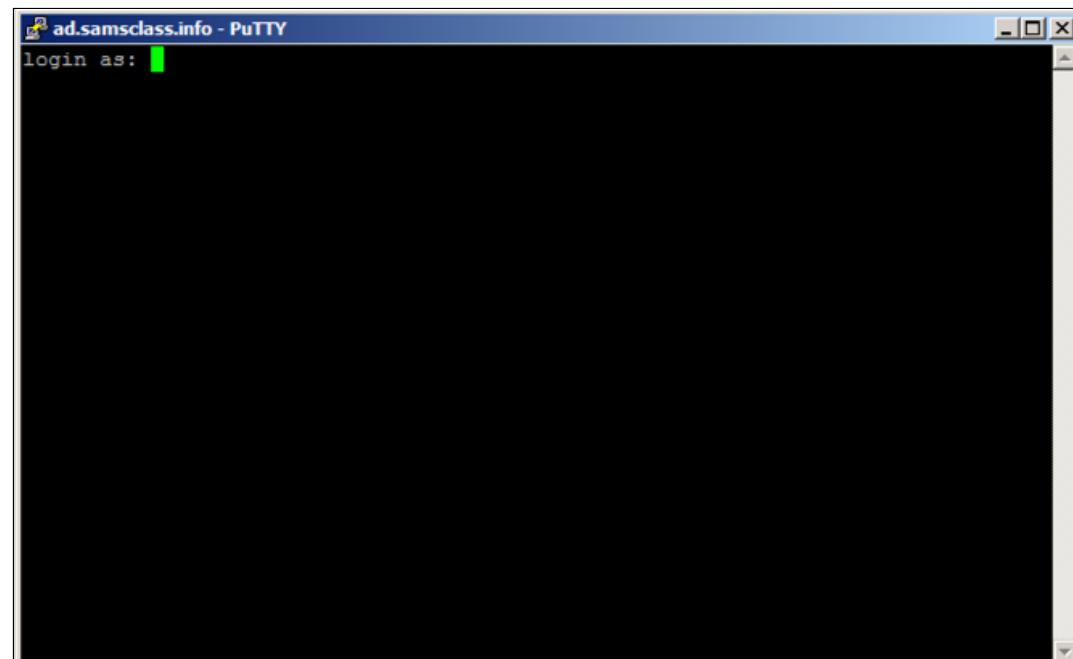
Module C:\Windows\system32\SAMLIB.dll Running

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 Ollydbg, so we can modify its execution.

Finding the "login as" Code

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

OllyDbg - putty.exe - [CPU - main thread, module putty]

File View Debug Plugins Options Window Help

Registers (FPU)

EAX	76FD48FF	kernel32.BaseThre
ECX	00000000	
EDX	004550F0	putty.<ModuleEnt
EBX	7FFDB000	
ESP	0012FF8C	
EBP	0012FF94	
ESI	00000000	
EDI	00000000	
EIP	004550F0	putty.<ModuleEnt
C	0	ES 0023 32bit 0<FFFFFFFFFF
P	1	CS 001B 32bit 0<FFFFFFFFFF
A	0	SS 0023 32bit 0<FFFFFFFFFF
Z	1	DS 0023 32bit 0<FFFFFFFFFF
S	0	FS 003B 32bit 7FFDF000<FI
		00 NULL

Stack Dump

GetVersionInformation

GetVersionExA

Name (label) in current module Ctrl+N

Name in all modules

Backup

Copy

Binary

Assemble Space

Label :

Comment ;

Breakpoint

Hit trace

Run trace

Go to

Follow in Dump

View call tree Ctrl+K

Search for

All referenced text strings

User-defined

User-defined content

Address Hex dump ASCII

0047B000	00 00 00 00 BE A9 45 00E.
0047B008	00 00 00 00 00 00 00 00
0047B010	ED 3D 45 00 25 99 45 00	g=E.ZOE.
0047B018	A0 A9 45 00 00 00 00 00	árE....
0047B020	00 00 00 00 93 3E 45 00ô>E.

Program entry point

FF8C 76FD4911 RE

0012FF90 7FFDB000

0012FF94 0012FFD4

0012FF98 7742E4B6 RE

0012FF9C 7FFDB000

0012FFA0 744B831B

Paused

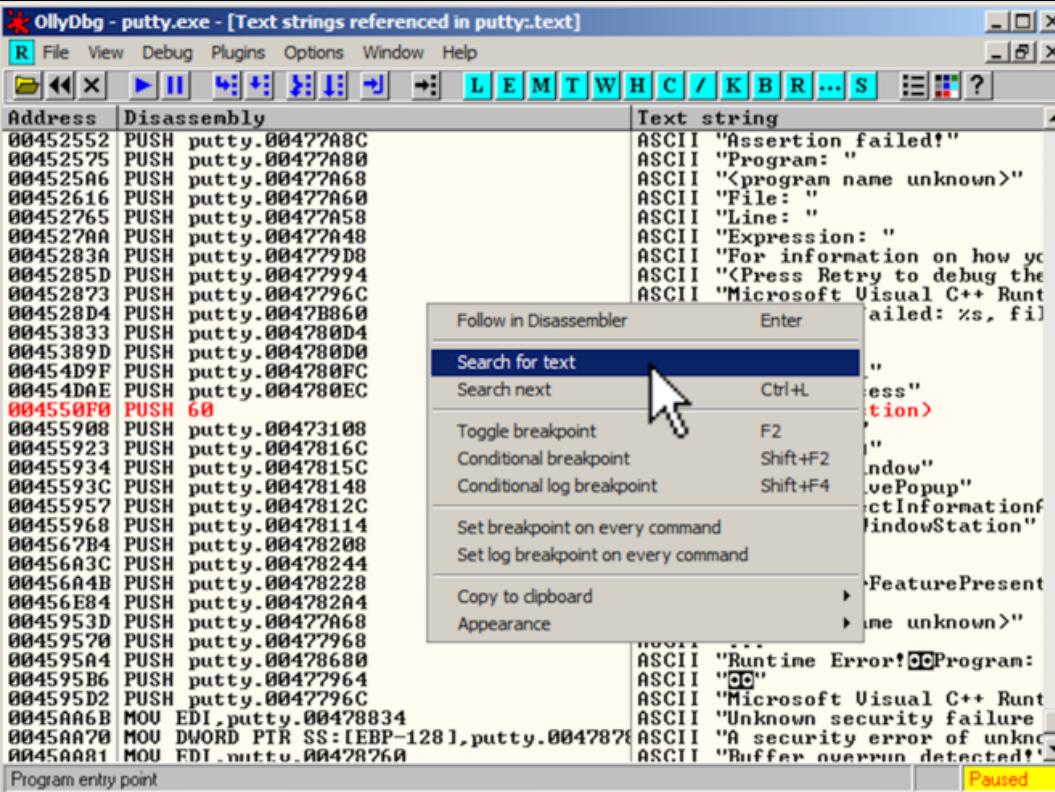
A "Text strings referenced in putty:.text" window opens, showing all the strings in the program.

Troubleshooting

If the title of this window does not contain "putty", but some library such as "ntdll", click "Debug", "Restart" and run Putty again.

To make this text easier to read, right-click, point to **Appearance**, **Font**, and click "**OEM Fixed Font**".

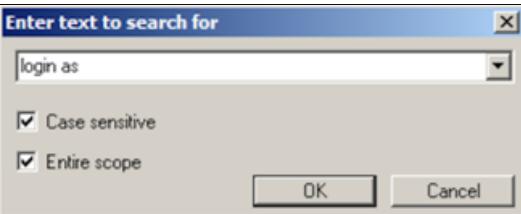
Right-click in that window, and click "**Search for text**", as shown below.



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

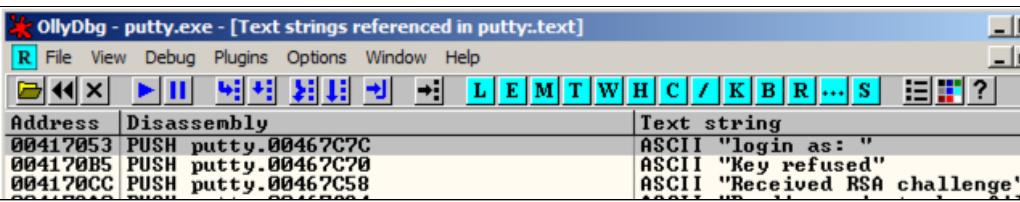
login as

as shown below. Check the "Entire scope" box.



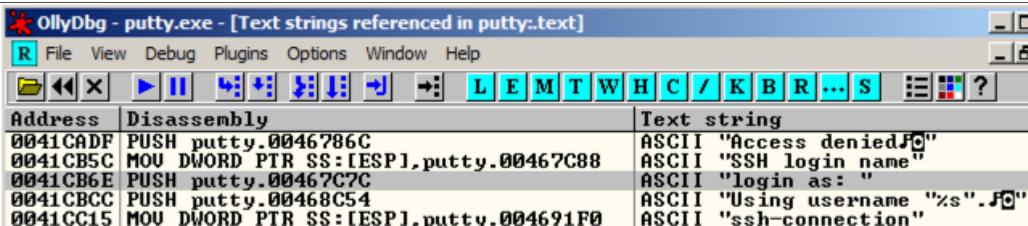
Click **OK**.

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



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

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



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

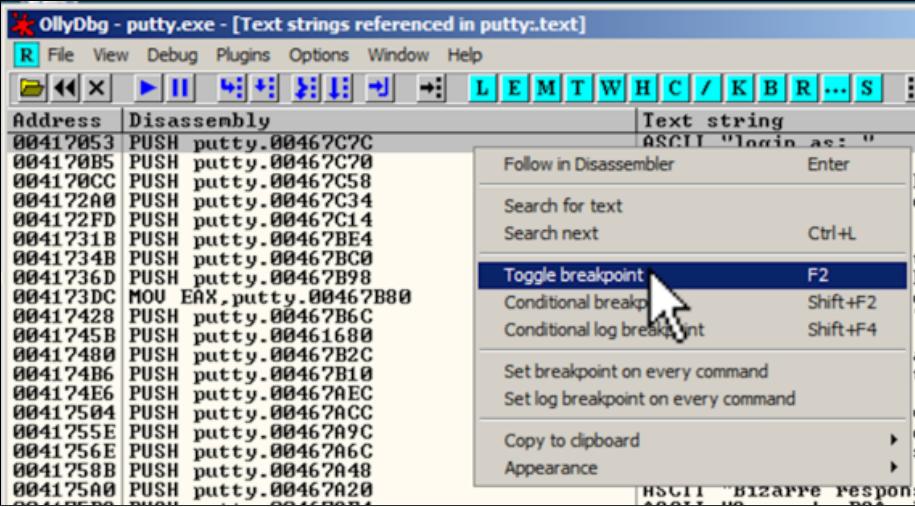
A message appears at the bottom of the window saying "Item not found". There are only two commands in the program that use this string.

Using Breakpoints

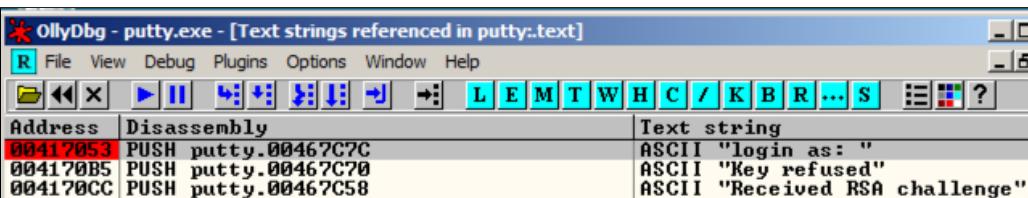
We'll set breakpoints at those instructions to see which one is used when logging in to an SSH server.

In the "Text strings referenced in putty:.text" window, right-click again, and click "**Search text**". In the "Enter text to search for" box, click **OK**.

The instruction at 00417053 appears again. Right-click this instruction and click "**Toggle breakpoint**", as shown below.

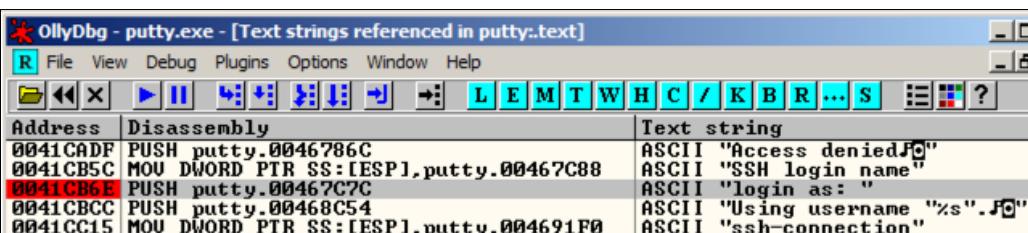


The address turns red, as shown below, to indicate that there's a breakpoint here.



Right-click again, and click "**Search next**". The instruction at address 0041CB6E appears. Right-click it and click "**Toggle breakpoint**".

The address turns red, as shown below.



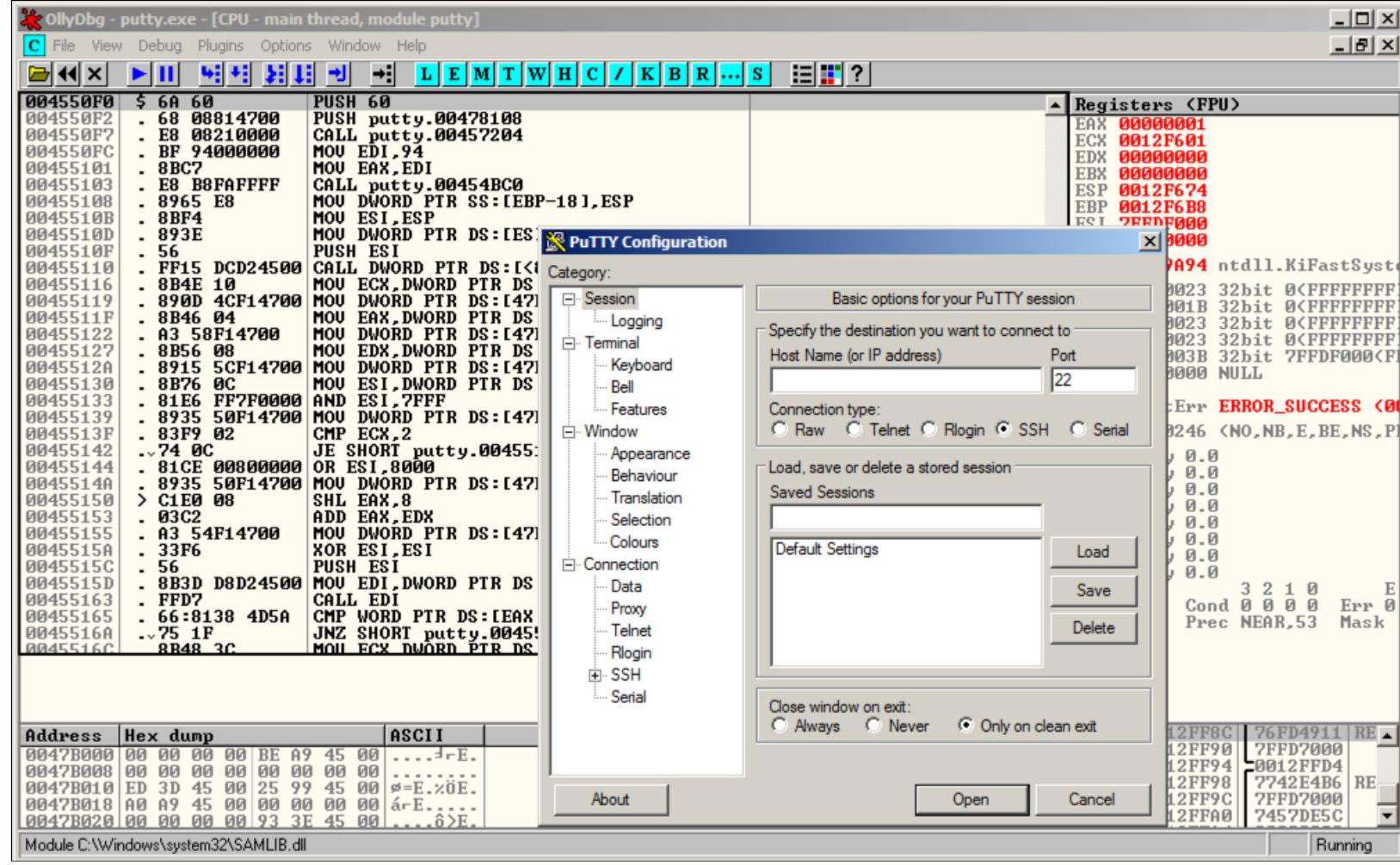
In Ollydbg, from the menu bar, click **Debug, Restart**.

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

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

If the lower-right corner of OllyDbg still shows a "Paused" message, click **Debug, Run** again.

A Putty window opens. Bring it to the front, as shown below.

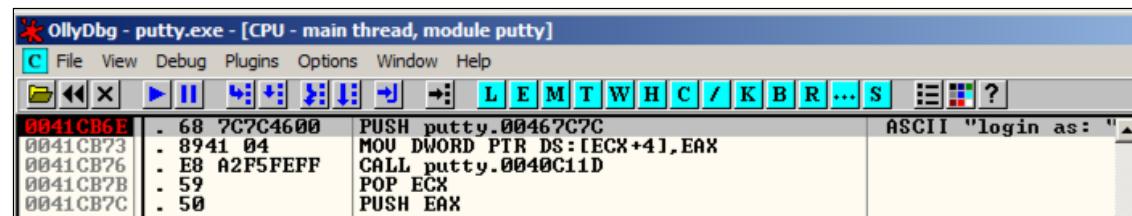


Click in the Putty window. In the "Host Name (or IP address)" box, type

ad.samsclass.info

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

A black window opens and closes quickly, and 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 Breakpoints

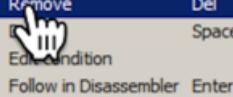
We don't need the breakpoints any more, so we'll remove them.

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

A "Breakpoints" window opens, showing two breakpoints.

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

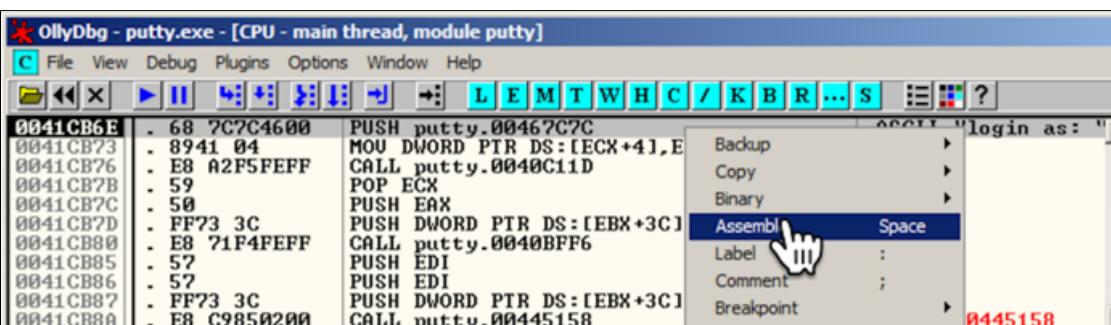
Breakpoints			
Address	Module	Active	Disassembly
00417053	putty	Always	
0041CB6E	putty	Always	



Repeat the process to remove the other breakpoint. Close the "Breakpoints" window.

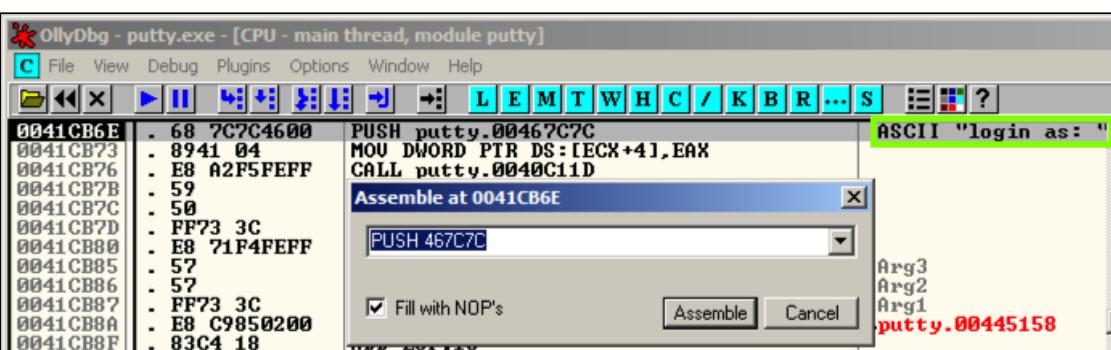
Removing One Letter From the Message

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

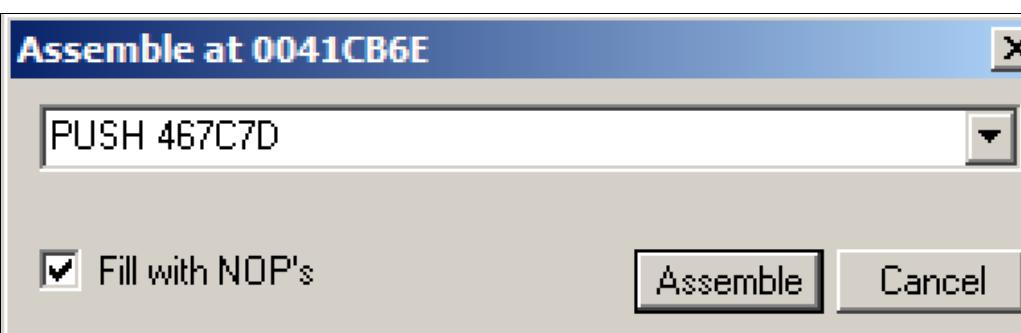


An "Assemble at 0041CB6E" box appears, as shown below.

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 "I" in the ASCII string "login as: ", as shown on the right side of the instruction line, outlined in green in the image below.



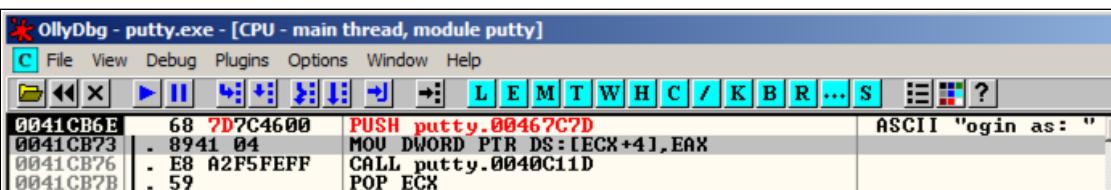
In the "Assemble at 0041CB6E" box, change the last character to **D**, as shown below. This will move the pointer from the "I" to the "o" in the string "login as: ".



Click the **Assemble** button.

Click the **Cancel** button.

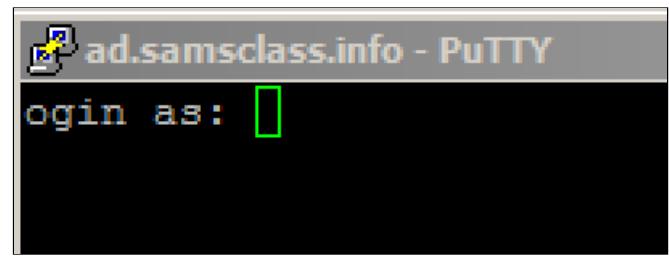
The message on the right now says "ogin as: ", as shown below.



Running the Modified Program

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

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

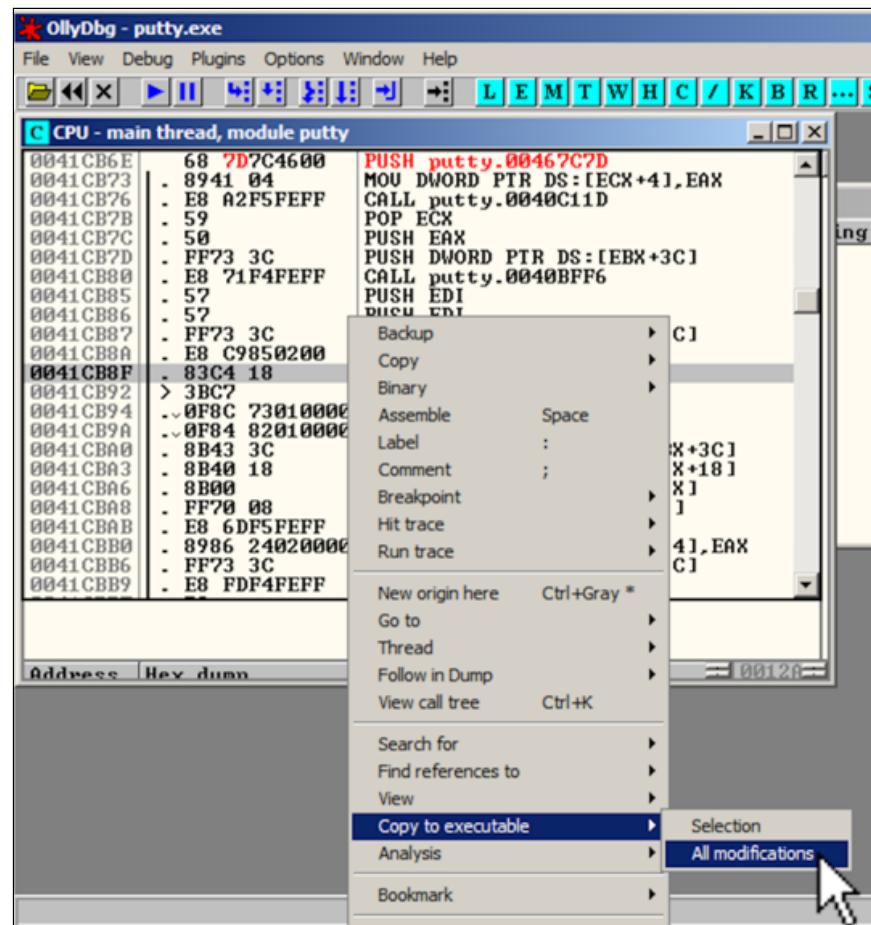


When I did it, an error box also popped up saying "Server unexpectedly closed network connection". If that happens, just close the error box.

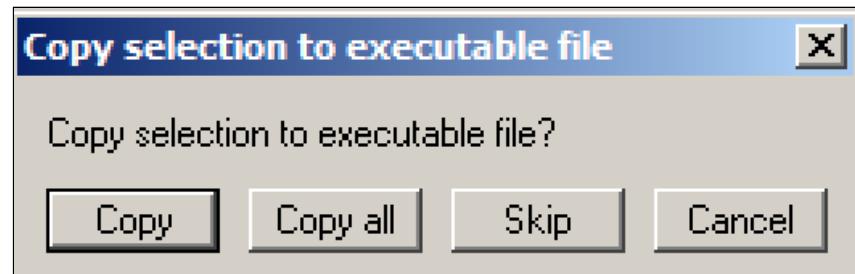
Saving the Modified .text Section

We have now changed an assembly language instruction; all executable code is in the **.text** section of the file.

In Ollydbg, in the top left pane of the CPU window, right-click, point to "**Copy to Executable**", and click "**All modifications**", as shown below.



A "Copy selection to executable file" box pops up, as shown below. Click the "Copy all" button.



A new window pops up, with a title ending in "putty.exe", as shown below.

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

D File C:\Users\Administrator\Downloads\putty.exe

0001CB6E	68 7D7C4600	PUSH 467C7D
0001CB73	8941 04	MOU DWORD PTR DS:[ECX+4], EAX
0001CB76	E8 A2F5FEFF	CALL 0000C11D
0001CB7B	59	POP ECX
0001CB7C	50	PUSH EAX
0001CB7D	FF73 3C	PUSH DWORD PTR DS:[EBX+3C]
0001CB80	E8 ?1F4FEFF	CALL 0000BFFF6
0001CB85	57	PUSH EDI
0001CB86	57	PUSH EDI
0001CB87	FF73 3C	PUSH DWORD PTR DS:[EBX+3C]
0001CB8A	E8 C9850200	CALL 00045158
0001CB8F	83C4 18	ADD ESP, 18
0001CB92	3BC7	CMP EAX, EDI
0001CB94	0F8C 73010000	JL 0001CD0D
0001CB9A	0F84 82010000	JE 0001CD22

Save the file as **puttymod.exe**.

Running the Modified EXE

Close Ollydbg.

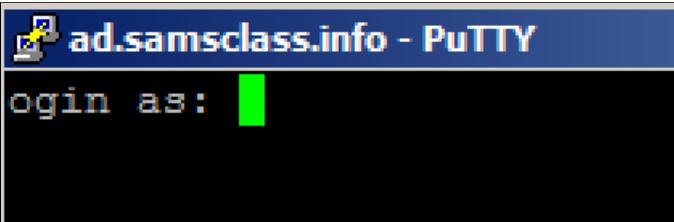
Double-click **puttymod.exe**.

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

ad.samsclass.info

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

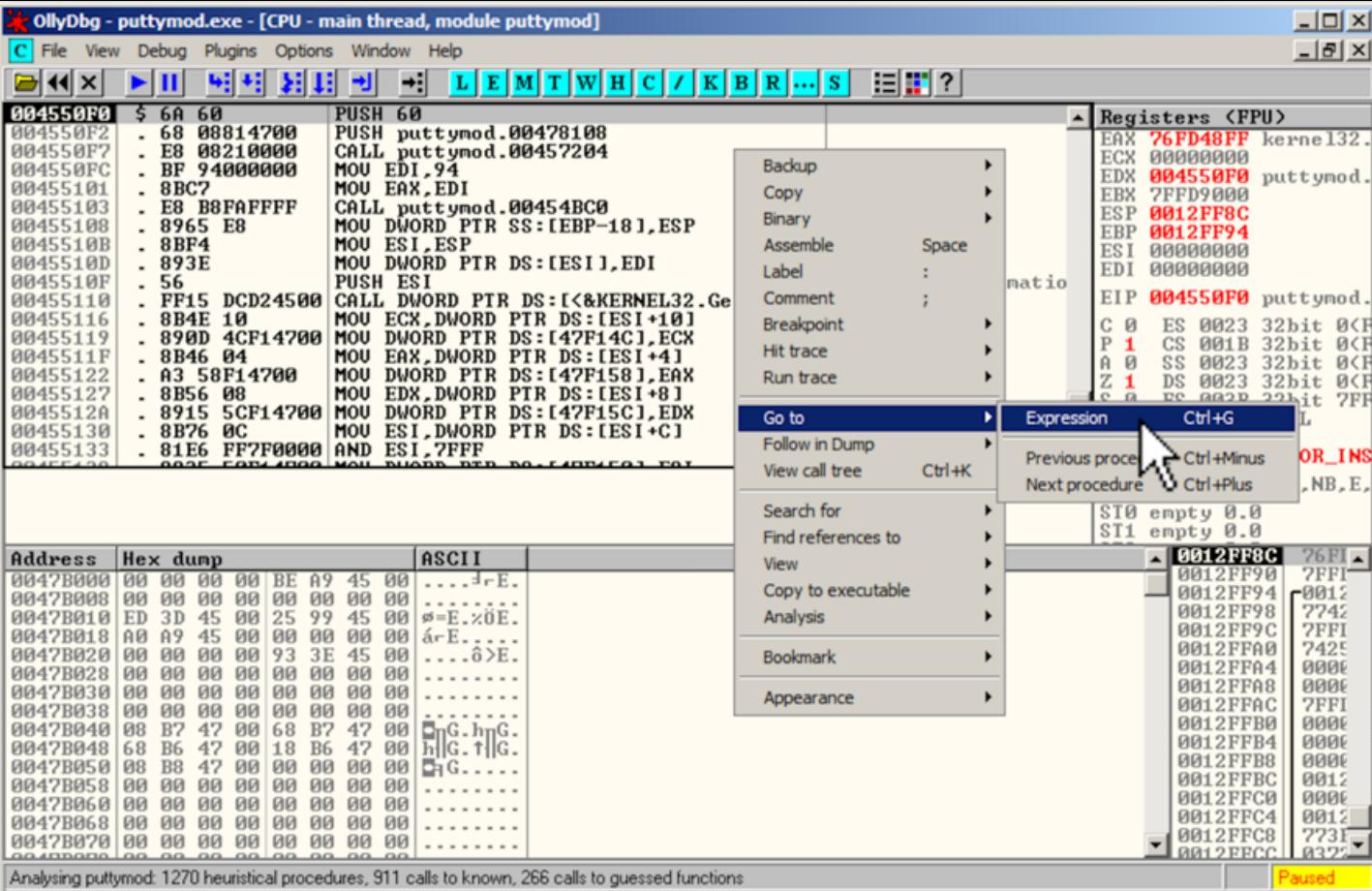
A black box opens, and shows a "login as:" prompt, as shown below.



Modifying the puttymod File

Open Ollydbg and load the **puttymod.exe** file.

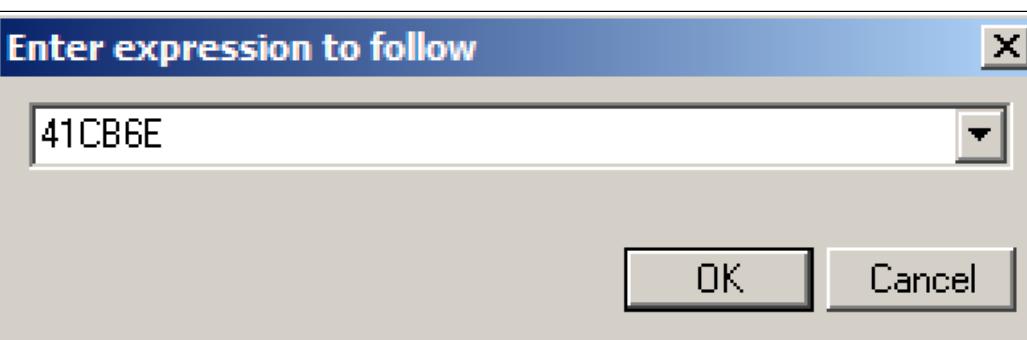
In the top left pane of the CPU window, right-click, point to "**Go to**", and click **Expression**, as shown below.



In the "Enter expression to follow" box, enter

41CB6E

as shown below. Click **OK**.



Changing the Login Message

In the top left pane of the CPU window, right-click **00467C7D**, as shown below. Point to "Follow in Dump" and click "Immediate constant".

OllyDbg - puttymod.exe - [CPU - main thread, module puttymod]

File View Debug Plugins Options Window Help

Registers <FPU>

EAX	76FD48FF	kernel32.
ECX	00000000	
EDX	004550F0	puttymod.
EBX	7FFD9000	
ESP	0012FF8C	
EBP	0012FF94	
ESI	00000000	
EDI	00000000	
EIP	004550F0	puttymod.
C	0	ES 0023 32bit 0<F
P	1	CS 001B 32bit 0<F
A	0	SS 0023 32bit 0<F
Z	1	DS 0023 32bit 0<F
S	0	FS 003B 32bit 7FF
T	0	GS 0000 NULL
D	0	LastErr ERROR_INS
O	0	EFL 00000246 <NO,NB,E,

ST0 empty 0.0
ST1 empty 0.0

0012FF8C 76F1

0012FF90	7FF1
0012FF94	0012
0012FF98	7742
0012FF9C	7FF1
0012FFA0	7425
0012FFA4	000E
0012FFA8	000E
0012FFAC	7FF1
0012FFB0	000E
0012FFB4	000E
0012FFB8	000E
0012FFBC	0012
0012FFC0	000E
0012FFC4	0012
0012FFC8	7731
0012FECC	0322

ASCII "ogin as:"

Follow in Dump Selection Immediate constant

View call tree Ctrl+K

Search for

Find references to

View

Copy to executable

Analysis

Bookmark

Appearance

Address Hex dump ASCII

0047B000	00 00 00 00 BE A9 45 003r.E.
0047B008	00 00 00 00 00 00 00 00	
0047B010	ED 3D 45 00 25 99 45 00	g=E..z0E.
0047B018	A0 A9 45 00 00 00 00 00	á=E.....
0047B020	00 00 00 00 93 3E 45 006>E.
0047B028	00 00 00 00 00 00 00 00
0047B030	00 00 00 00 00 00 00 00
0047B038	00 00 00 00 00 00 00 00
0047B040	08 B7 47 00 68 B7 47 00	h G.h G.
0047B048	68 B6 47 00 18 B6 47 00	h G.↑ G.
0047B050	08 B8 47 00 00 00 00 00	h G.....
0047B058	00 00 00 00 00 00 00 00
0047B060	00 00 00 00 00 00 00 00
0047B068	00 00 00 00 00 00 00 00
0047B070	00 00 00 00 00 00 00 00

Analysing puttymod: 1270 heuristical procedures, 911 calls to known, 266 calls to guessed functions

Paused

The Hex Dump pane, in the lower left, shows the text "ogin as:", as shown below.

In the Hex Dump pane, highlight "ogin as:", as shown below. Right-click the highlighted text. Point to **Binary**. Click **Edit**.

OllyDbg - puttymod.exe - [CPU - main thread, module puttymod]

File View Debug Plugins Options Window Help

Registers <FPU>

EAX	76FD48FF	kernel32.
ECX	00000000	
EDX	004550F0	puttymod.
EBX	7FFD9000	
ESP	0012FF8C	
EBP	0012FF94	
ESI	00000000	
EDI	00000000	
EIP	004550F0	puttymod.
C	0	ES 0023 32bit 0<F
P	1	CS 001B 32bit 0<F
A	0	SS 0023 32bit 0<F
Z	1	DS 0023 32bit 0<F
S	0	FS 003B 32bit 7FF
T	0	GS 0000 NULL
D	0	LastErr ERROR_INS
O	0	EFL 00000246 <NO,NB,E,

ST0 empty 0.0
ST1 empty 0.0

0012FF8C 76F1

0012FF90	7FF1
0012FF94	0012
0012FF98	7742
0012FF9C	7FF1
0012FFA0	7425
0012FFA4	000E
0012FFA8	000E
0012FFAC	7FF1
0012FFB0	000E
0012FFB4	000E
0012FFB8	000E
0012FFBC	0012
0012FFC0	000E
0012FFC4	0012
0012FFC8	7731
0012FECC	0322

ASCII "ogin as:"

Follow in Dump Selection Immediate constant

View call tree Ctrl+K

Search for

Find references to

View

Copy to executable

Analysis

Bookmark

Appearance

Address Hex dump ASCII

00467C7D	6F 67 69 6E 20 61 73 3A	ogin as:
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 25 63 63 65	e..Succe
00467C9D	73 73 66 75 6C 6C 79 20	ssfully
00467CA5	73 74 61 72 74 65 64 20	started
00467CAD	65 6E 63 72 79 70 74 69	encrypti
00467CBS	6F 6E 00 45 6E 63 72 79	on.Encry
00467CBD	70 74 69 6F 6E 20 6E 6F	ption no

Backup

Copy

Binary

Label

Breakpoint

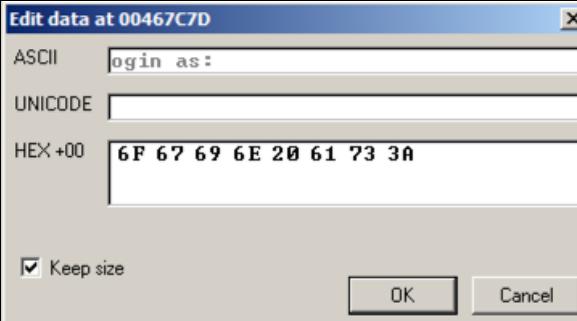
Search for

Edit Ctrl+E

Fill with 0's

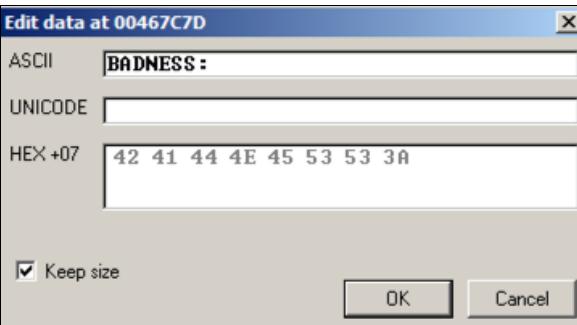
Fill with FF's

An "Edit data at 00467C7D" box opens, as shown below.



Click in the ASCII field, press Backspace to move back to the start, and overwrite the message with this text, as shown below:

BADNESS



Click **OK**. The modified text appears in red letters in the Dump, as shown below.

Address	Hex dump	ASCII
00467C7D	42 41 44 4E 45 53 53 3A	BADNESS:
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	23 73 66 75 6C 6C 79 20	ssfully
00467CA5	73 74 61 72 74 65 64 20	started

Saving the Modified ASCII Text

In Ollydbg, in the lower left "Dump" pane of the CPU window, right-click and click "**Copy to executable file**", as shown below.

OllyDbg - puttymod.exe - [CPU - main thread, module puttymod]

File View Debug Plugins Options Window Help

LEMTWH C / KBR...S

0041CB6E - 68 ?D?C4600 PUSH puttymod.0046?C?D
 0041CB73 - 8941 04 MOU DWORD PTR DS:[ECX+4],EAX
 0041CB76 - E8 A2F5FEFF CALL puttymod.0040C11D
 0041CB7B - 59 POP ECX
 0041CB7C - 50 PUSH EAX
 0041CB7D - FF73 3C PUSH DWORD PTR DS:[EBX+3C]
 0041CB80 - E8 71F4FEFF CALL puttymod.0040BFFF6
 0041CB85 - 57 PUSH EDI
 0041CB86 - 57 PUSH EDI
 0041CB87 - FF73 3C PUSH DWORD PTR DS:[EBX+3C]
 0041CB8A - E8 C9850200 CALL puttymod.00445158
 0041CB8F - 83C4 18 ADD ESP,18
 0041CB92 - > 3BC7 CMP EAX,EDI
 0041CB94 - .v 0F8C 73010000 JL puttymod.0041CD0D
 0041CB9A - .v 0F84 82010000 JE puttymod.0041CD22
 0041CBA0 - . 8B43 3C MOU EAX,DWORD PTR DS:
 0041CBA3 - . 8B40 18 MOU EAX,DWORD PTR DS:
 0041CBA6 - . 8B00 MOU EAX,DWORD PTR DS:
 0041CBA8 - . FF70 08 PUSH DWORD PTR DS:[EA
 0046?C?D=puttymod.0046?C?D <ASCII "BADNESS: "

Backup
 Undo selection Alt+BkSp
 Copy
 Binary
 Label :
 Breakpoint
 Search for
 Find references Ctrl+R
 View executable file

Copy to executable file

Address Hex dump ASCII

0046?C?D	42 41 44 4E 45 53 53 3A	BADNESS:
0046?C85	20 00 00 53 53 48 20 6C	.SSH 1
0046?C8D	6F 67 69 6E 20 6E 61 6D	ogin nam
0046?C95	65 00 00 53 75 63 63 65	e..Succe
0046?C9D	73 73 66 75 6C 6C 79 20	ssfully
0046?CA5	73 74 61 72 74 65 64 20	started
0046?CAD	65 6E 63 72 79 70 74 69	encrypti
0046?CB5	6F 6E 00 45 6E 63 72 79	on.Encry
0046?CBD	70 74 69 6F 6E 20 6E 6F	ption no
0046?CC5	74 20 73 75 63 63 65 73	t succes
0046?CCD	73 66 75 6C 6C 79 20 65	sf fully e
0046?CD5	6E 61 62 6C 65 64 00 46	nabled.F
0046?CDD	61 69 6C 65 64 20 74 6F	ailed to
0046?CE5	20 72 65 61 64 20 53 53	read SS
0046?CED	48 2D 31 20 70 75 62 6C	H-1 publ
0046?CF5	69 63 20 6B 65 79 73 20	ic keys
0046?CFD	66 72 6F 6D 20 70 75 62	from pub
0046?D05	6C 69 63 20 6B 65 79 20	lic key
0046?D0D	70 61 63 6B 65 74 00 53	packet.S
0046?D15	53 48 2D 31 20 70 75 62	SH-1 pub

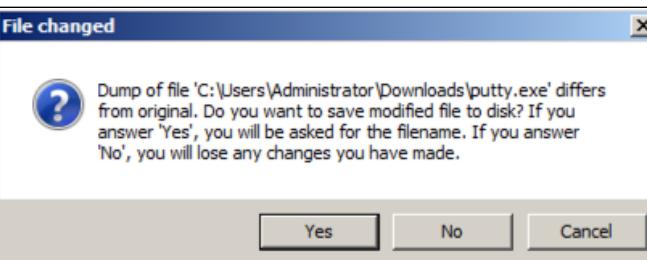
Analysing puttymod: 1270 heuristical procedures, 911 calls to known, 266 calls to guessed functions

A box with a long title ending in "puttymod.exe" appears showing the modified file, as shown below. Click the X in the top right of that box to close it.

D File C:\Users\Administrator\Downloads\putty....

0006?C?D 42 41 44 4E 45 53 53 3A BADNESS:
 0006?C85 20 00 00 53 53 48 20 6C ..SSH 1
 0006?C8D 6F 67 69 6E 20 6E 61 6D ogin nam
 0006?C95 65 00 00 53 75 63 63 65 e..Succe
 0006?C9D 73 73 66 75 6C 6C 79 20 ssfully
 0006?CA5 73 74 61 72 74 65 64 20 started
 0006?CAD 65 6E 63 72 79 70 74 69 encrypti
 0006?CB5 6F 6E 00 45 6E 63 72 79 on.Encry
 0006?CBD 70 74 69 6F 6E 20 6E 6F option no
 0006?CC5 74 20 73 75 63 63 65 73 t succes
 0006?CCD 73 66 75 6C 6C 79 20 65 sf fully e
 0006?CD5 6E 61 62 6C 65 64 00 46 nabled.F
 0006?CDD 61 69 6C 65 64 20 74 6F ailed to
 0006?CE5 20 72 65 61 64 20 53 53 read SS
 0006?CED 48 2D 31 20 70 75 62 6C H-1 publ
 0006?CF5 69 63 20 6B 65 79 73 20 ic keys
 0006?CFD 66 72 6F 6D 20 70 75 62 from pub
 0006?D05 6C 69 63 20 6B 65 79 20 lic key
 0006?D0D 70 61 63 6B 65 74 00 53 packet.S
 0006?D15 53 48 2D 31 20 70 75 62 SH-1 pub

A "File changed" box appears, as shown below. Click Yes.



Save the file as "puttymod2.exe".

Running the Modified EXE

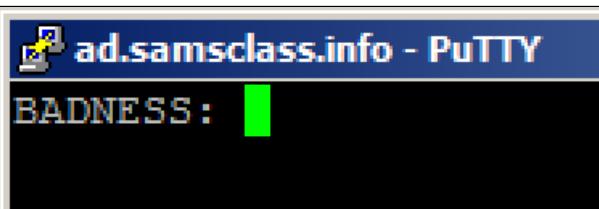
Close Ollydbg.

Double-click puttymod2.exe.

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

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

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



PMA 401.1: Calculating the Hash (20 pts)

Calculate the CRC32 hash of **puttymod2.exe**

The flag is that hash like this: **07b01710**

Patching More EXEs

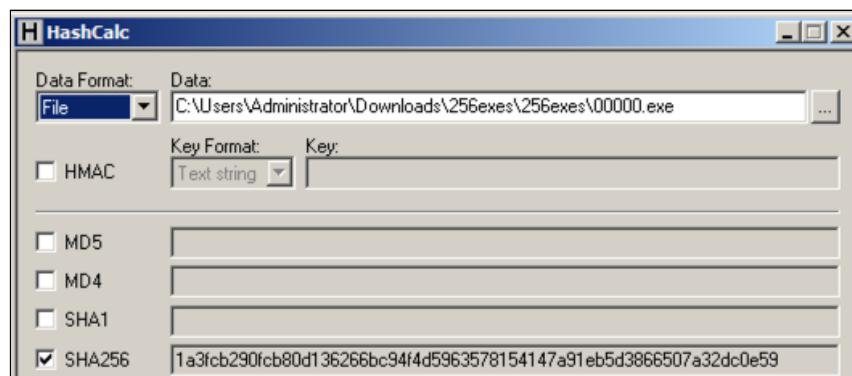
Getting the Files

You need several files to examine. Download them with these links:

- [00000.exe](#)
- [3EXEs.zip](#)
- [easy.zip](#)
- [256exes.zip](#)

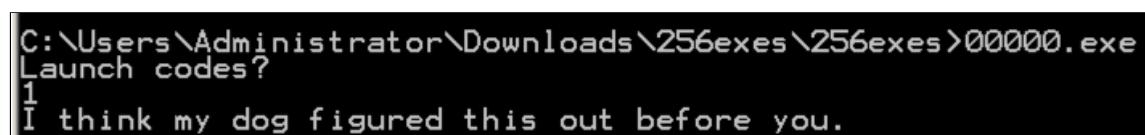
Checking the Hash

Calculate the SHA256 hash of the **00000.exe** file. It should match the value shown below.



Running 00000.exe

Run **00000.exe** in a Command Prompt. It asks for a "Launch code" and if you guess wrong, it insults you, as shown below.



Examining the EXE with Ollydbg

Open the file in OllyDbg, as shown below.

If you are using a 64-bit Windows system, click **Debug**, **Run** once to get to the start of the 32-bit code, as shown below.

In the assembly code pane, look at the rightmost column, and you can easily see what the program does; it prints out "Launch codes?", reads in a decimal number (%d), and then chooses to print either a winning message with a result, or an insult.

The choice is performed by two instructions: CMP (Compare) and JNZ (Jump if Not Zero), outlined in green in the image below.

```
00402006 $ 68 5E304000 PUSH 00000.0040305E
0040200B . FF15 44104000 CALL DWORD PTR DS:[&msvcrt.puts]
00402011 . 58
00402012 . 68 6C304000 PUSH 00000.0040306C
00402017 . 68 04304000 PUSH 00000.00403004
0040201C . FF15 48104000 CALL DWORD PTR DS:[&msvcrt.scprintf]
00402022 . 83C4 08 ADD ESP,8
00402025 . A1 00304000 MOV EAX,DWORD PTR DS:[403000]
0040202A . B9 EDA7A8A1 MOV ECX,A1A8A7ED
0040202F . E8 CFFFFFFF CALL 00000.00402003
00402034 . 3B05 6C304000 CMP EAX,DWORD PTR DS:[40306C]
0040203A . 75 1E JNZ SHORT 00000.0040205A
0040203C . 8A0D 07304000 MOU CL,BYTE PTR DS:[403007]
00402042 . D3F8
00402044 . 25 FF000000 AND EAX,0FF
00402049 . 50 PUSH EAX
0040204A . 68 34304000 PUSH 00000.00403034
0040204F . FF15 4C104000 CALL DWORD PTR DS:[&msvcrt.printf]
00402055 . 83C4 08 ADD ESP,8
00402058 . EB 0C JMP SHORT 00000.00402066
0040205A > 68 08304000 PUSH 00000.00403008
0040205F . FF15 44104000 CALL DWORD PTR DS:[&msvcrt.puts]
00402065 . 58 POP EAX
```

Replace those instructions with NOP (No Operation), as shown below. Save the patched file.

```
00402006 $ 68 5E304000 PUSH 00000.0040305E
0040200B . FF15 44104000 CALL DWORD PTR DS:[&msvcrt.puts]
00402011 . 58
00402012 . 68 6C304000 PUSH 00000.0040306C
00402017 . 68 04304000 PUSH 00000.00403004
0040201C . FF15 48104000 CALL DWORD PTR DS:[&msvcrt.scprintf]
00402022 . 83C4 08 ADD ESP,8
00402025 . A1 00304000 MOV EAX,DWORD PTR DS:[403000]
0040202A . B9 EDA7A8A1 MOV ECX,A1A8A7ED
0040202F . E8 CFFFFFFF CALL 00000.00402003
00402034 . 90 NOP
00402035 . 90 NOP
00402036 . 90 NOP
00402037 . 90 NOP
00402038 . 90 NOP
00402039 . 90 NOP
0040203A . 90 NOP
0040203B . 90 NOP
0040203C . 8A0D 07304000 MOU CL,BYTE PTR DS:[403007]
00402042 . D3F8
00402044 . 25 FF000000 AND EAX,0FF
00402049 . 50 PUSH EAX
0040204A . 68 34304000 PUSH 00000.00403034
0040204F . FF15 4C104000 CALL DWORD PTR DS:[&msvcrt.printf]
00402055 . 83C4 08 ADD ESP,8
00402058 . EB 0C JMP SHORT 00000.00402066
0040205A > 68 08304000 PUSH 00000.00403008
0040205F . FF15 44104000 CALL DWORD PTR DS:[&msvcrt.puts]
00402065 . 58 POP EAX
```

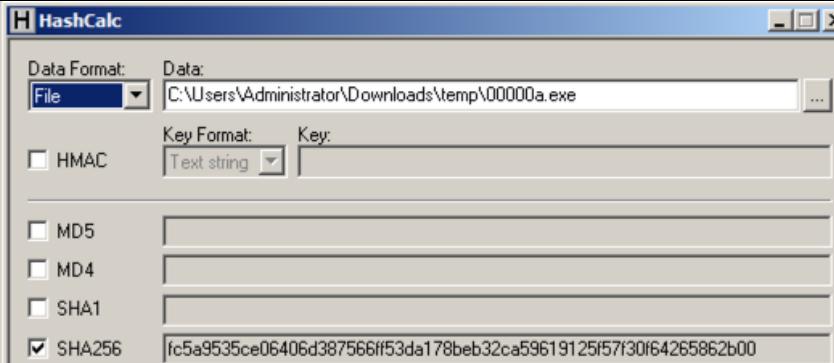
Running the Patched File

The patched file will accept any Launch code, as shown below.

```
C:\Users\Administrator\Downloads>00000a.exe
Launch codes?
1
Wow you got it. Here is the result: (J)
```

Checking the Hash

Calculate the SHA256 hash of the patched file. It should match the value shown below.



PMA 401.2: CRC32 of Patched File (10 pts)

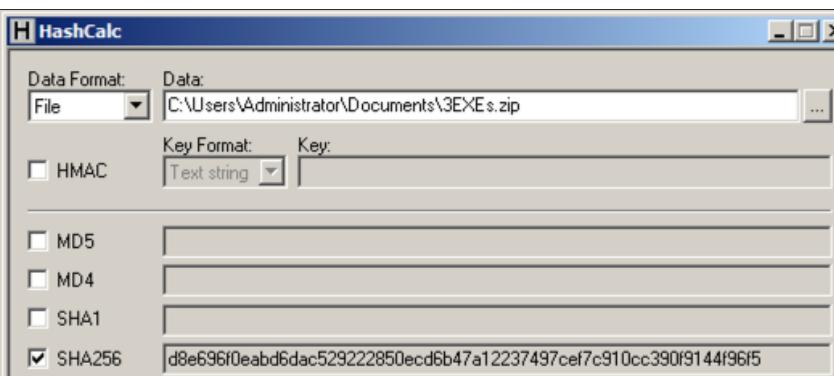
Calculate the CRC32 hash of the patched file.

The flag is that hash like this: **07b01710**

Patching Three EXEs

Checking the Hash

Calculate the SHA256 hash of the **3EXEs.zip** file. It should match the value shown below.



Patch the Files

Unzip the archive. Patch all 3 files so they will accept any input.

Gather the Results

Run the three patched files. Each one returns a single character as a result. Keep the files in alphabetical order, by filename, like this:

- File **00000.exe** Result C
- File **0000a.exe** Result A
- File **000a1.exe** Result T

If those were the results, the answer would be **CAT**

The actual results are different, of course.

PMA 401.3: Three Characters

The flag is the results, three characters like this: **CAT**

Patching 19 EXEs

Getting the EXEs

Unzip the **easy.zip** file. There are 19 EXEs in it.

Goal

Patch all 19 files, run them, and combine the Results to get a 19-character flag.

Hints

There are hints [here](#).

PMA 401.4: Nineteen Characters

The flag is the results, 19 Characters like this: **Impenetrable!Cyber!**

Patching 256 EXEs

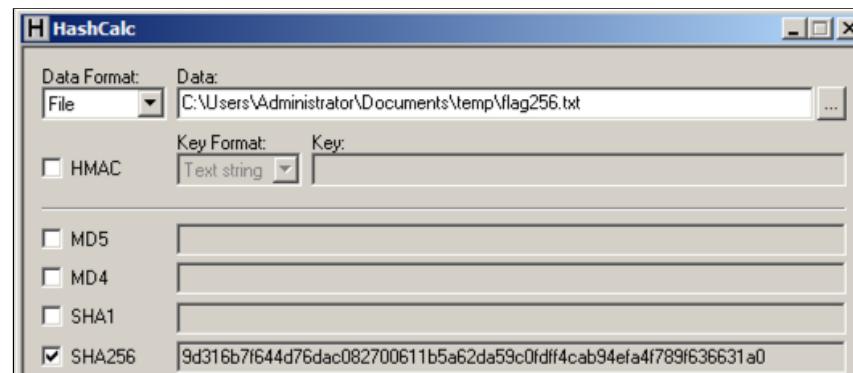
Getting the EXEs

Unzip the **256exes.zip** file. There are 256 EXEs in it.

Goal: Gather the Results

Patch all 256 files and run them. Each file will give you one "Result" character. Gather all those characters into a file 256 bytes long.

Calculate the SHA256 hash of that file. It should match the value shown below.



Calculate the CRC32 of that file to win.

PMA 401.5: CRC32

The flag is the CRC32 hash of the 256-byte file.

Credit

This is based on the 67k Challenge from EasyCTF 2017.

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
<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 Ollydbg debugger](#)
[Ollydbg 'Copy all modifications to executable' doesn't copy all modifications](#)

References to Win 2008 removed 2-23-21