**CSE 2010 SECURE CODING**

**LAB SLOT –L23+L24**

**NAME- K V P S GANESH M**

**REGISTRATION NUMBER-19BCN7073**

**LAB EXPERIMENT 10**

**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**

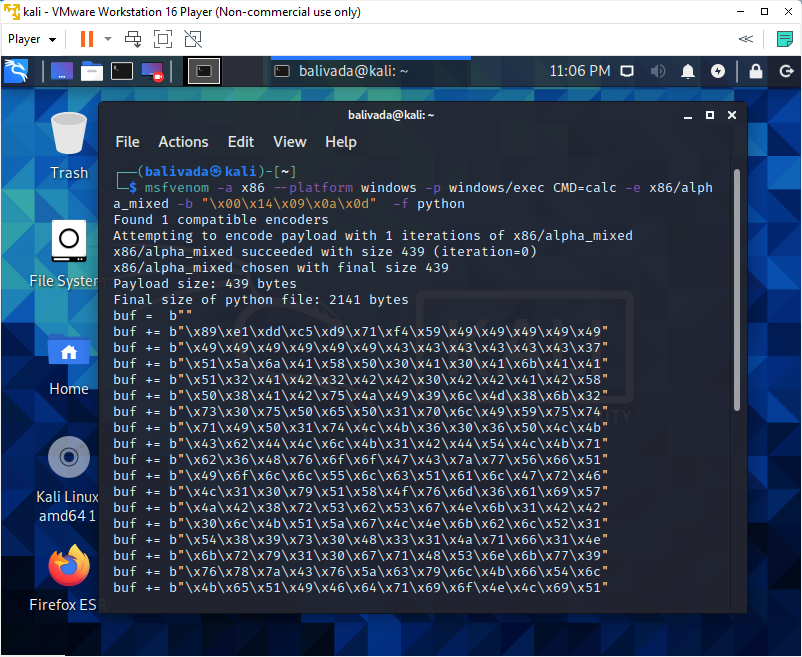
* **Try to crash the Frigate3\_Pro\_v36 and exploit it.**
* **Change the default trigger from cmd.exe to calc.exe (Use msfvenom in Kali linux).**

**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**

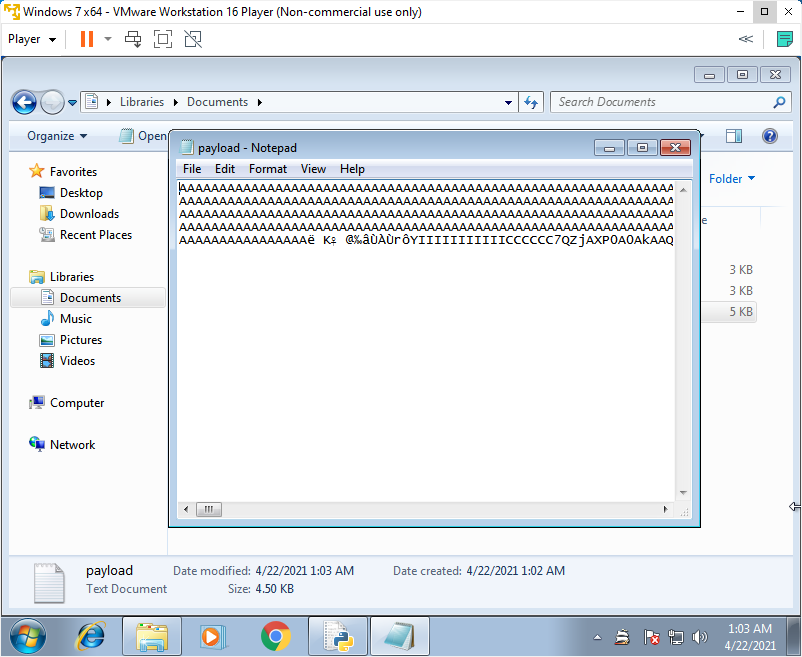
**Happy Learning !!!!!!**

****

**Replace this in the exploit2.py and generate the payload.**

**The payload generated is something like shown below in the screenshot.**

**Payload:**

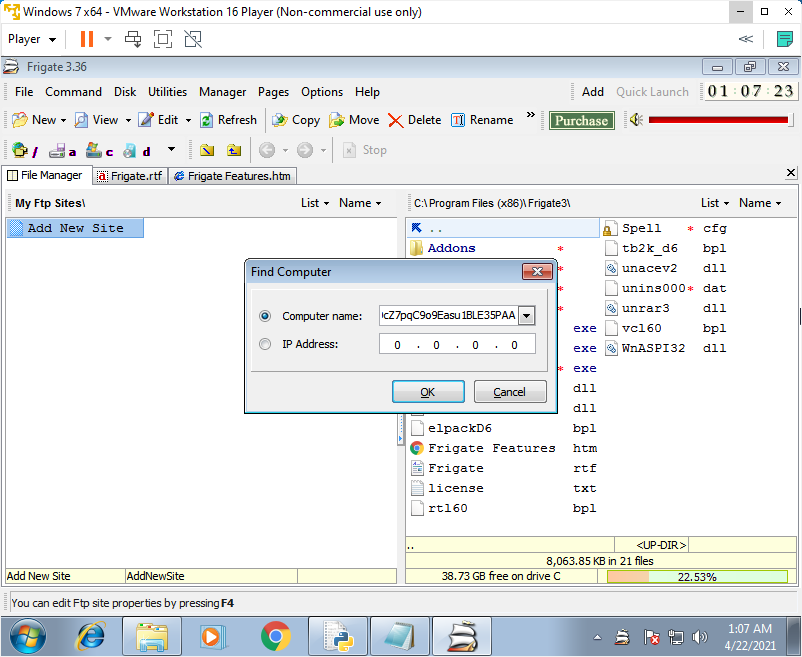
****

**The payload used here triggers calc.exe on crashing the application**

**The trigger is changed using msfvenom in kali linux.**

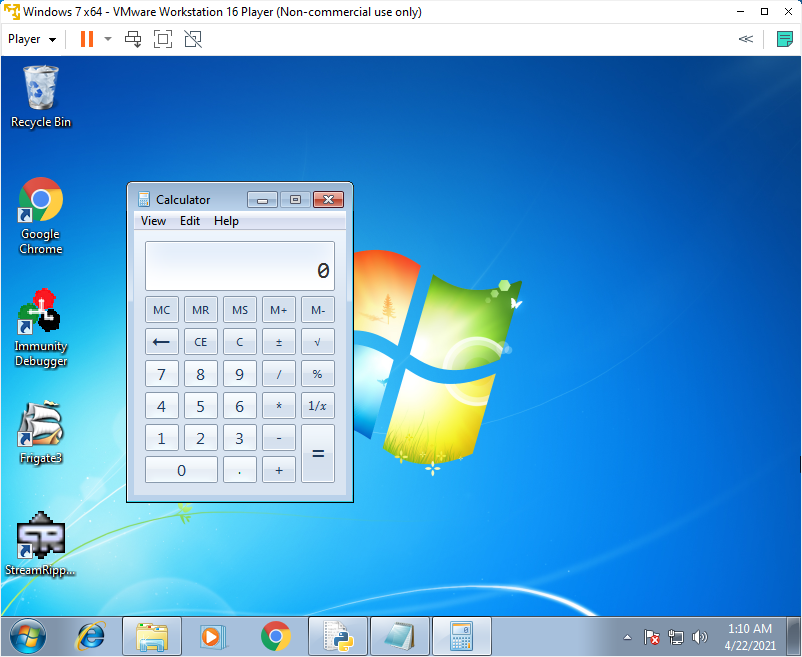
**The vulnerable field in Frigate is Find Computer in Disk menu.**

**Paste the generated payload in Computer name field**

****

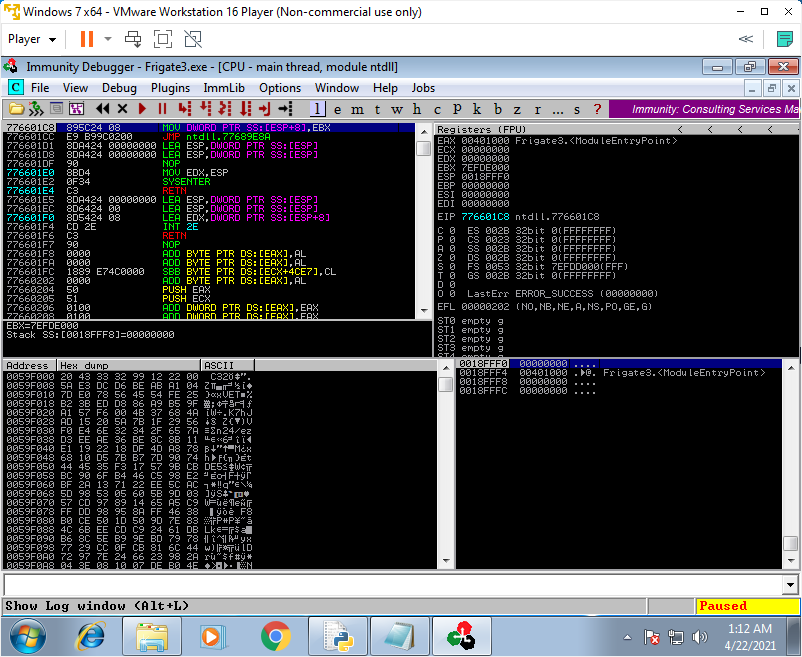
**On clicking okay buffer overflow occurs as the input writes over the adjacent memory locations causing it to crash**

**As we set the trigger as calc.exe, the calculator opens when after the application crashes.**

****

**Now let us check the EIP value using Immunity Debugger**

**Attach the Application in Immunity debugger**

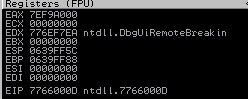
****

**EIP**

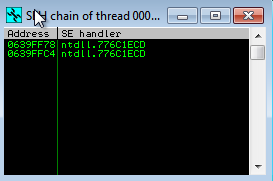
****

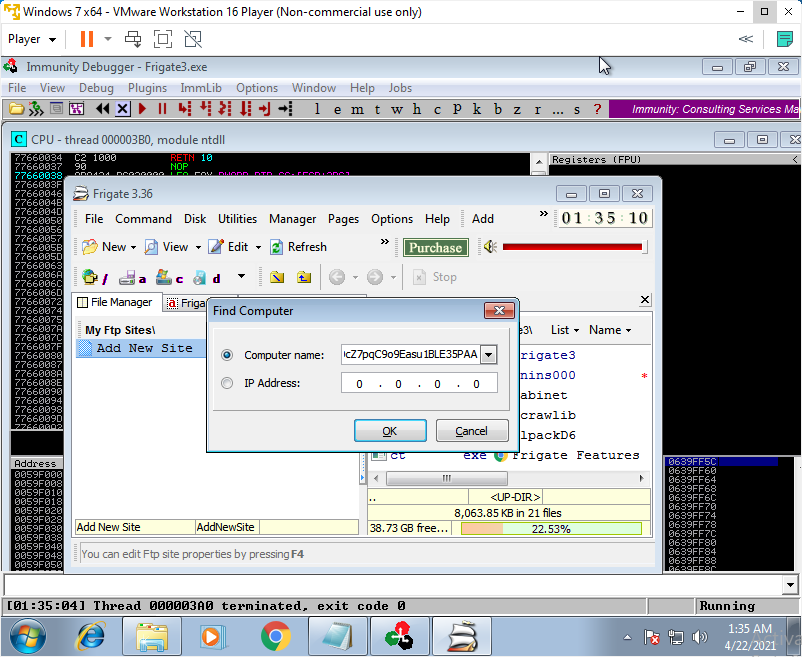
**Let us crash the application and see what happens to the registers**

**Initial before crash**

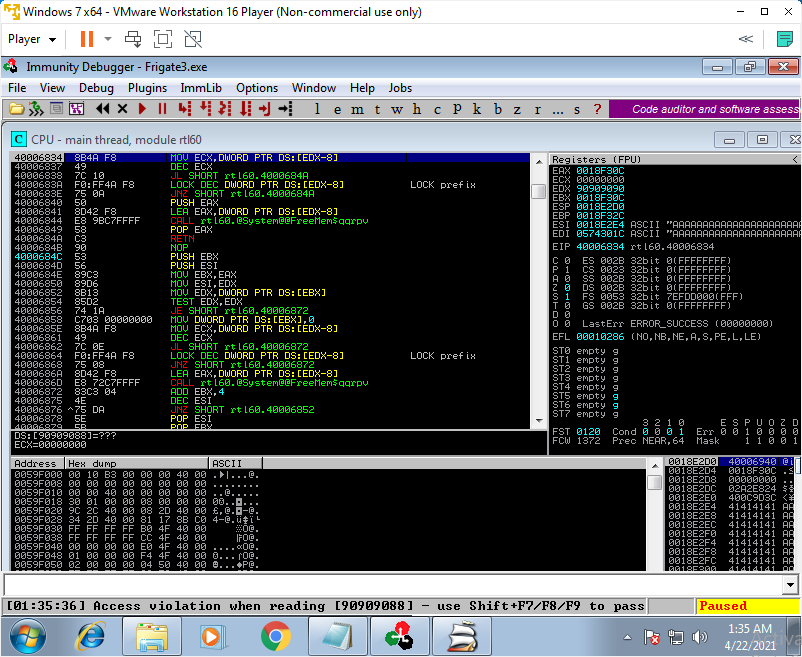
****

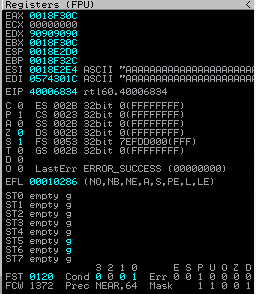
**SEH chain initially**

****

****

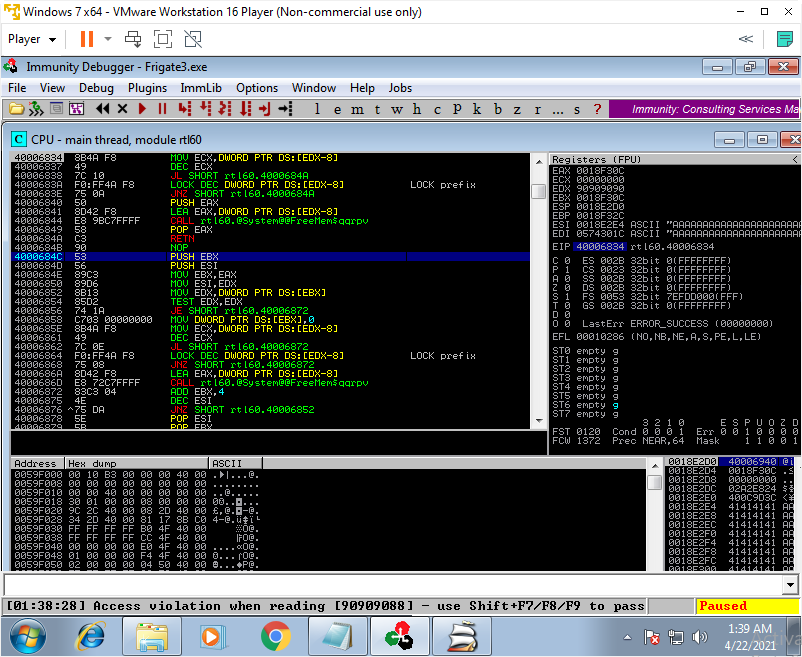
**After Crashing**

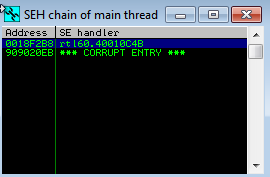
****

****

**Adjacent registers are over written.**

**Eip value is changed to 40006834**

****

****

**The dll rt160.40010c4B is loaded.**