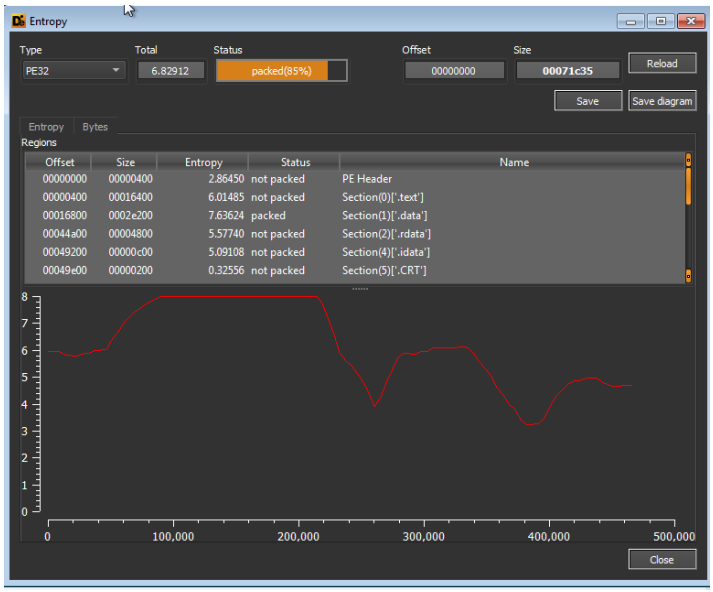
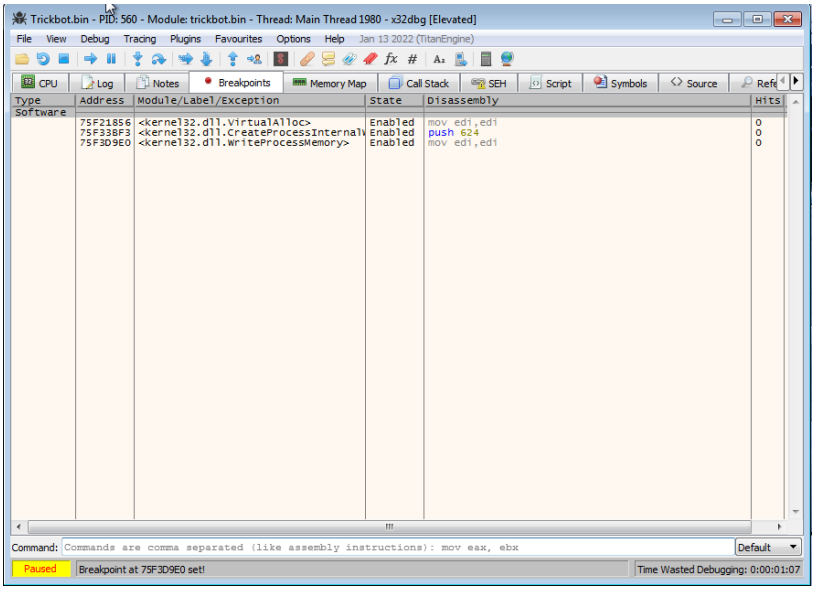
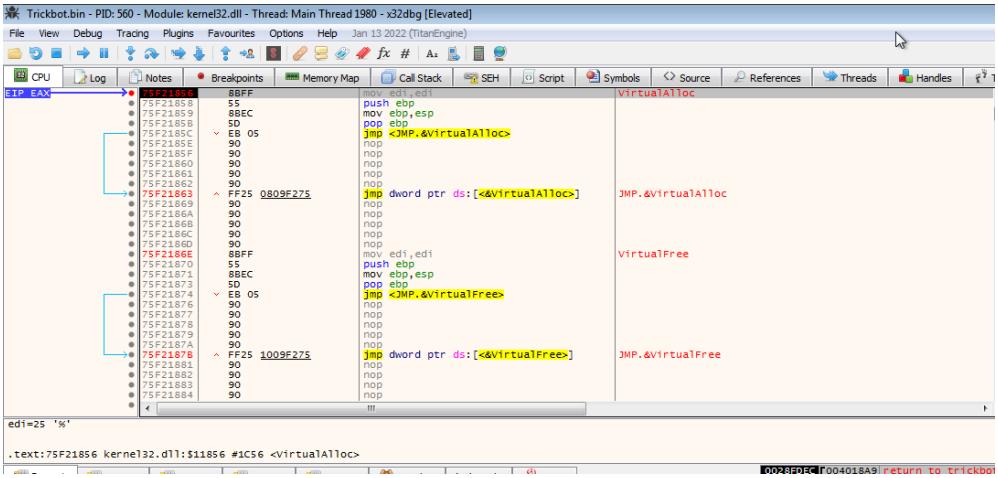
Initially load the sample in Detect it Easy in that we will open the entropy tab and here it is showing the status as packed.



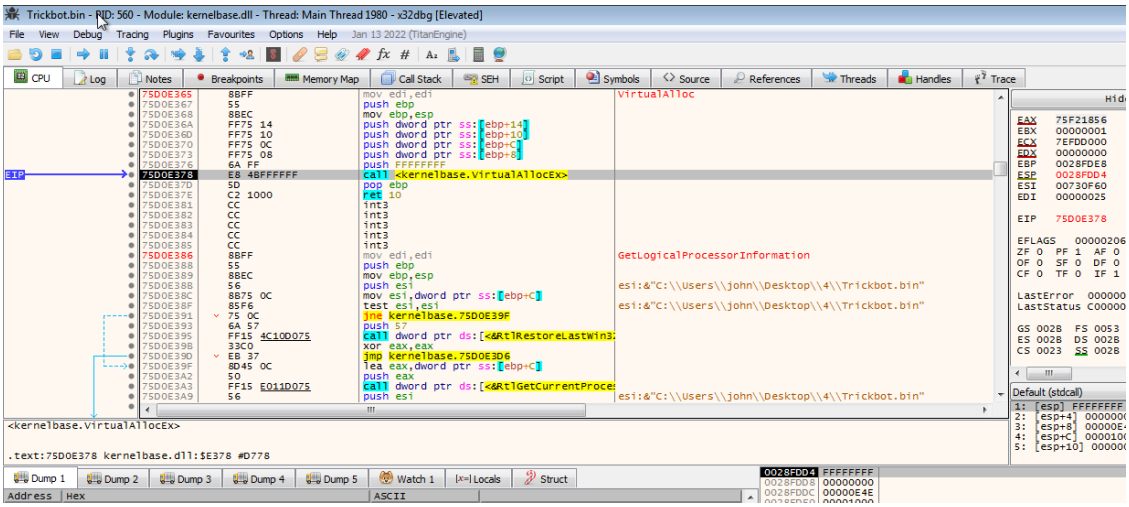
Now we will load the sample in x32dbg, and put breakpoint on VirtualAlloc, CreateProcessInternalW(when malware tries attach to newly created process or it can also create a new process by itself to run certain function), WriteProcessMemory(when malware tries to hijack an existing process by copying part of its code into that process memory in order to be stealthy).



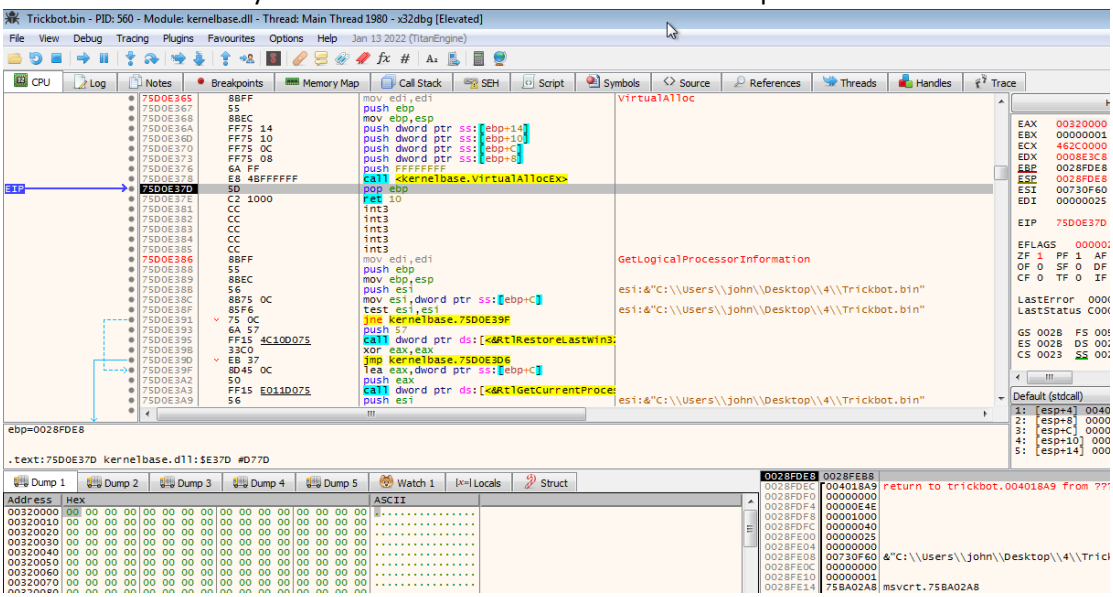
Press F9 and we will hit our first breakpoint aat VirtualAlloc, it will return the allocated region address into eax register.



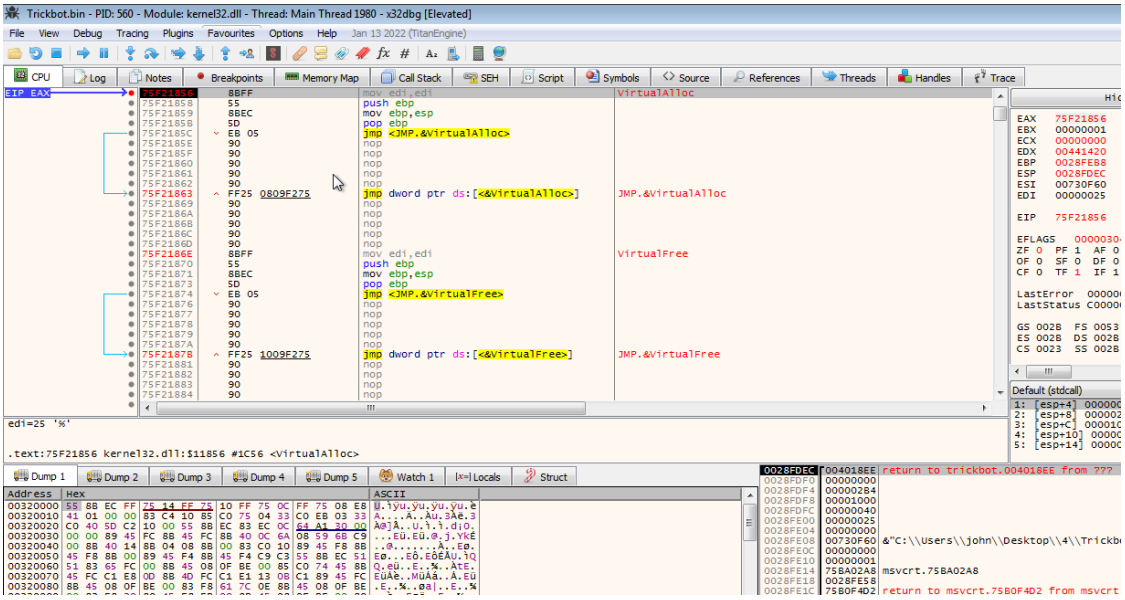
Here inside VirtualAlloc we will step over the kernelbase.VirtualAlloc and will note the address return in eax.



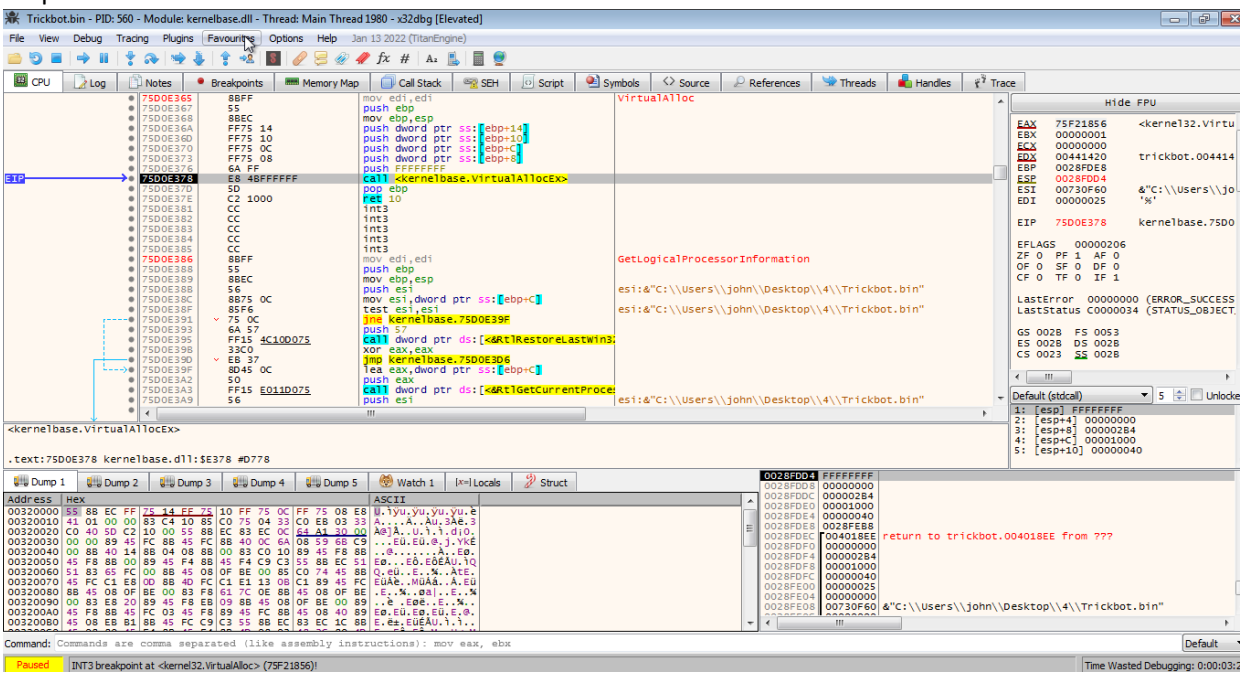
The address return by eax is 320000 follow this location in dump.



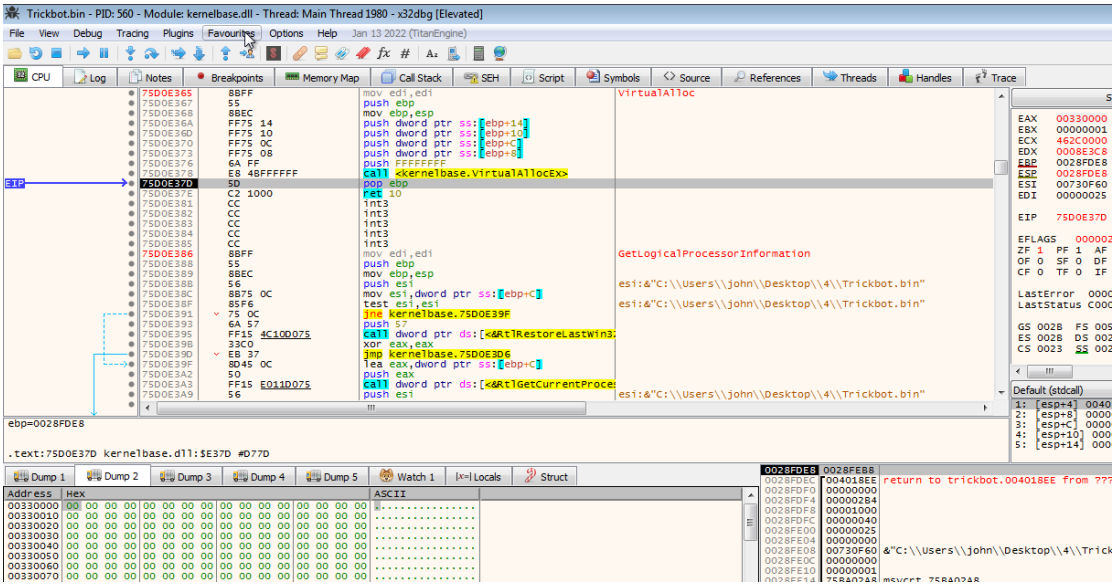
Now press f9 and here we can see the changes in dump also we hit the VirtualAlloc second time.



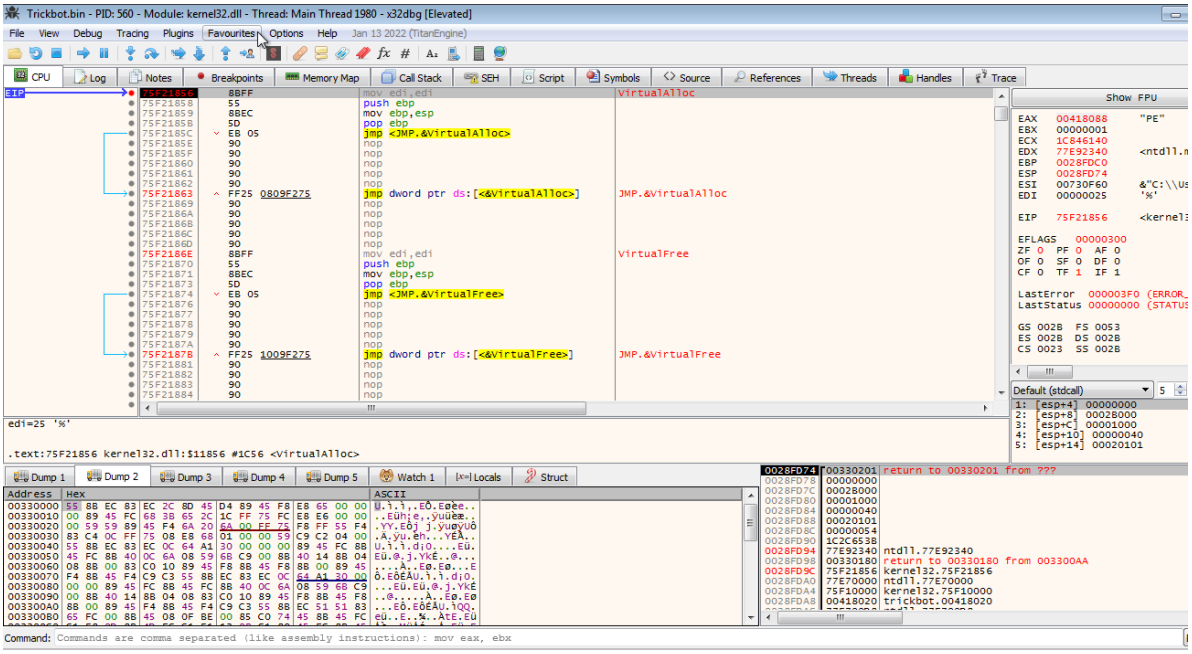
Step over it .



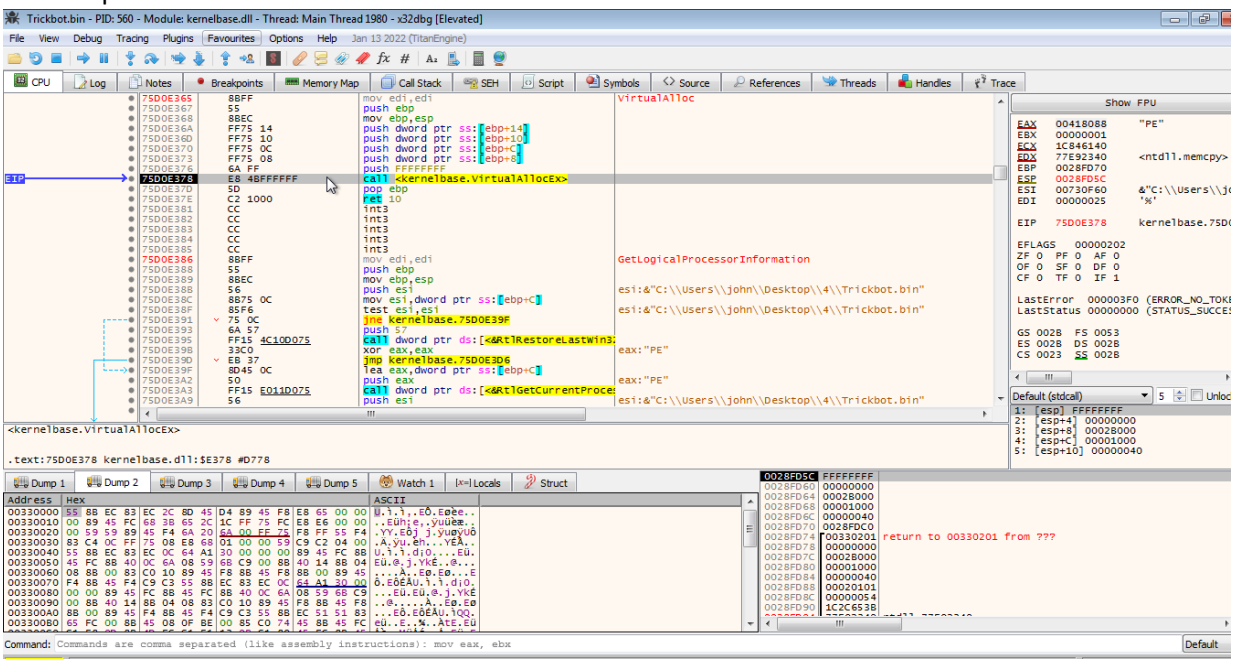
The address return by this VirtualAlloc is 330000 follow this region in Dump 2.



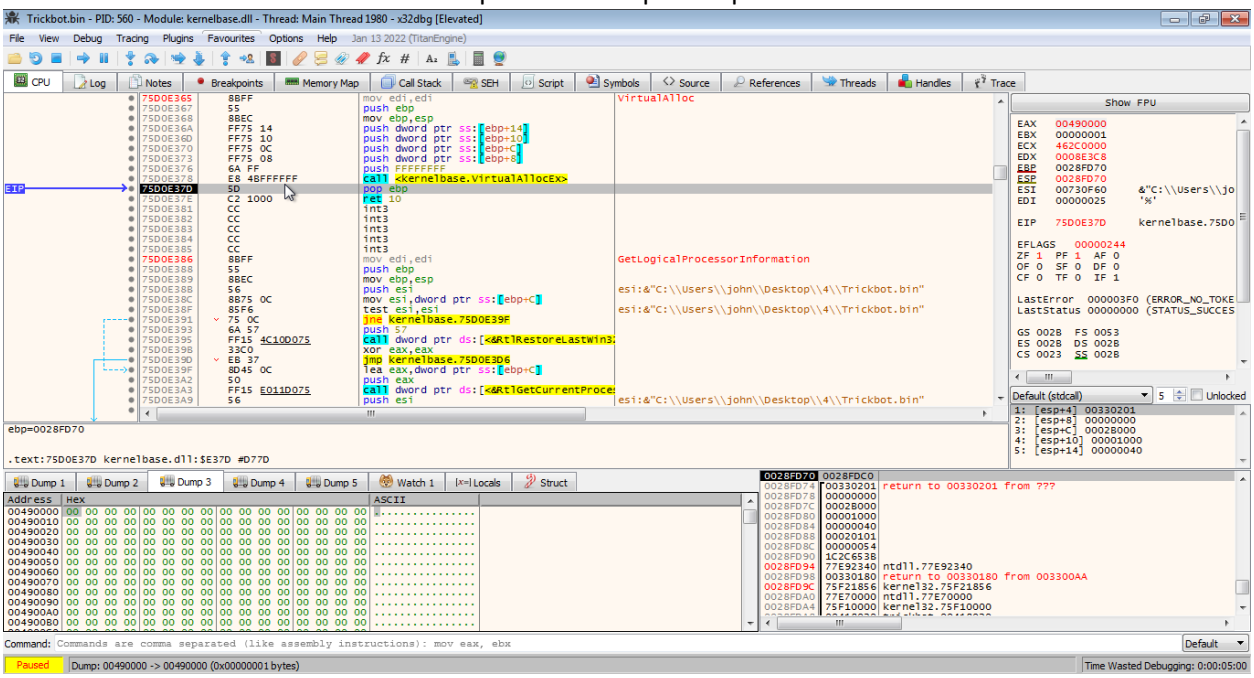
Now again presss f9 here we can see the changes in dump 2 unpacked shellcode and we will hit VirtualAlloc third time.



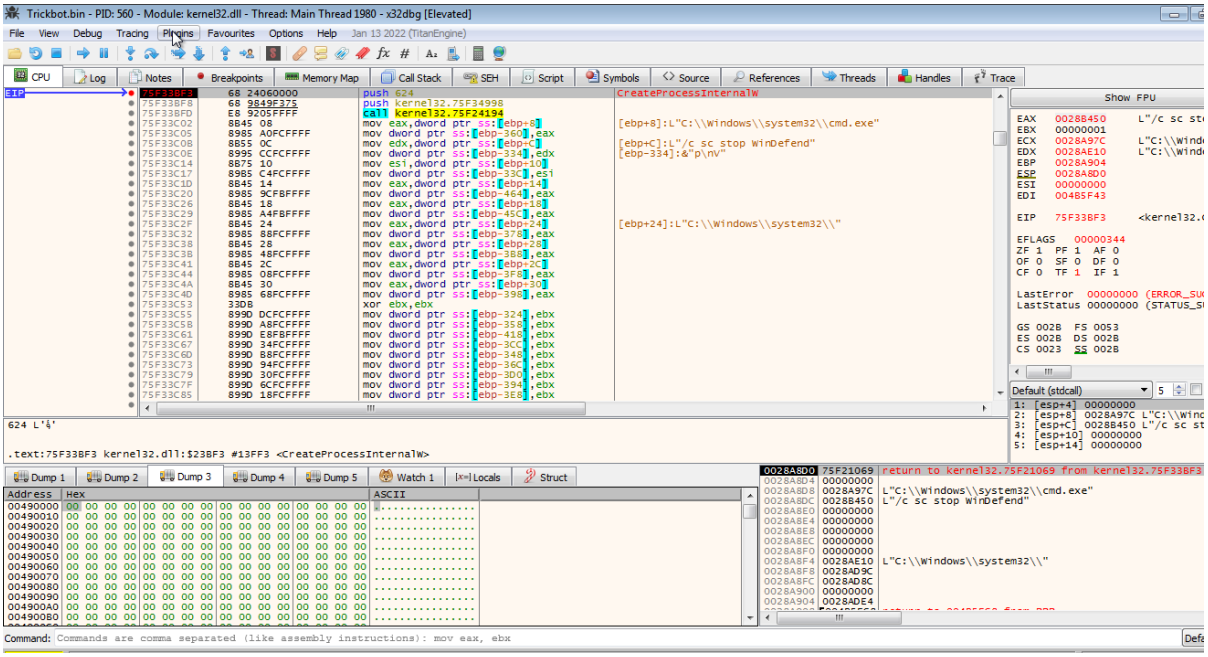
Now step over this.



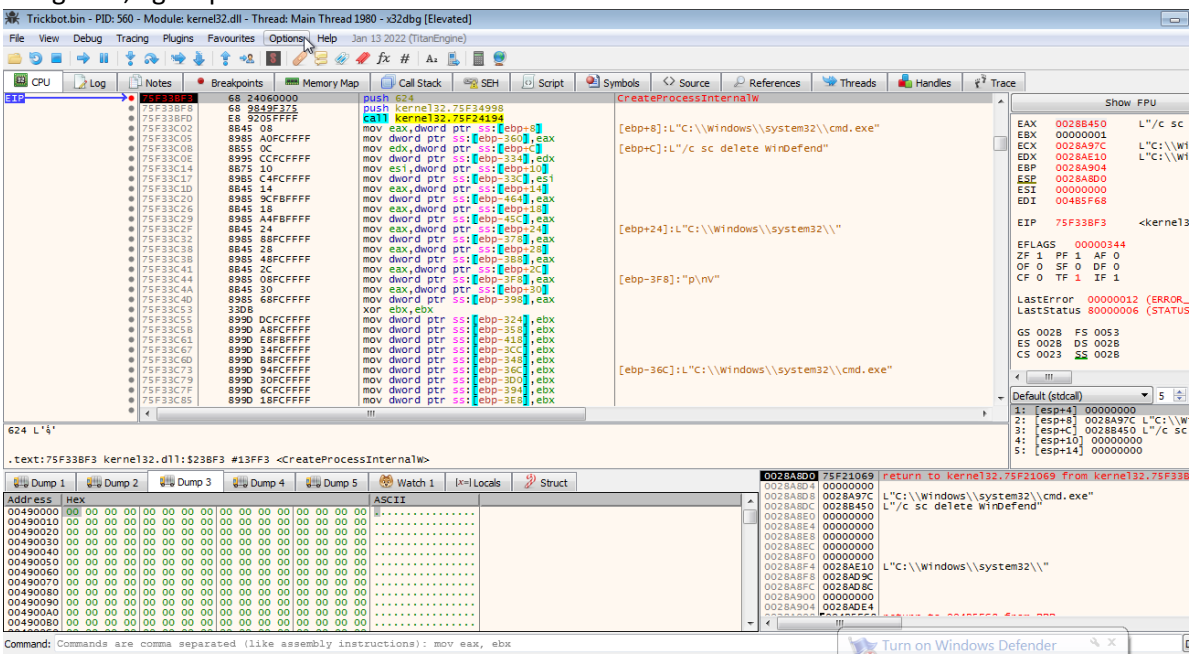
Note the address return in eax and dump that in Dump3 and press F9.



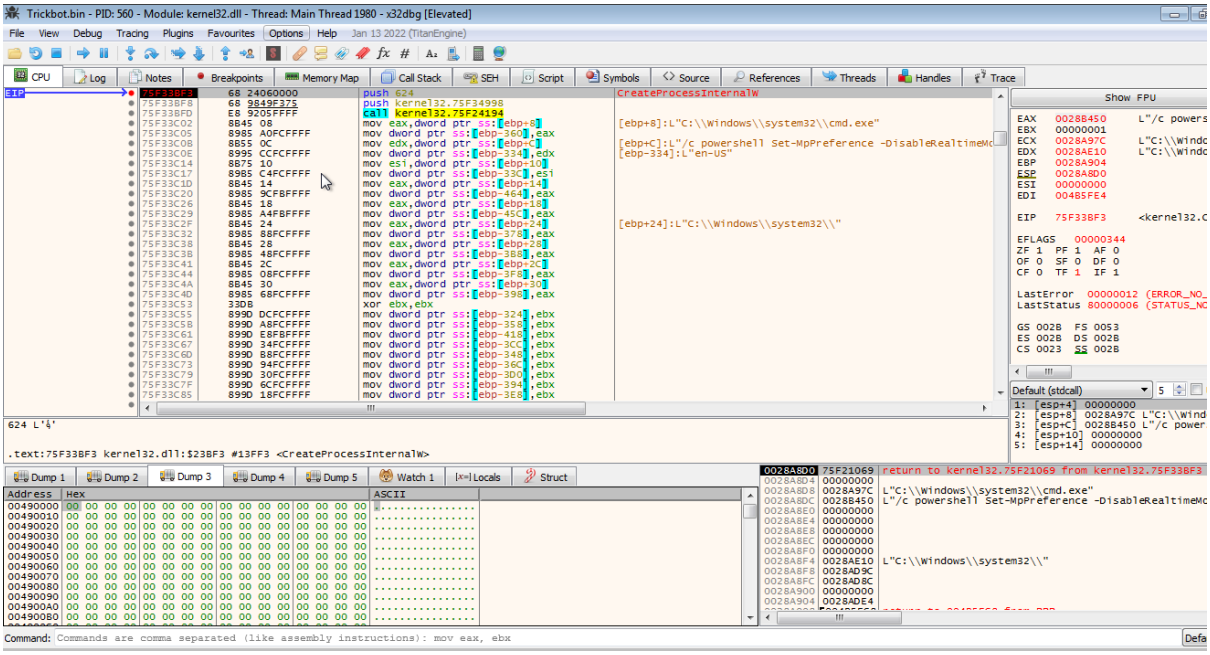
Now we hit another breakpoint CreateProcessInternalW it is using this API to run cmd command in order to stop Window Defender, press f9 to run.



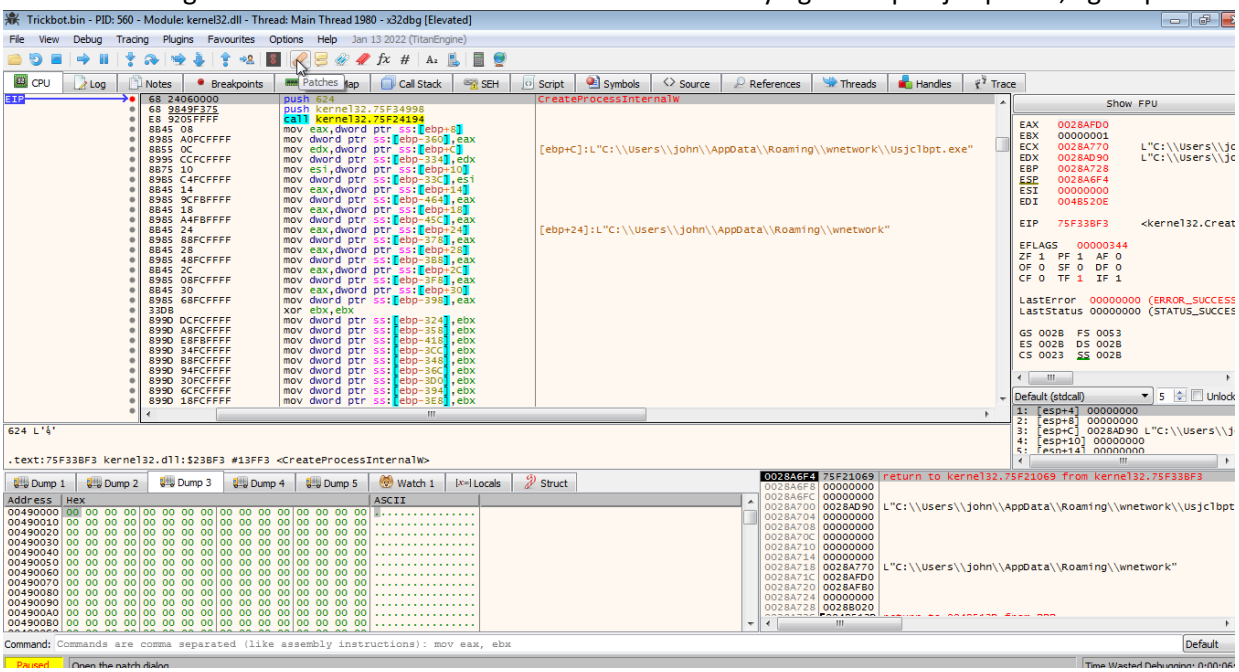
Now here we again hit CreateProcessInternalW now here it is trying to delete window defender service usnig cmd, again press F9.



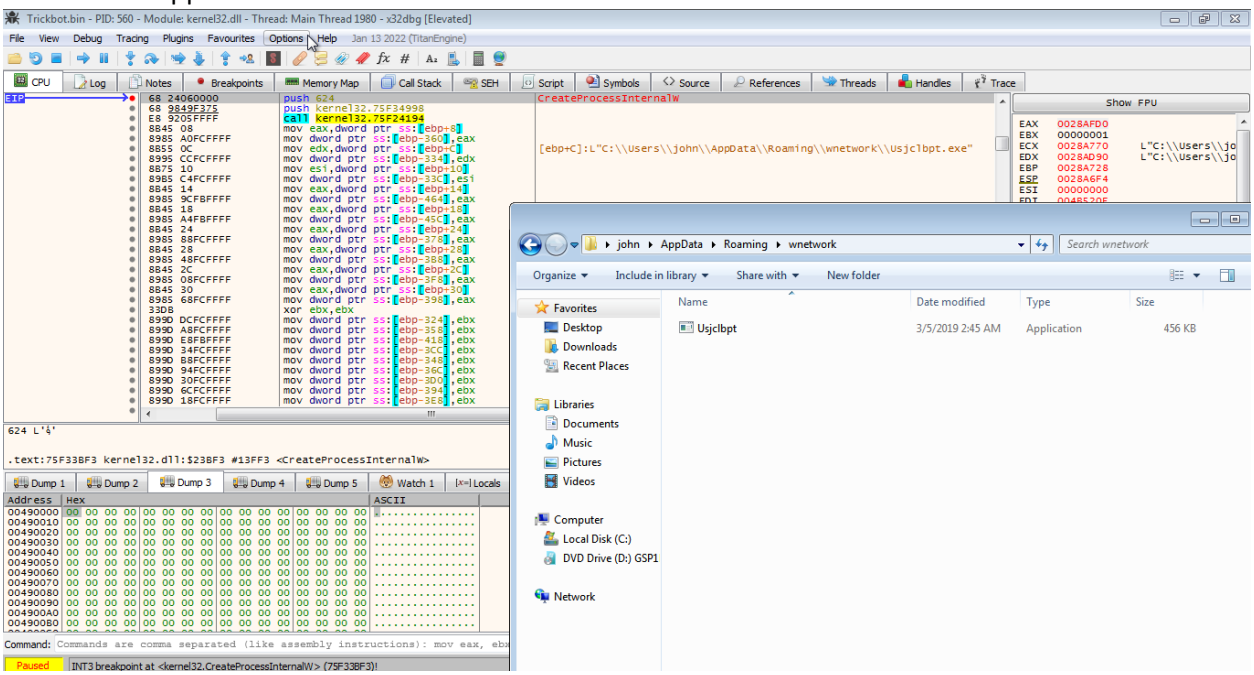
Now here we again hit CreateProcessInternalW now here it is trying to disable real time monitoring using powershell command, again press F9



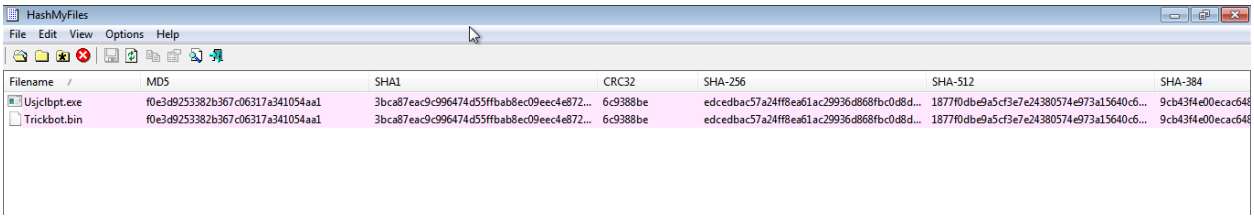
Now here we again hit CreateProcessInternalW now here it is trying to drop Usjclbpt.exe, again press F9



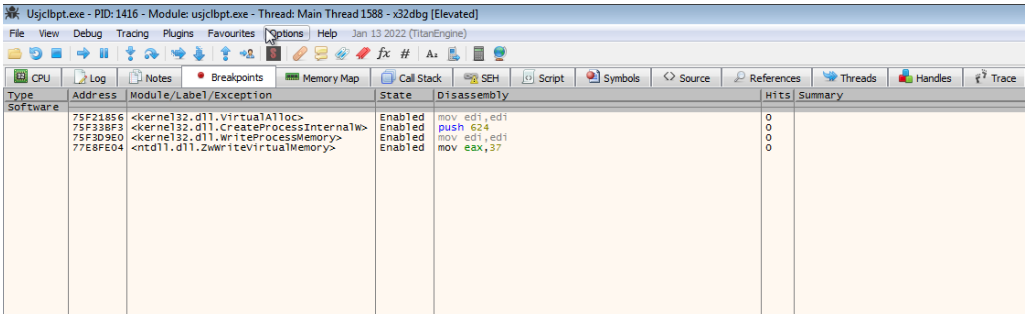
This is the dropped file.



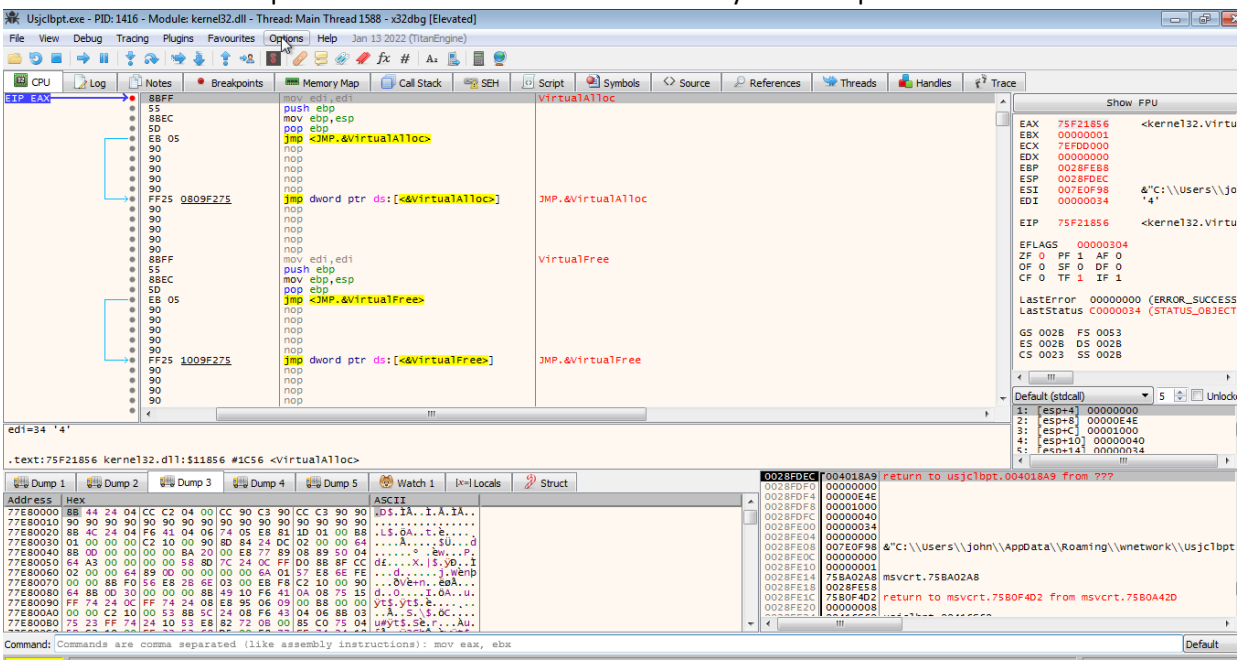
Now compare the hash of this file with the trickbot hash using HashMyFiles.Both file has same hash, trying to become more stealthy.



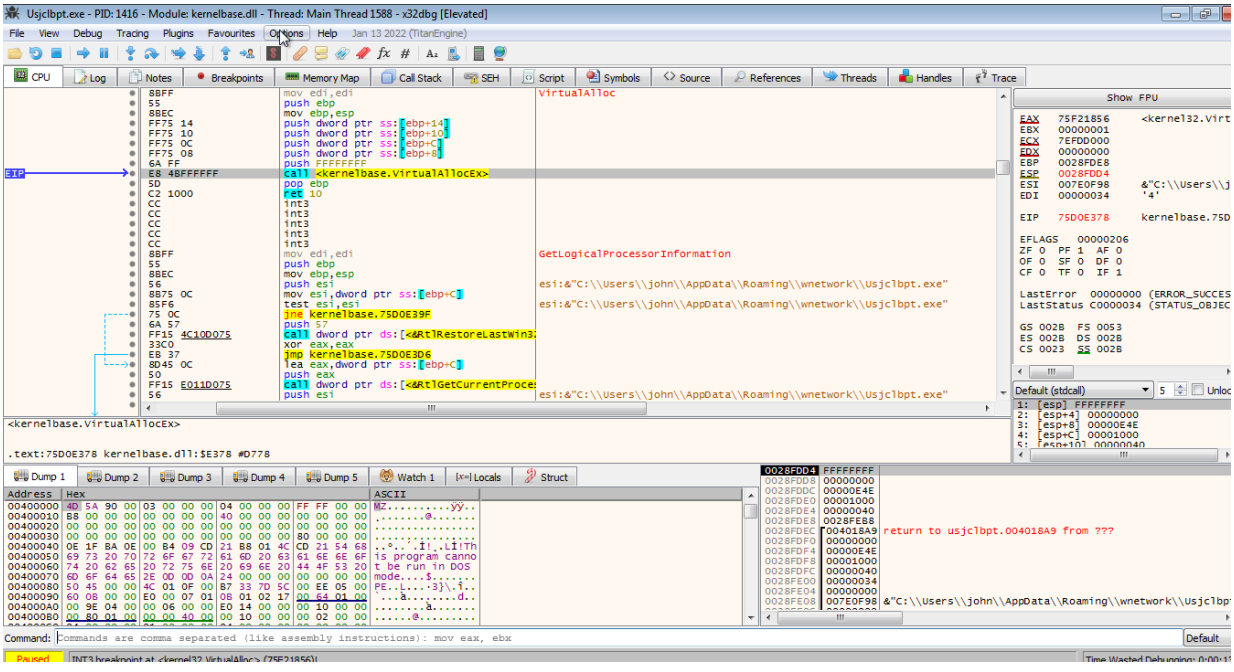
Now stop the ongoing x32dbg and load the newly dropped trickbot in x32dbg and apply the breakpoints VirtualAlloc, CreateProcessInternalW, WriteProcessMemory, ZwWriteVirtualMemory run it.



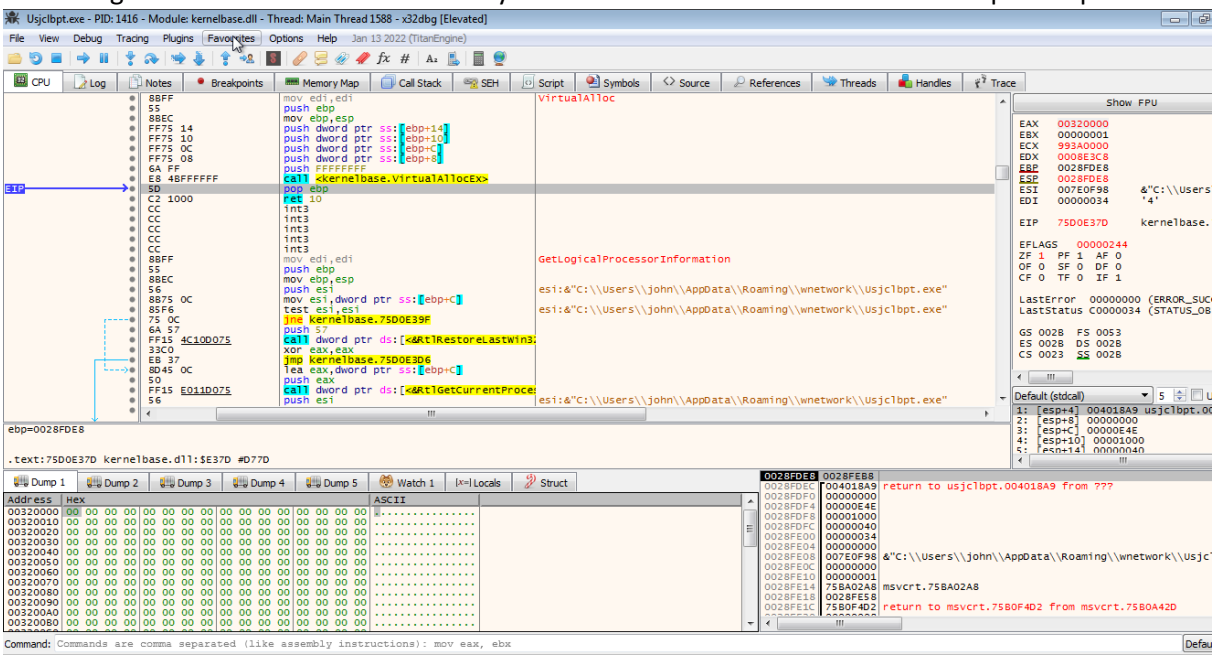
We hit VirtualAlloc step over it and load the address return by it in dump1.



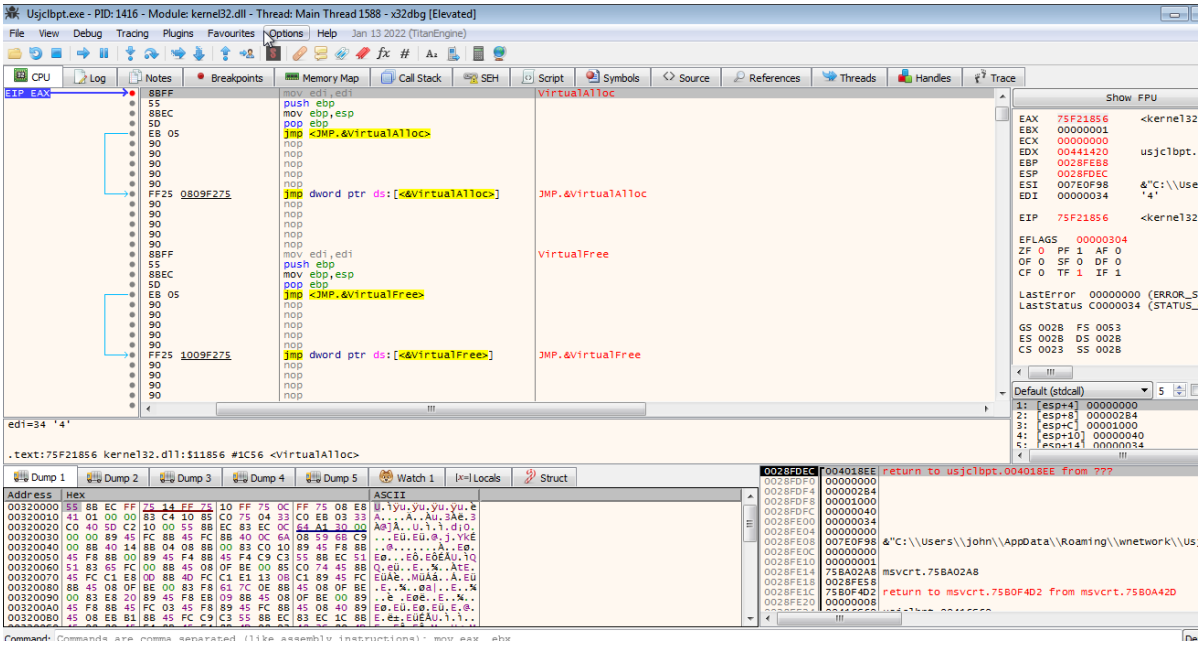
Here we step over it.



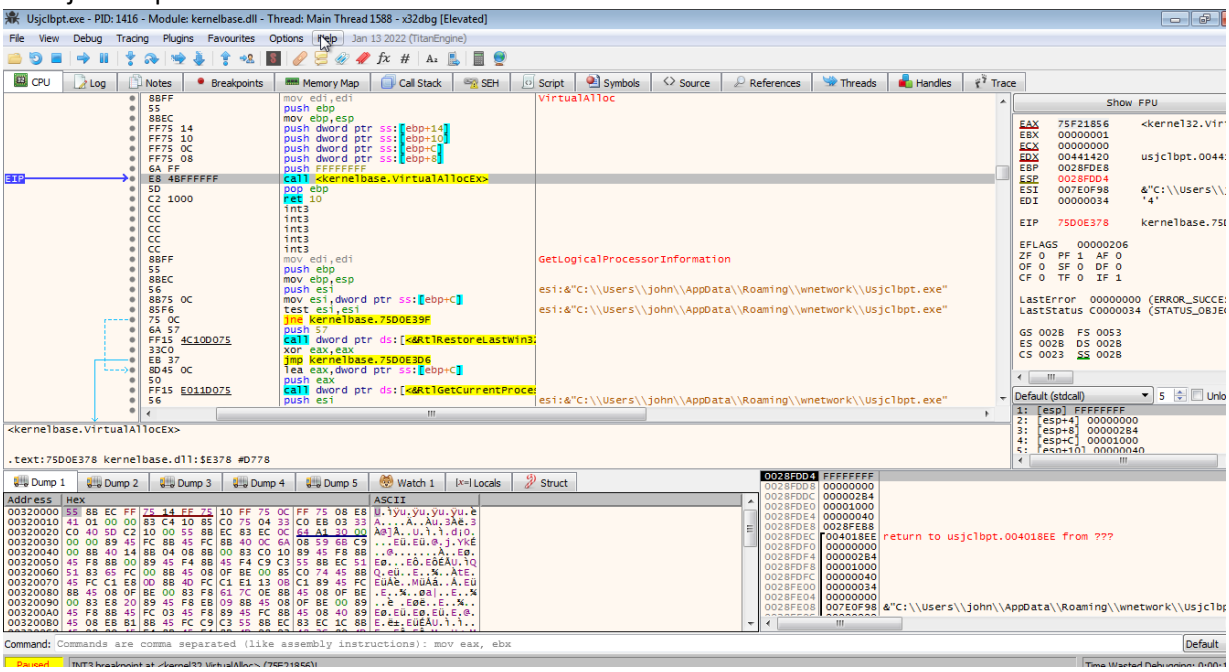
Here we get the return address 320000 by VirtualAlloc i.e in eax and load it in dump1 and press f9.



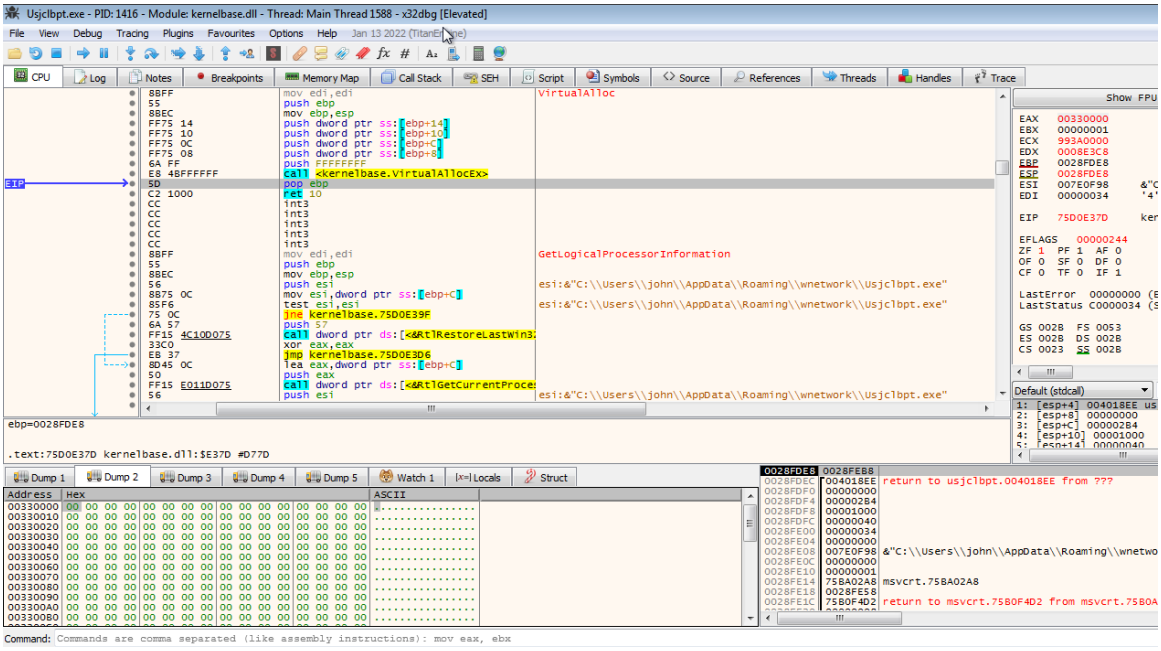
We hit VirtualAlloc step over it and also the dump1 has been filled by shellcode and load the address return by it in dump2.



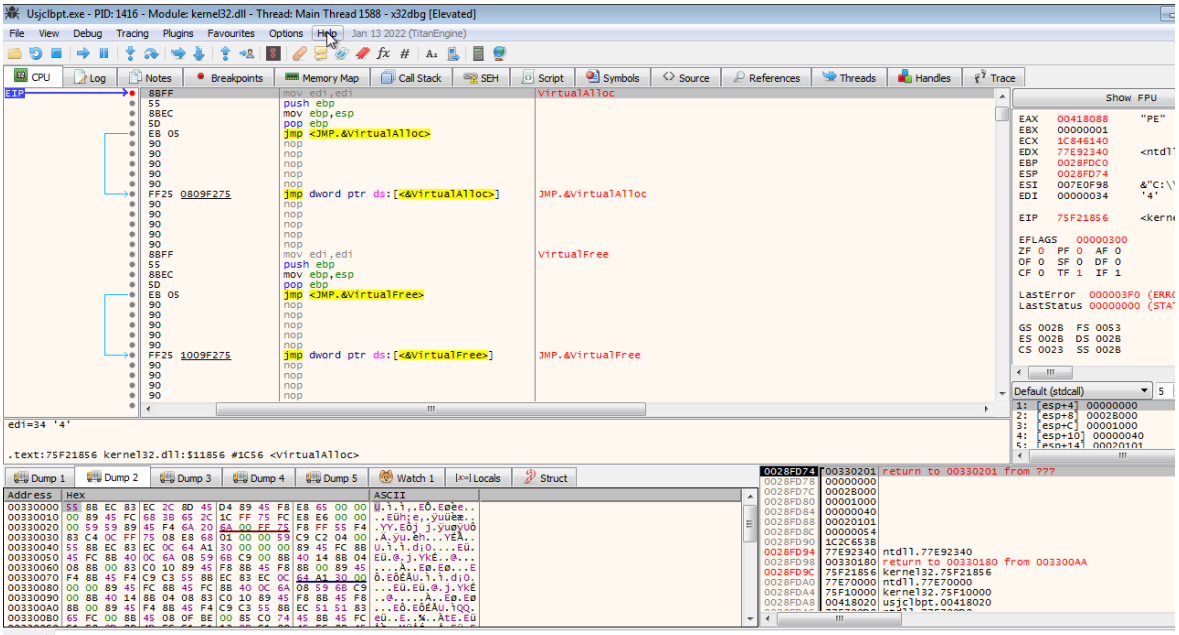
Here just step over it.



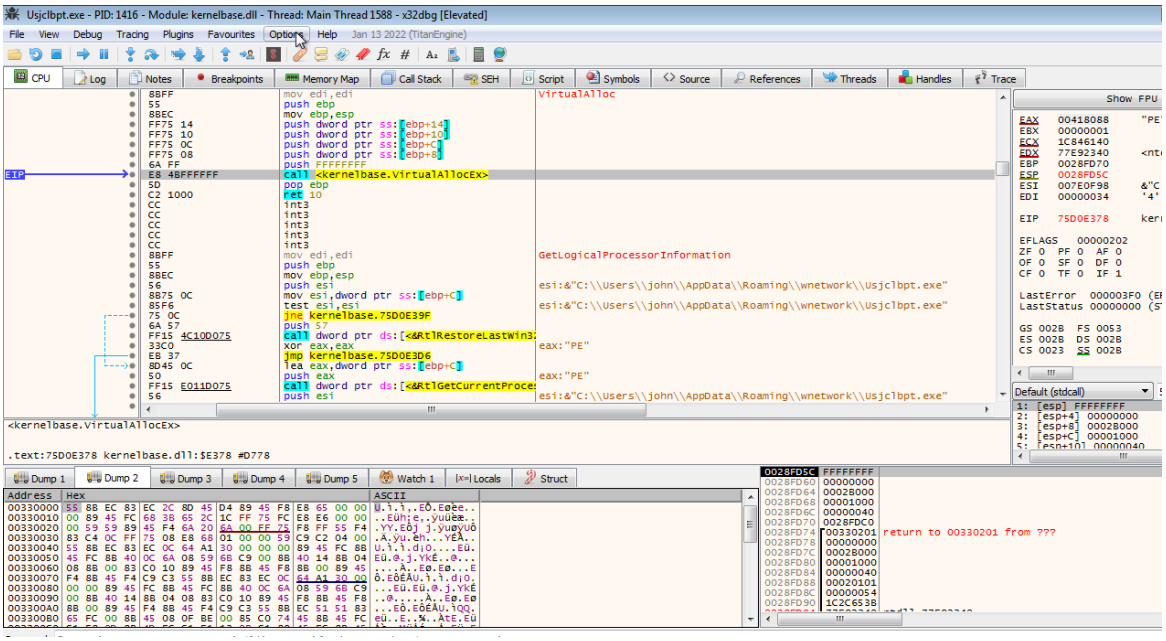
Load the address return by it in dump2 and press f9.



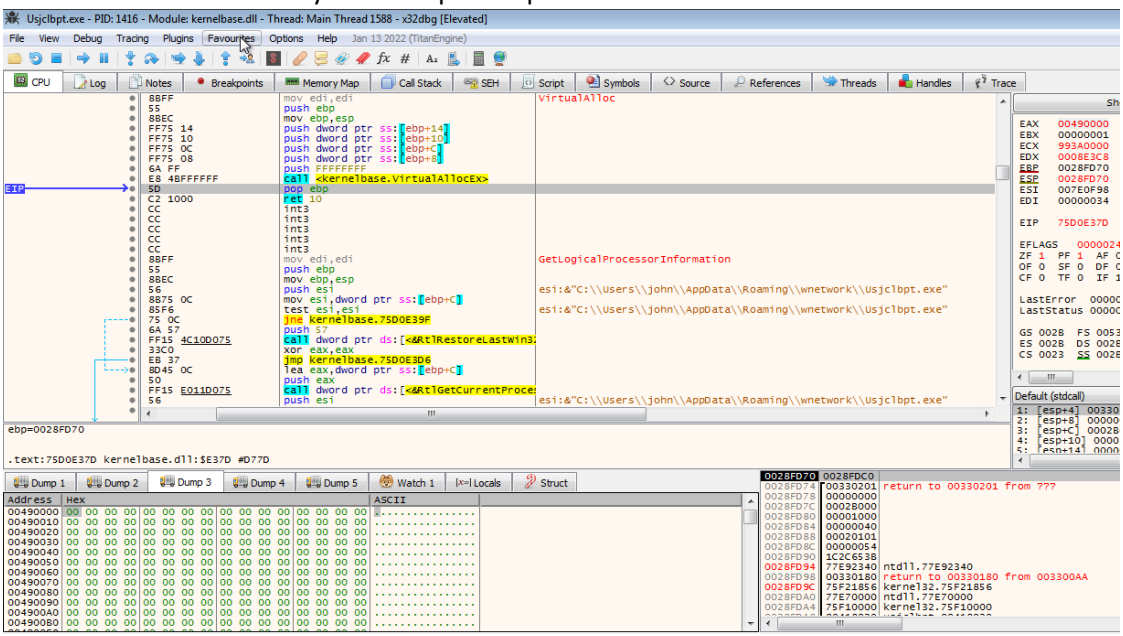
We hit VirtualAlloc step over it and also the dump2 has been filled by shellcode and load the address return by it in dump3.



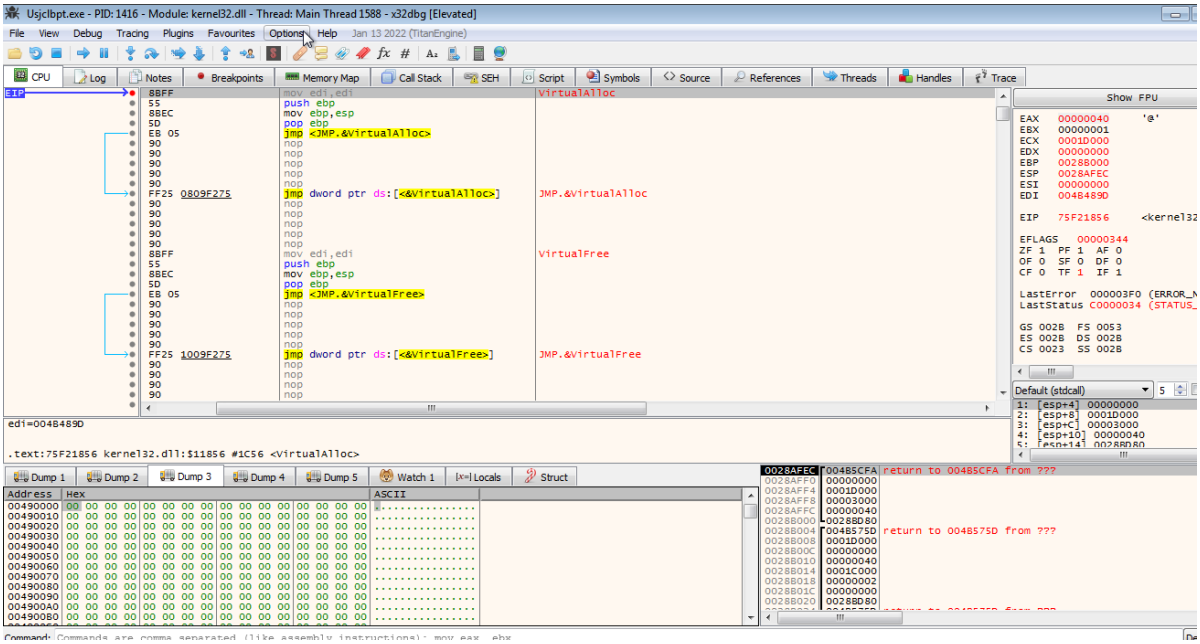
Here just step over it.



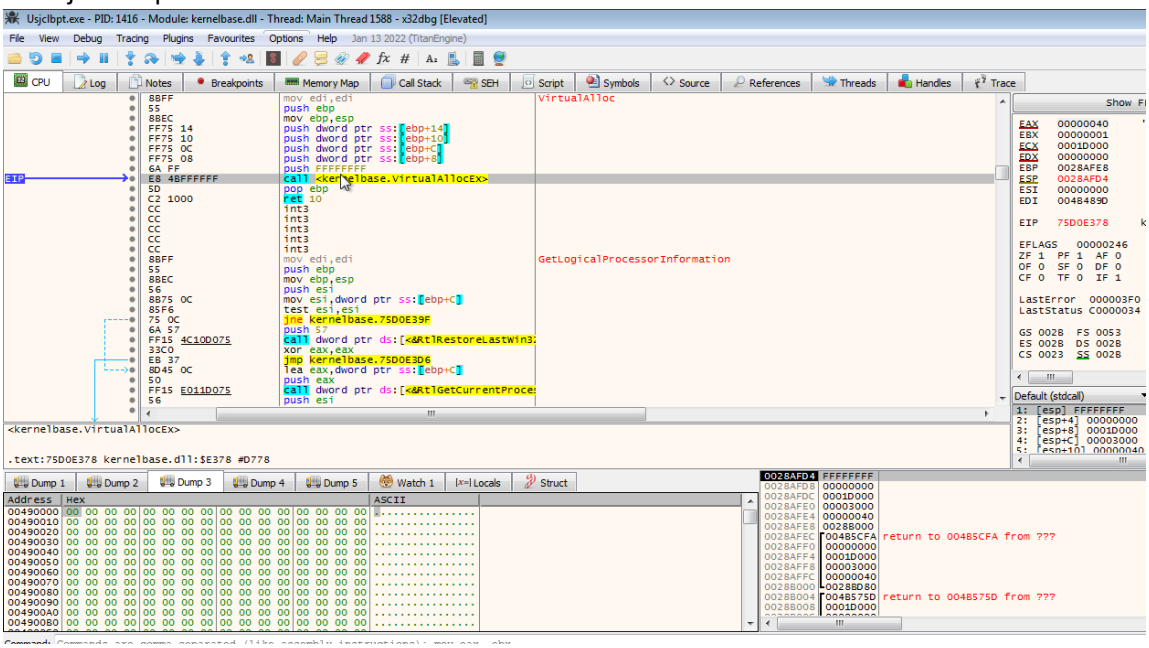
Load the address return by it in dump3 and press f9.



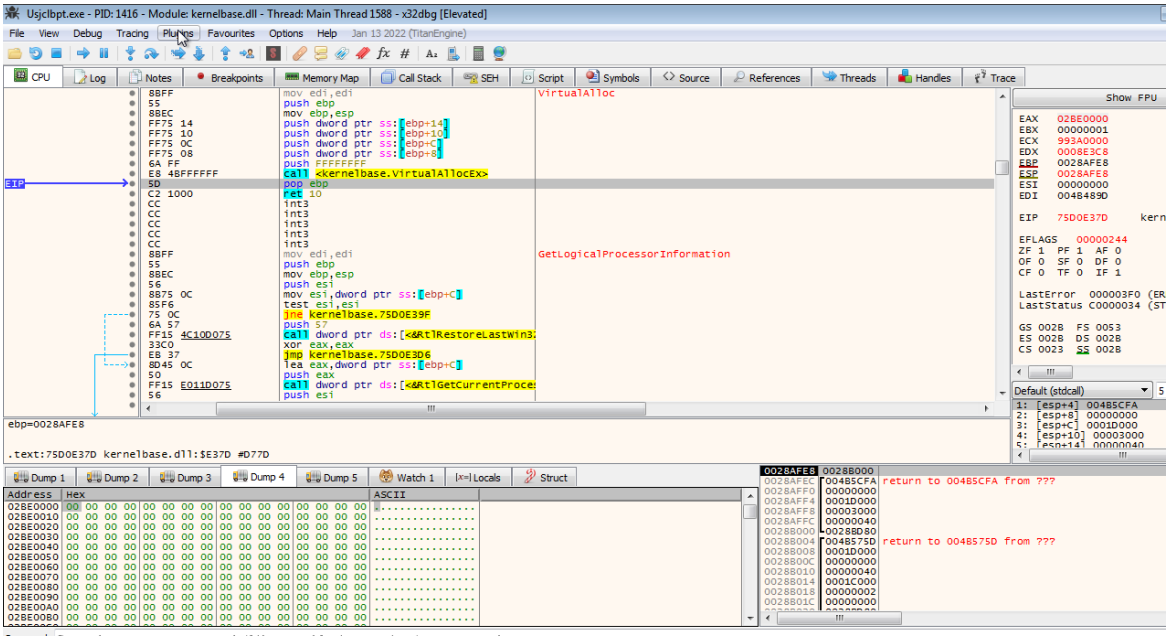
We hit VirtualAlloc step over it and also the dump3 has been filled by shellcode and load the address return by it in dump4.



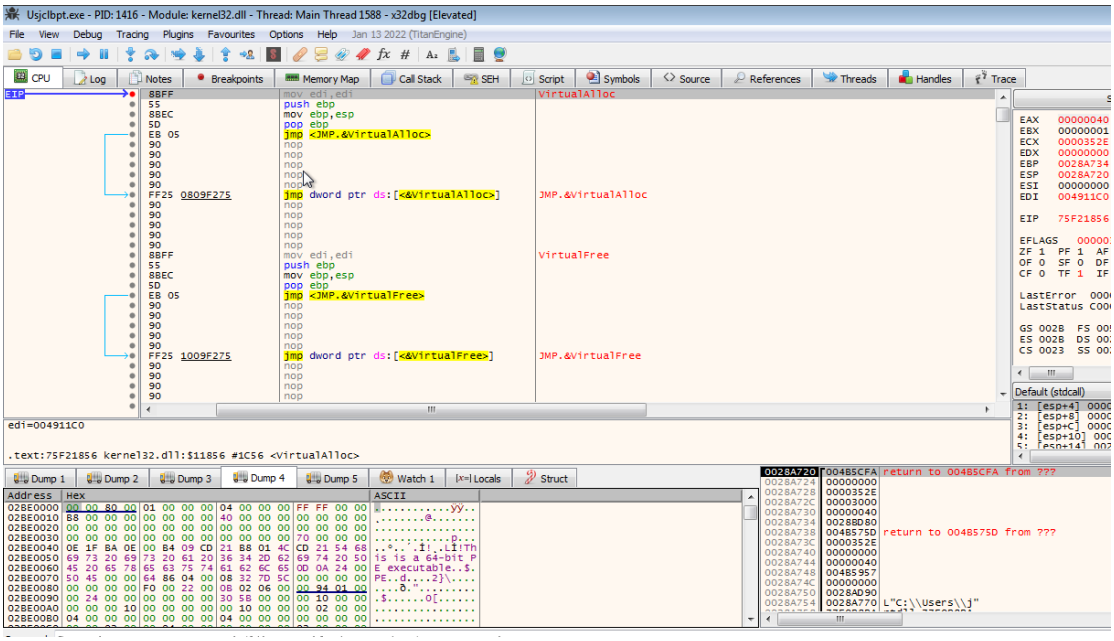
Here just step over it .



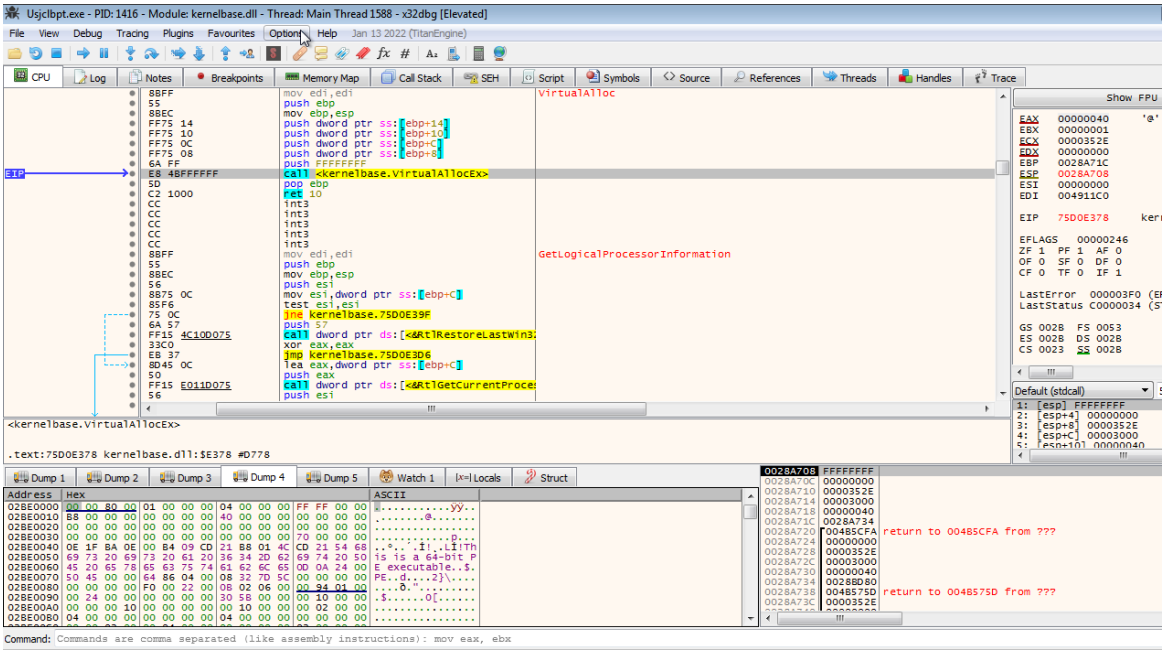
Load the address return by it in dump3 and press f9.



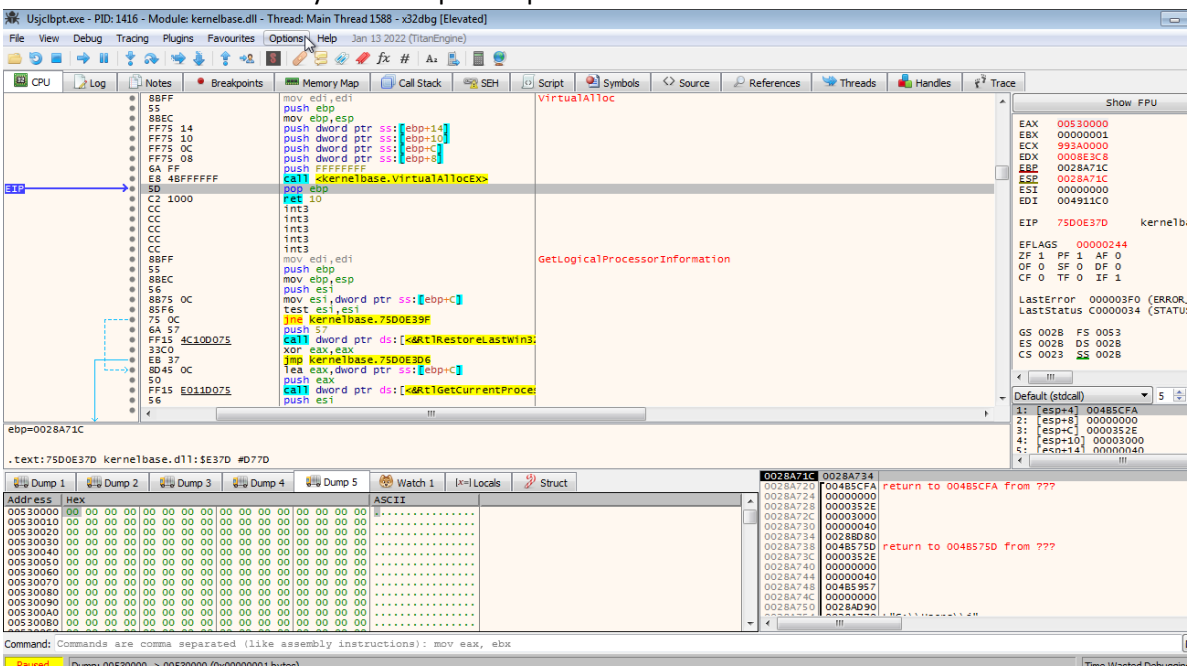
We hit VirtualAlloc step over it and also the dump4 has been filled by shellcode and load the address return by it in dump5.



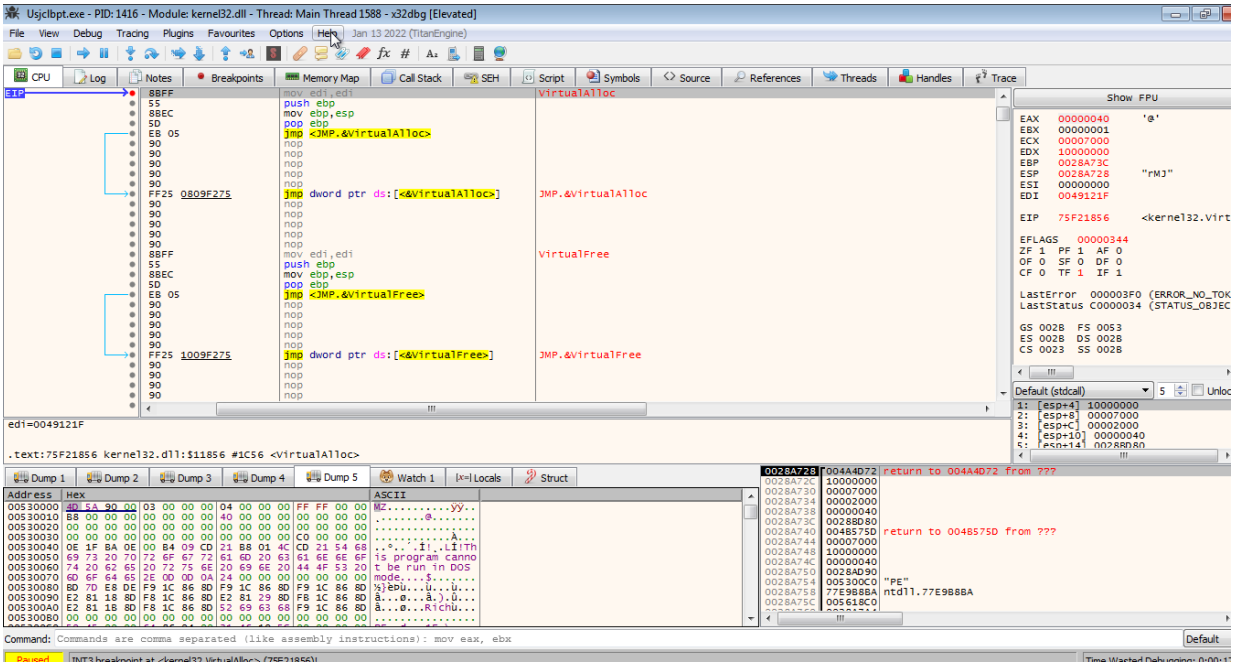
Here just step over it.



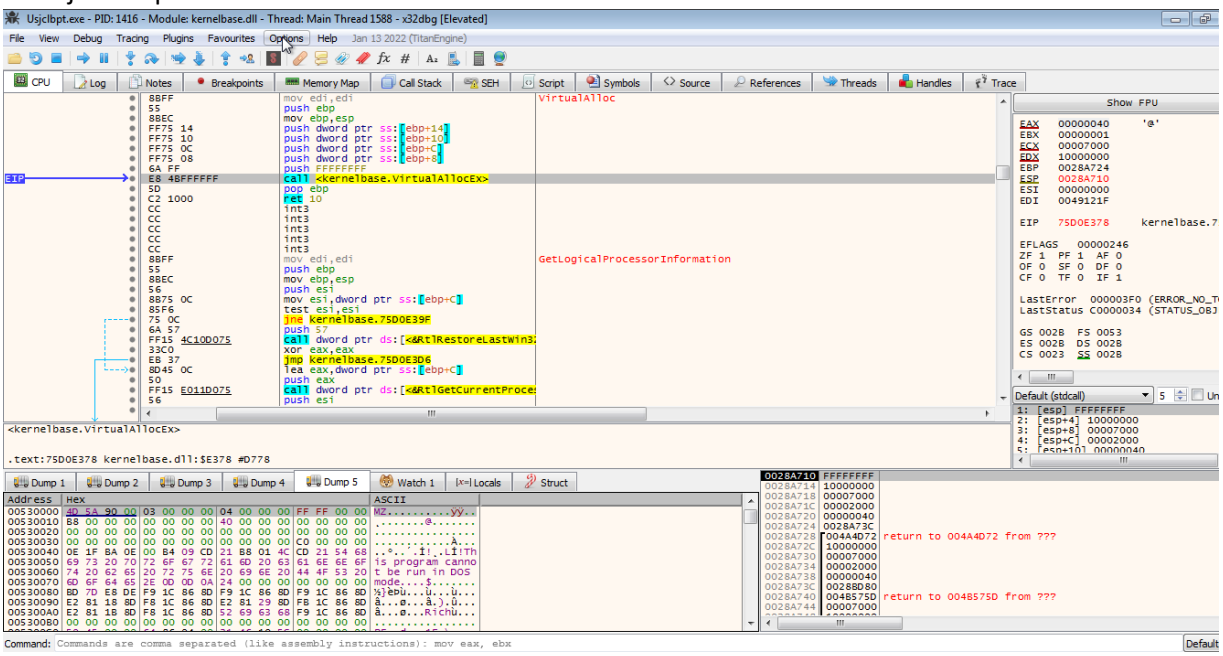
Load the address return by it in dump5 and press f9



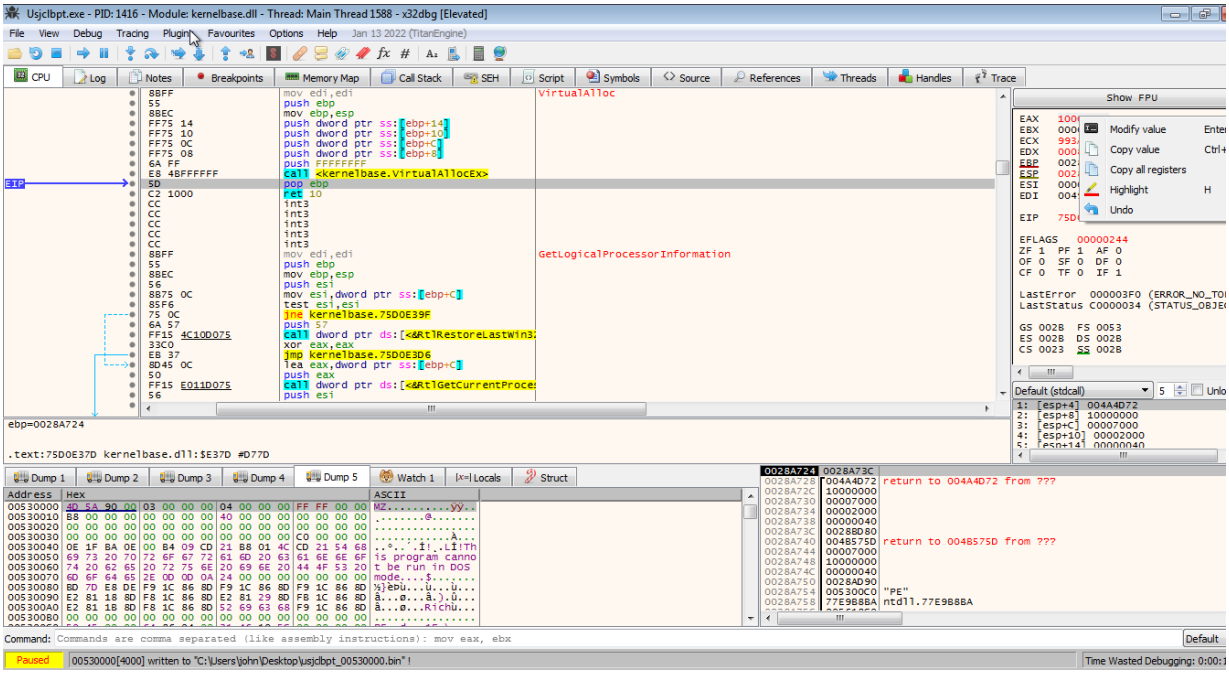
We hit VirtualAlloc step over it and also the dump5 has been filled by shellcode and load the address return by it in dump6



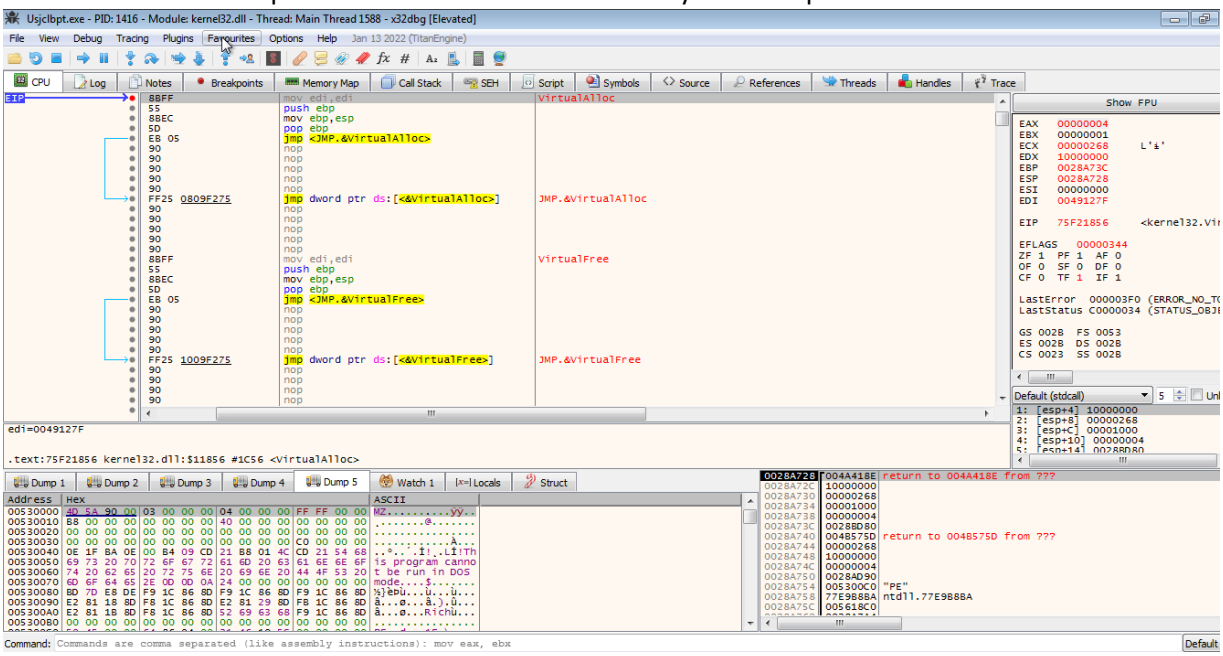
Here just step over it



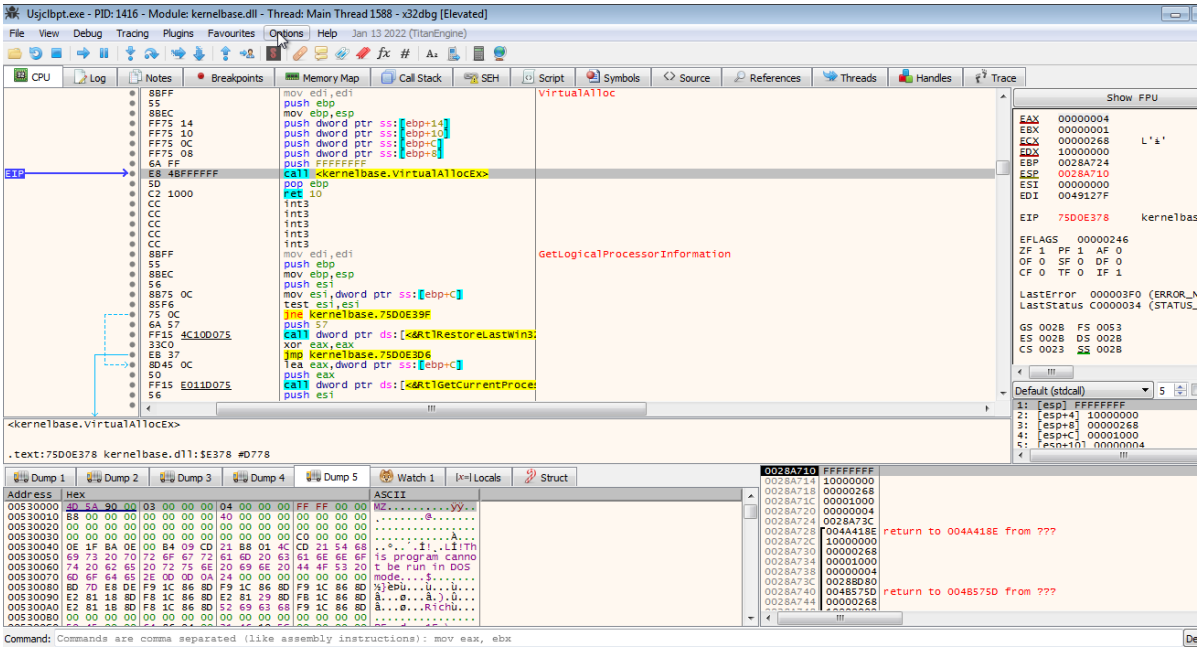
Here we can't load the return address so just press f9.



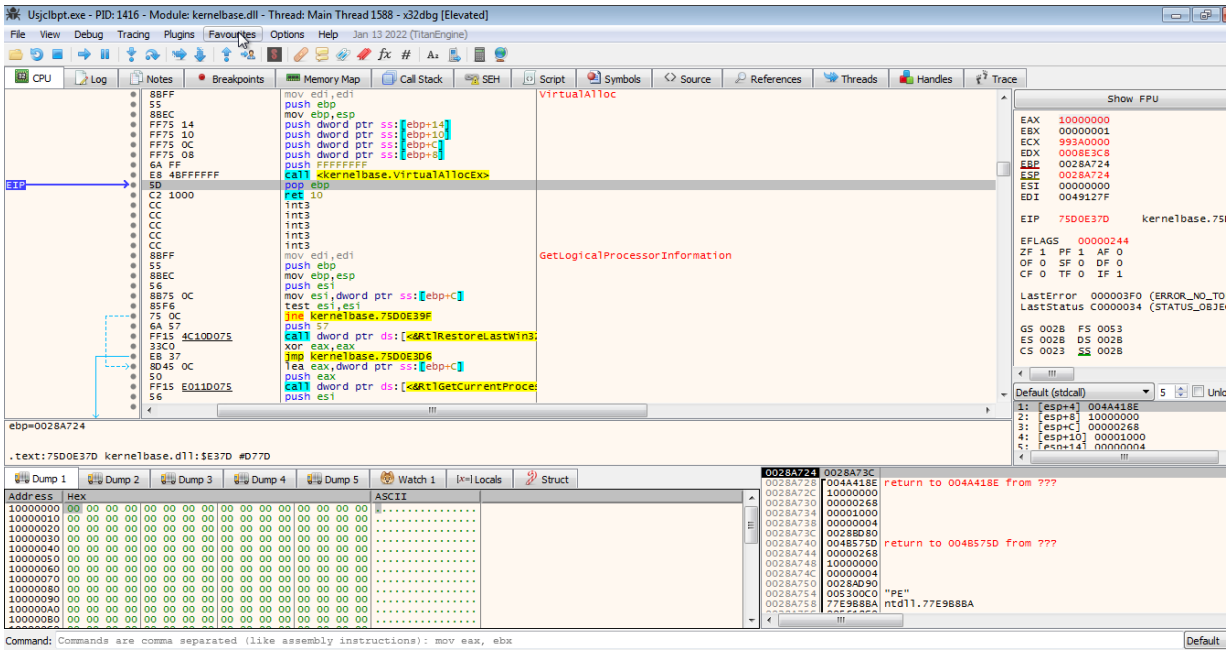
We hit VirtualAlloc step over it load the address return by it in dump.



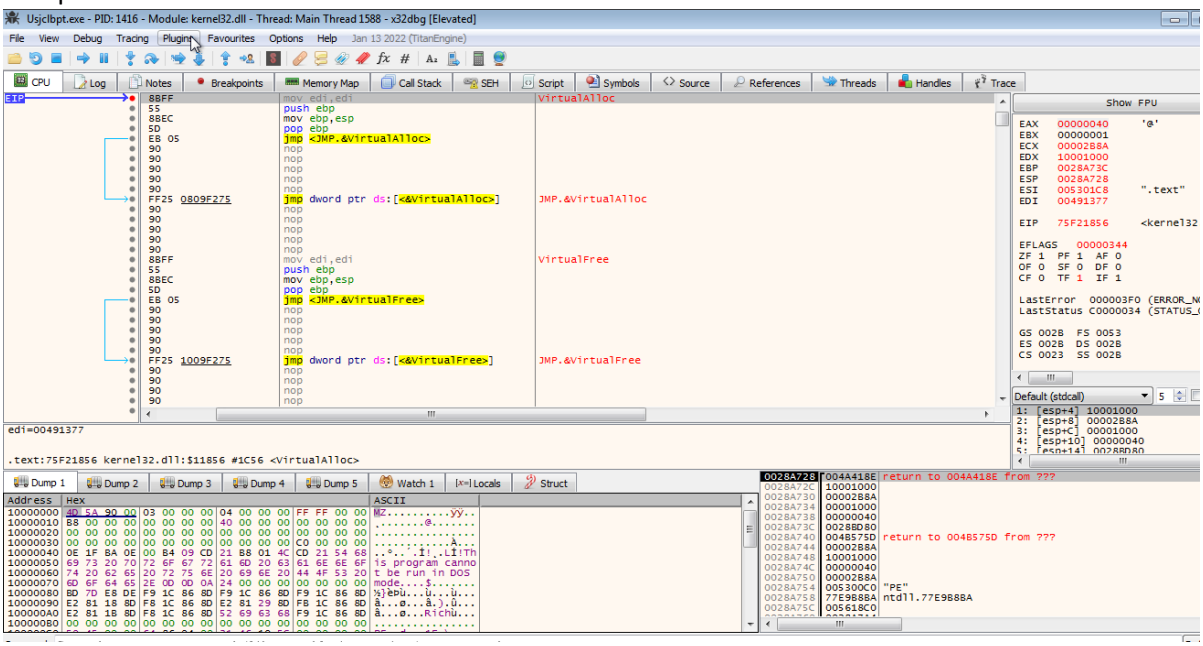
Here just step over it



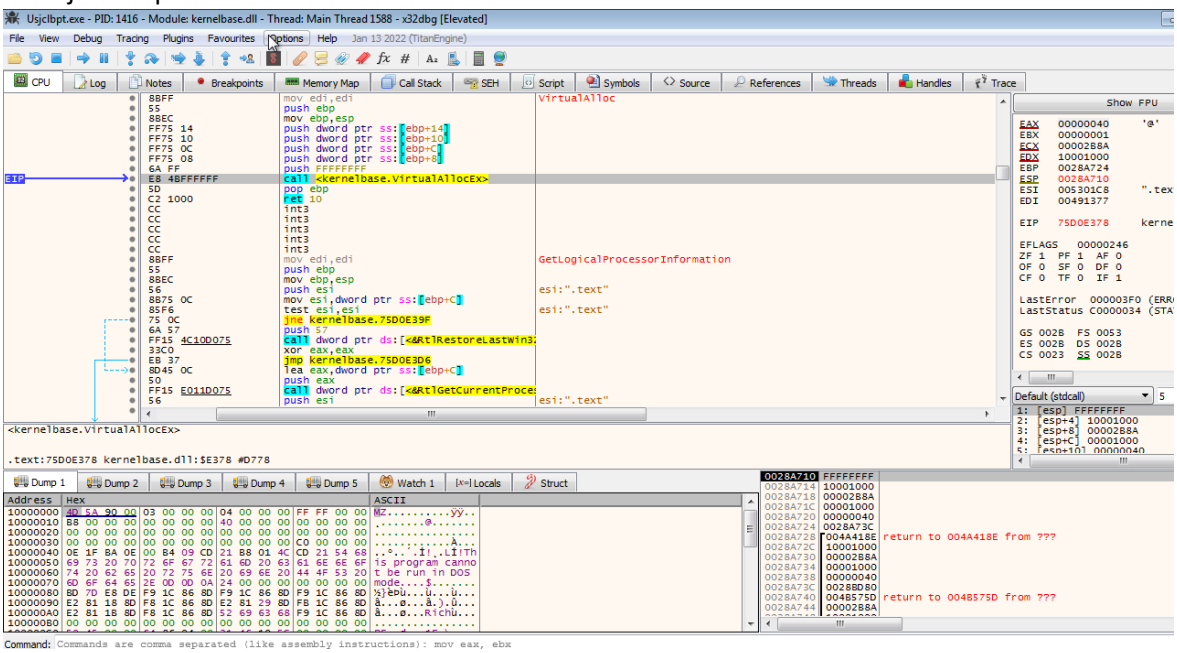
Load the address return by it in dump1 and press f9



We hit VirtualAlloc step over it and also the dump1 has been filled and load the address return by it in dump2

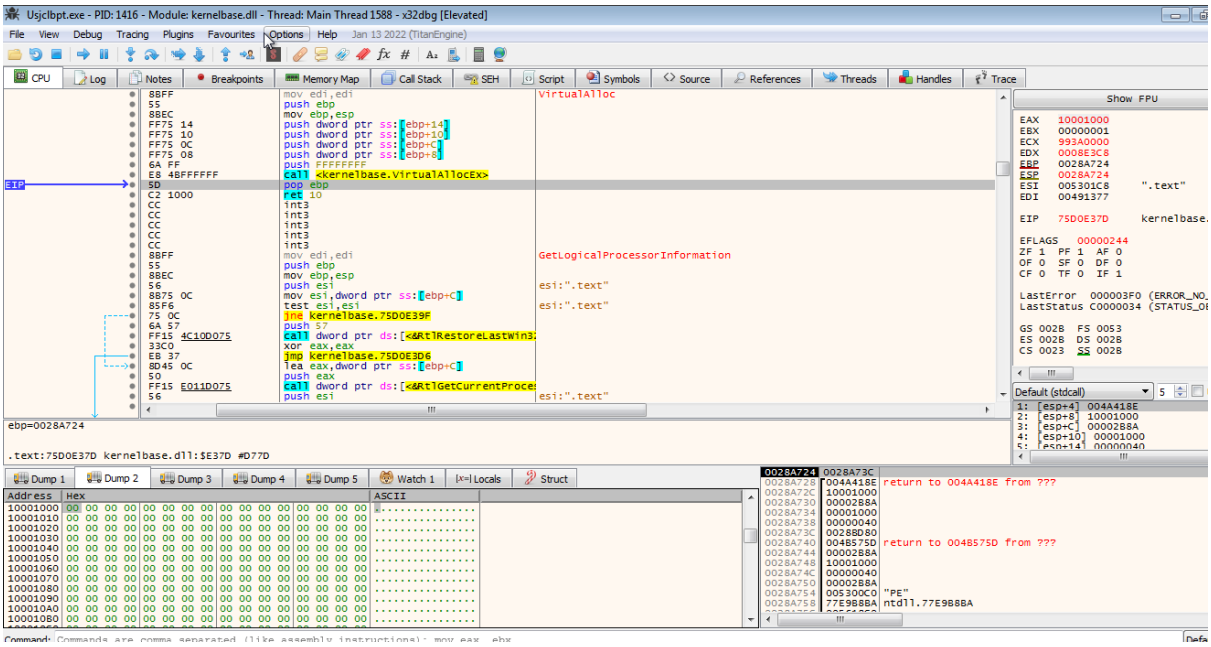


Here just step over it

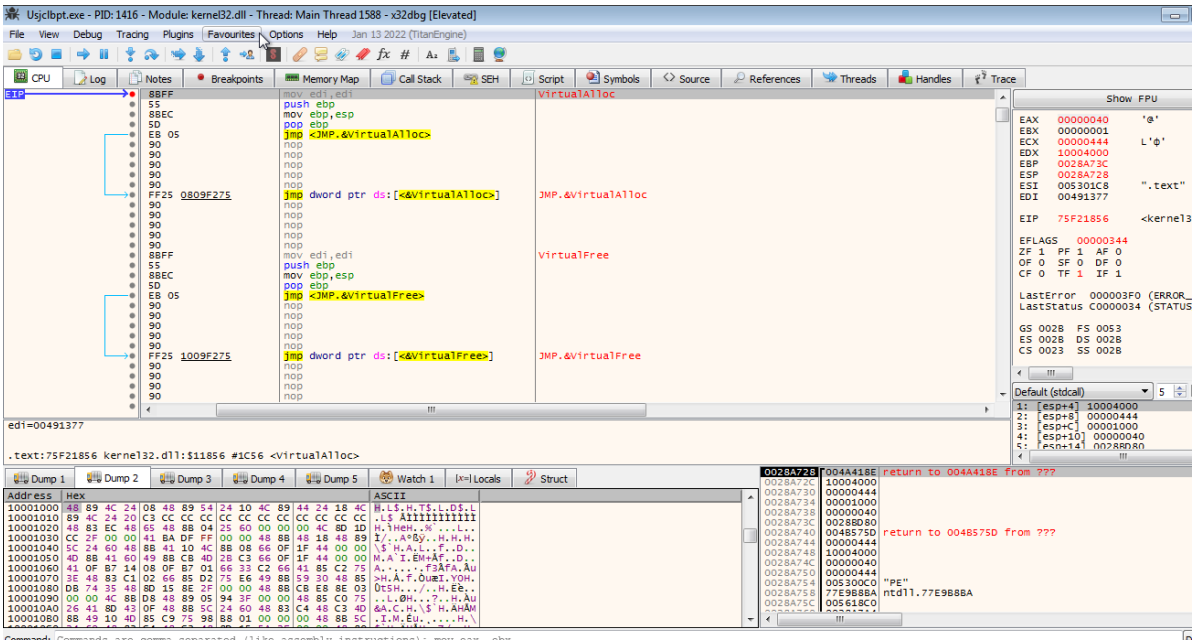


Load the address return by it in dump2 and press f9

Note:Here the address return is in continuation to the previosuly allocated chunk

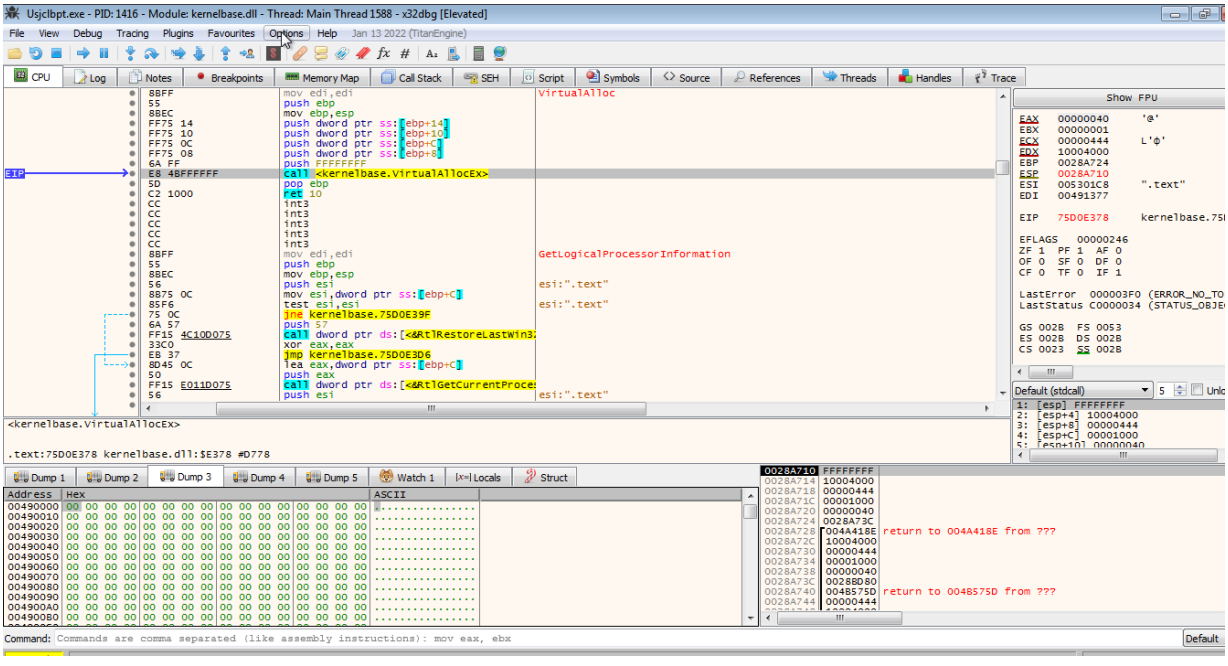


We hit VirtualAlloc step over it and also the dump2 has been filled and load the address return by it in dump3

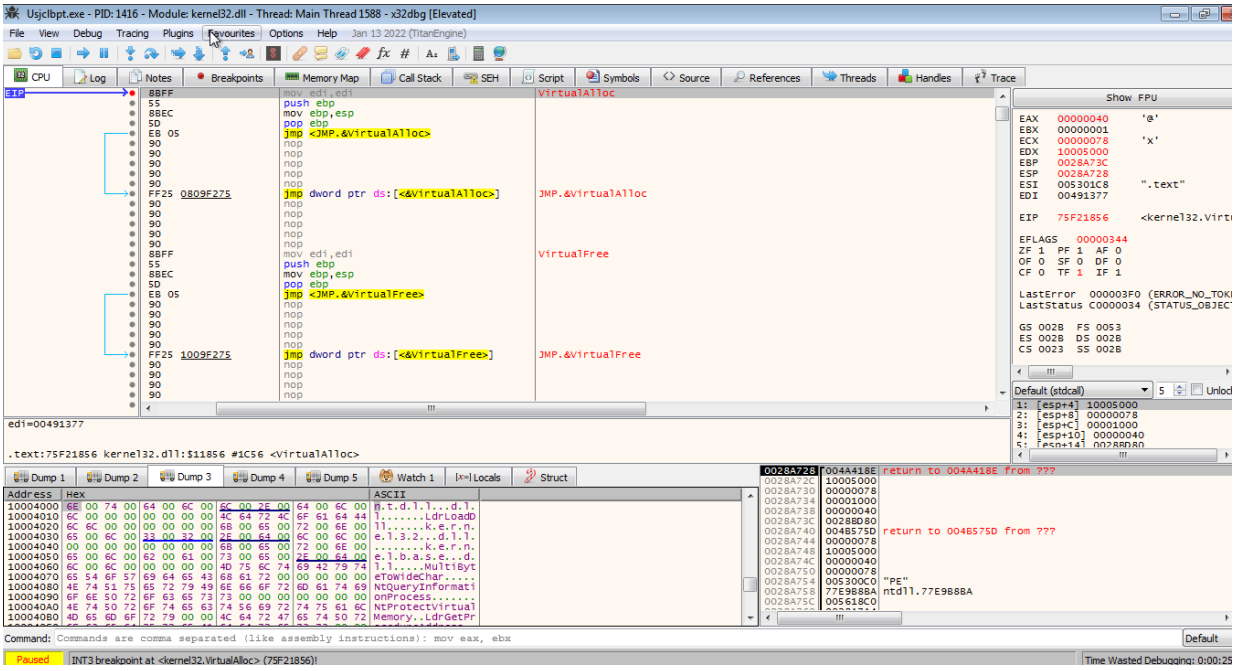


Here just step over it and load the address return by it in dump3.

Note:Here the address return(10004000) is in continuation to the previosuly allocated chunk(10001000)

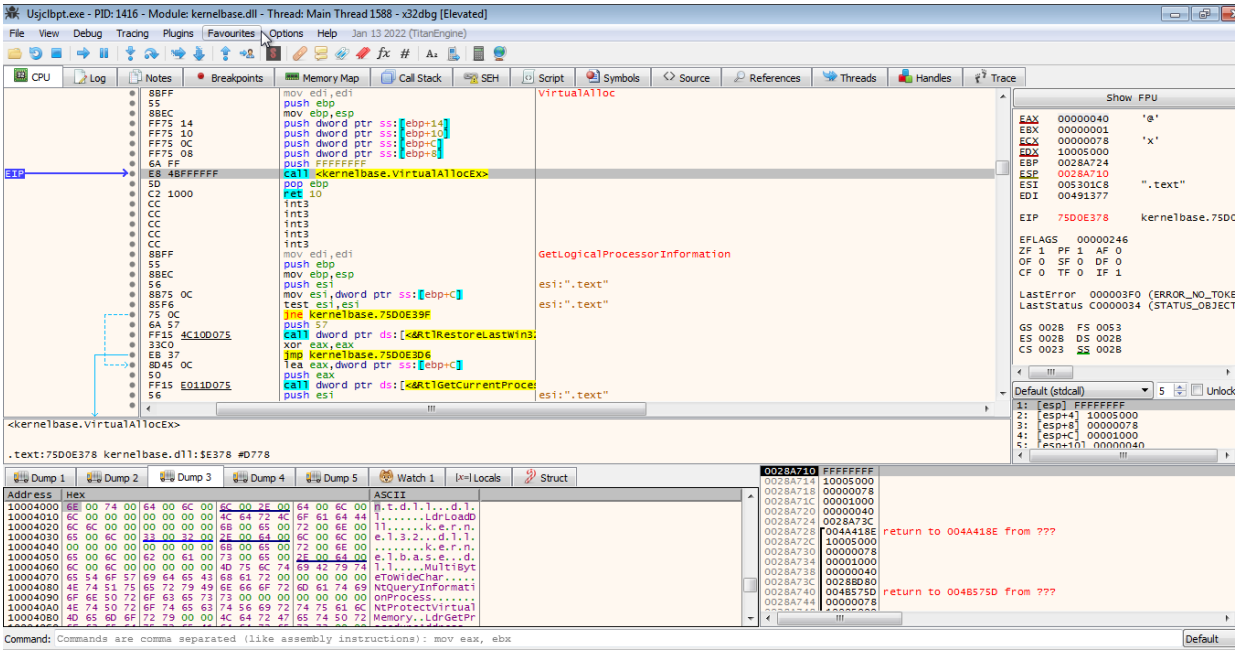


We hit VirtualAlloc step over it and also the dump3 has been filled and load the address return by it in dump4

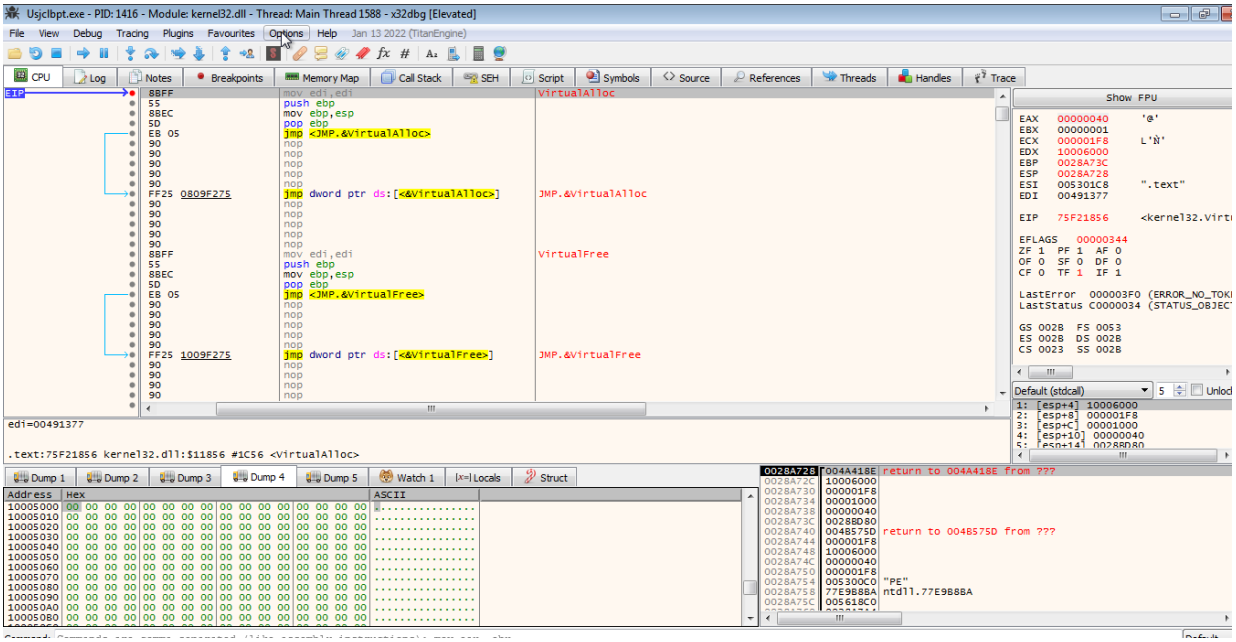


Here just step over it and load the address return by it in dump4

Note:Here the address return(10005000) is in continuation to the previosuly allocated chunk(10004000)

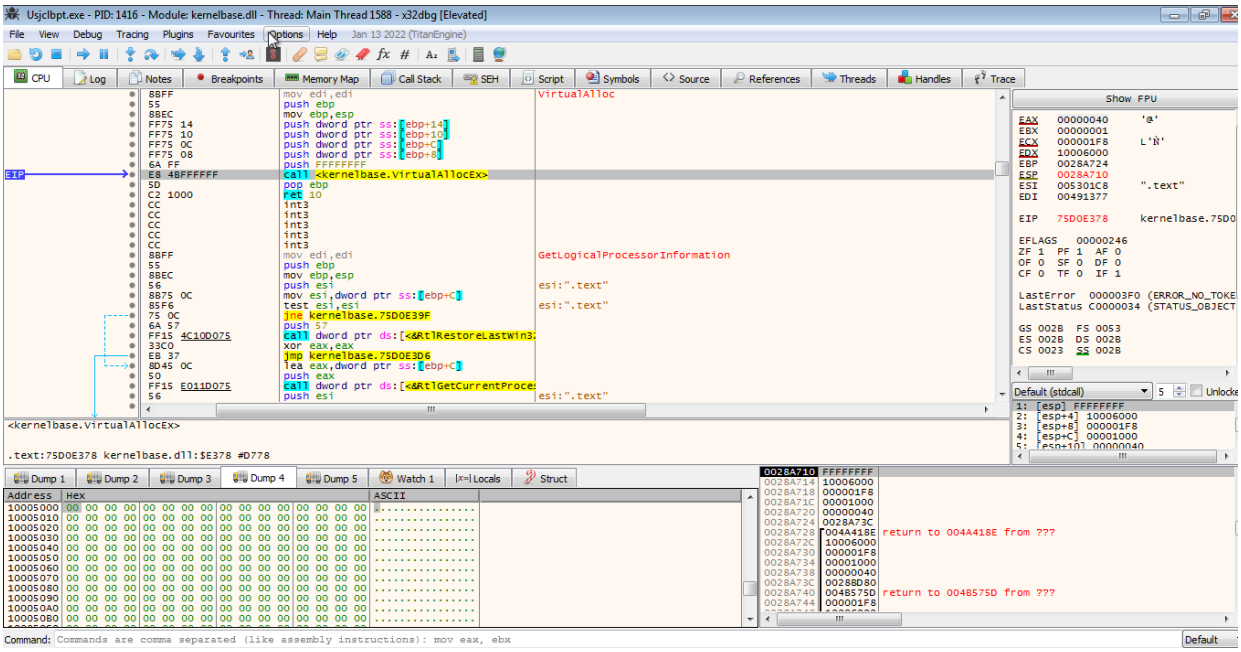


We hit VirtualAlloc step over it and also the dump4 is still empty and load the address return by it in dump5

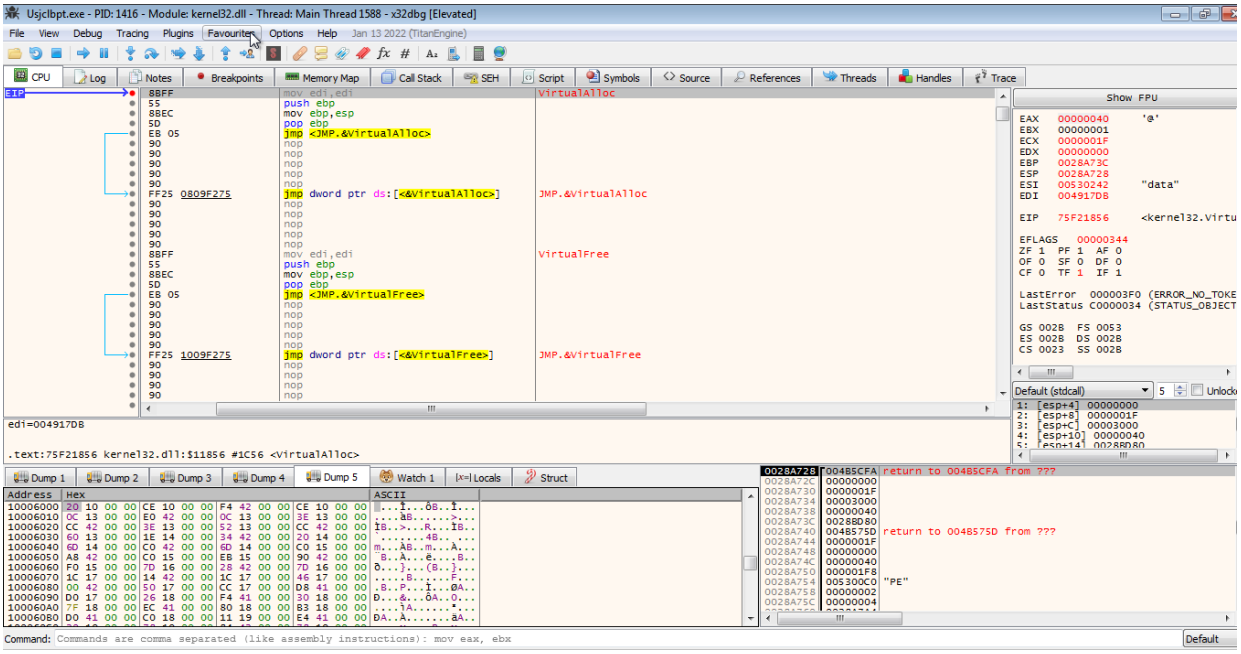


Here just step over it and load the address return by it in dump5

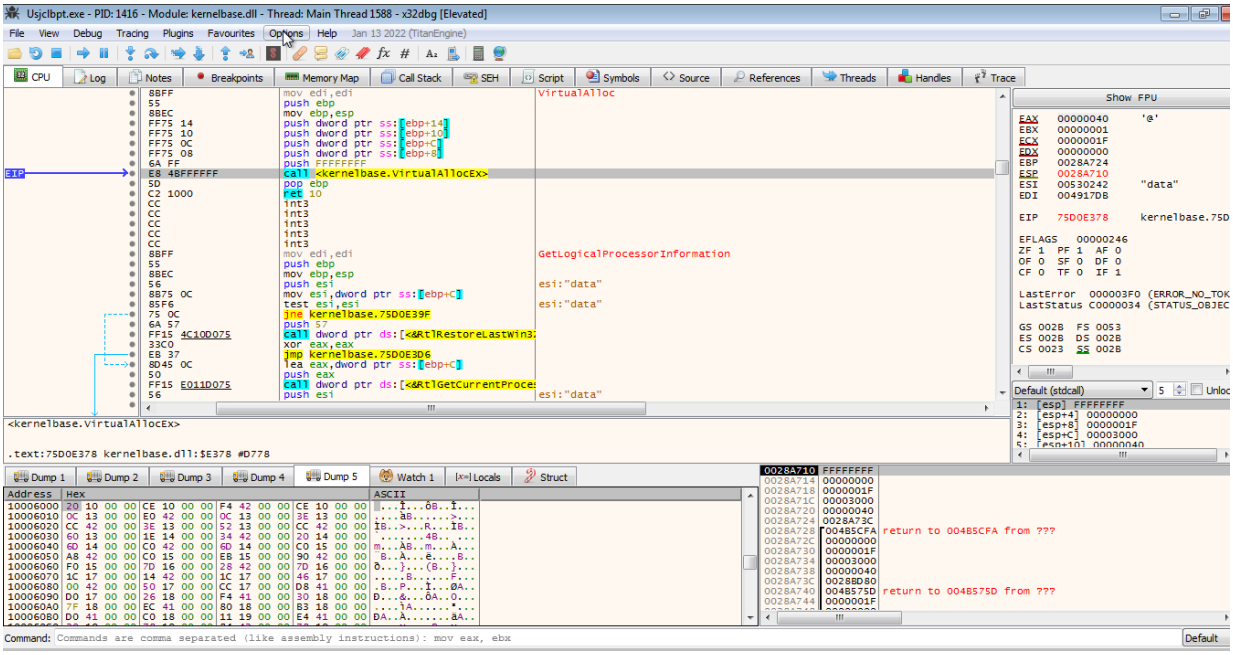
Note:Here the address return(10006000) is in continuation to the previosuly allocated chunk(10005000)



We hit VirtualAlloc step over it and also the dump5 filled with shellcode and load the address return by it in dump1

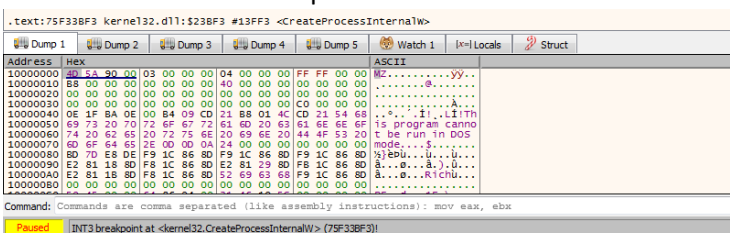


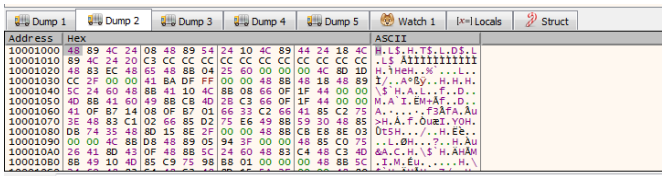
Here just step over it and load the address return by it in dump2

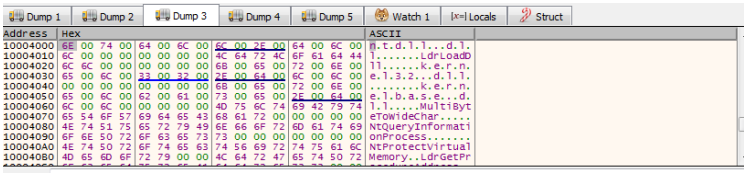


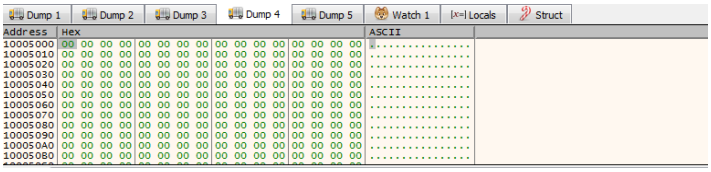
Here just step over it and load the address return by it in dump, Note:Here the address return is not in continuation to the previosuly allocated chunk so it means it has finished unpacking so we can ignore this.

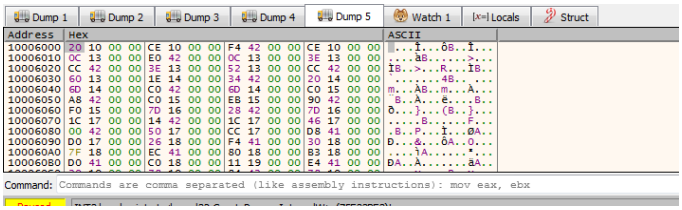
Note:these are the dumps with continuous addresses.



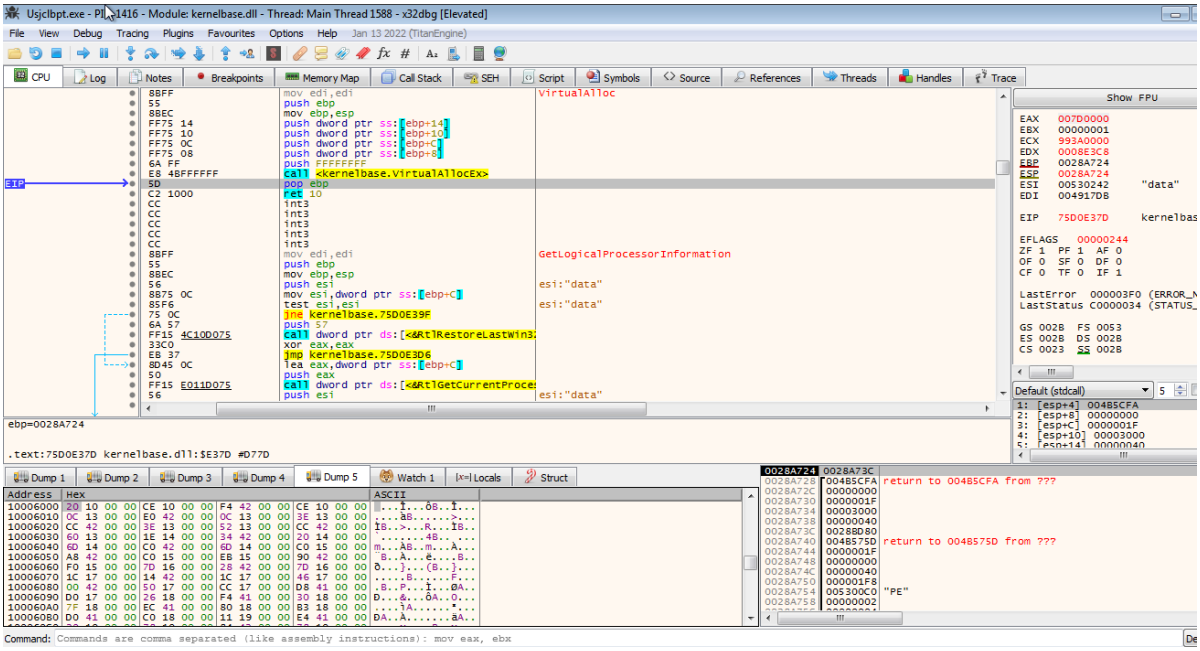




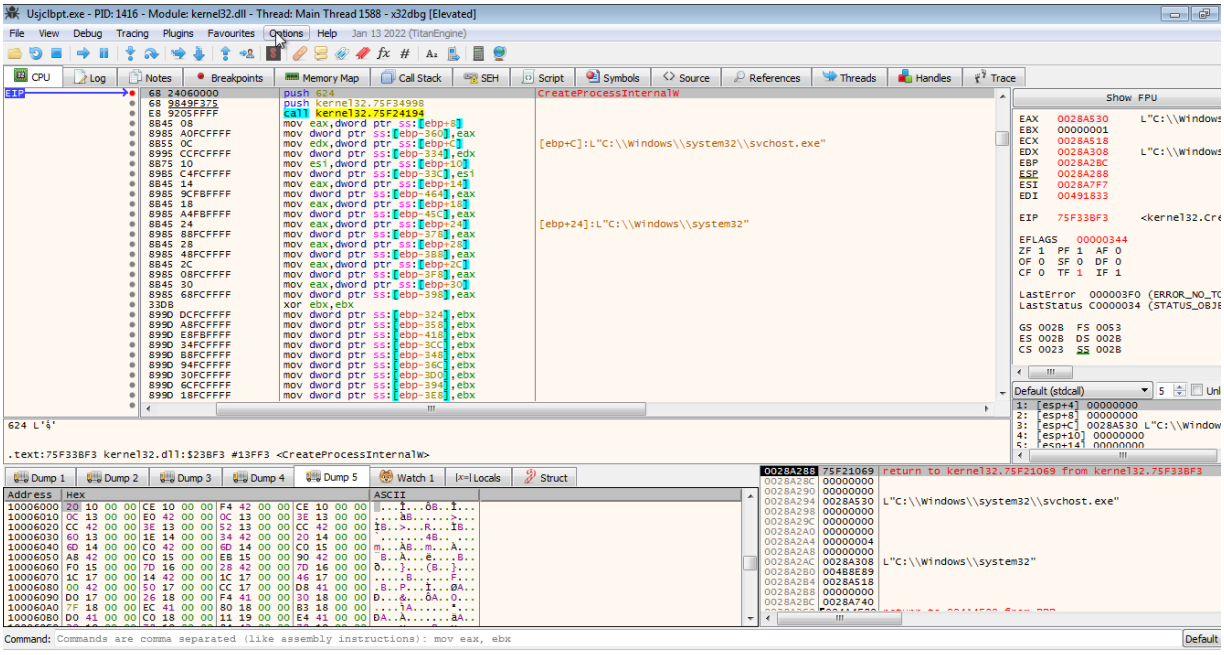




.

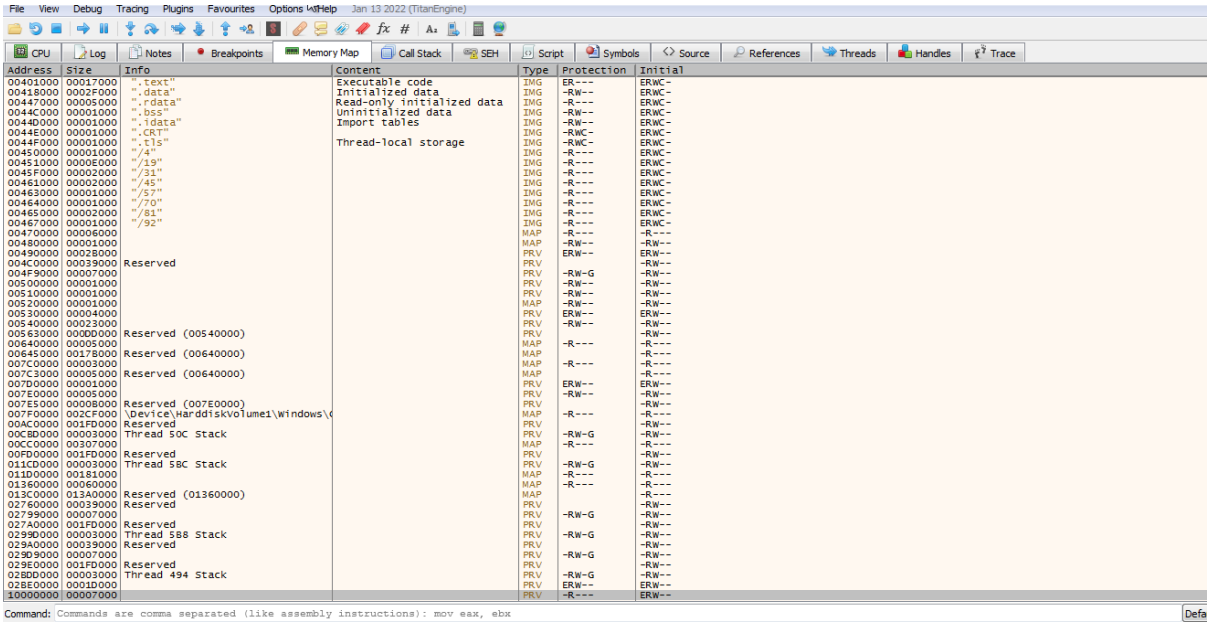


Now we have hit breakpoint at CreateProcessInternalW(using this API it has created svchost after this it will write the previously unpacked code in it)

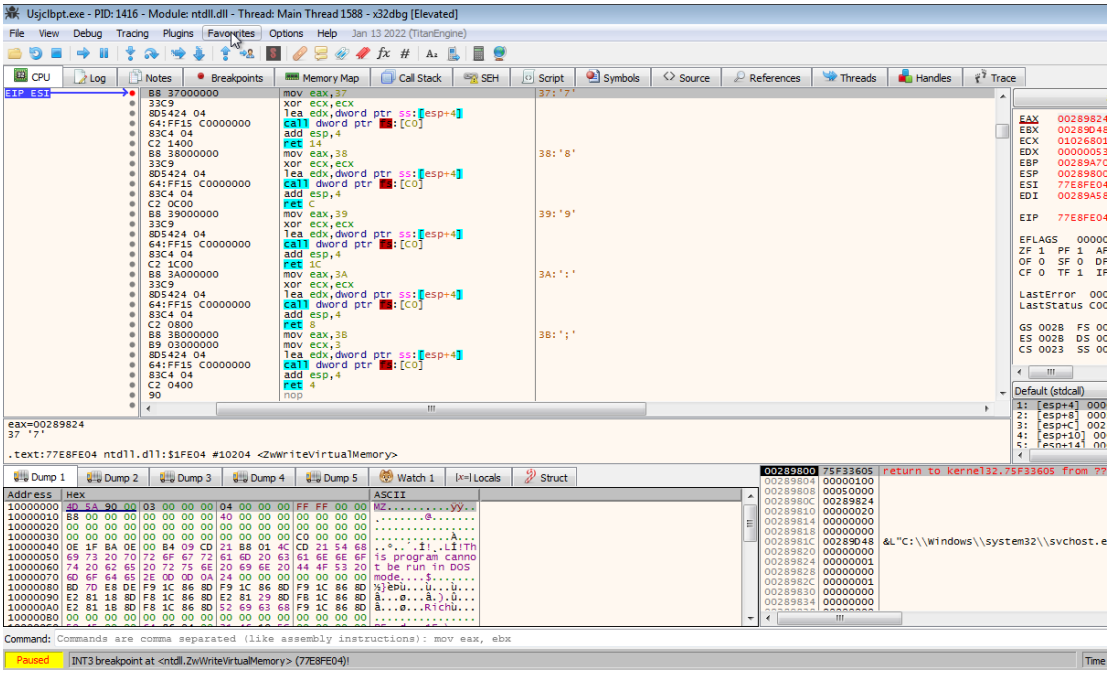


Now we are ready to dump, open dump1 and right click on memory 10000000 and open it in memory map.

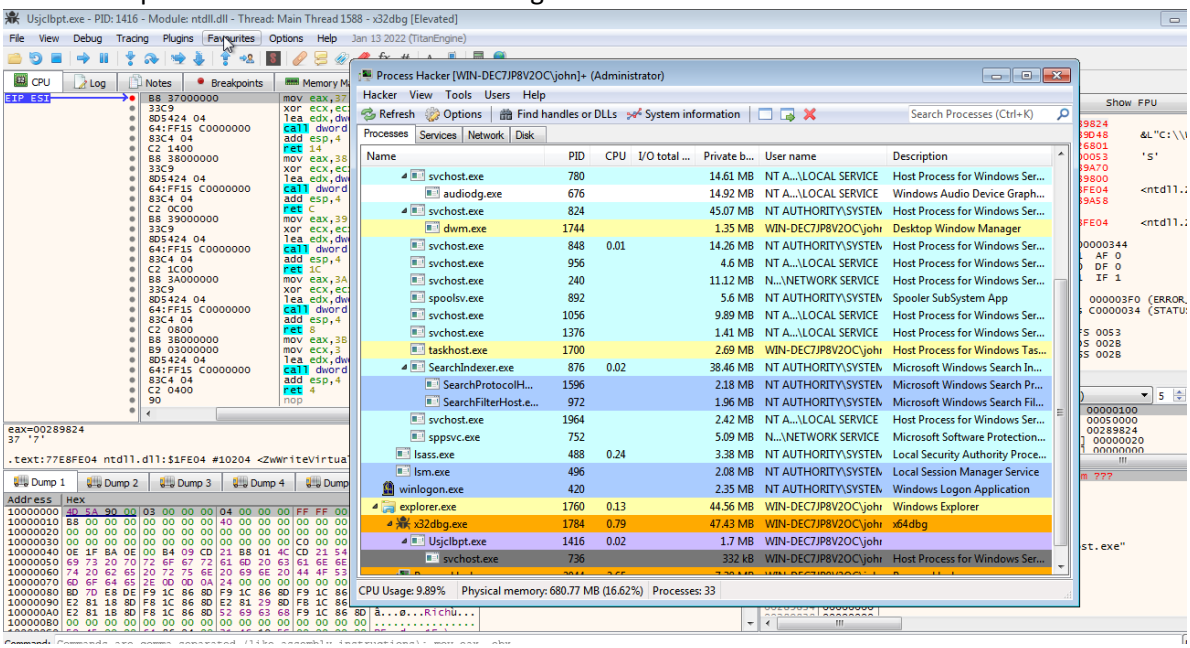
Now from memory one can directly dump that it into a file , by write clicking on Address 1000000.Now press f9.



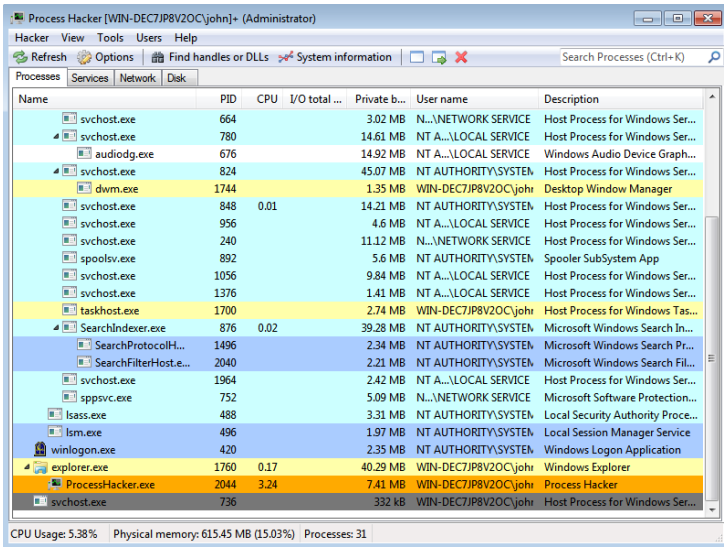
Now we can note that ZwWriteVirtualMemory it is going to write it into svchost, can verify it by opening process hacker.



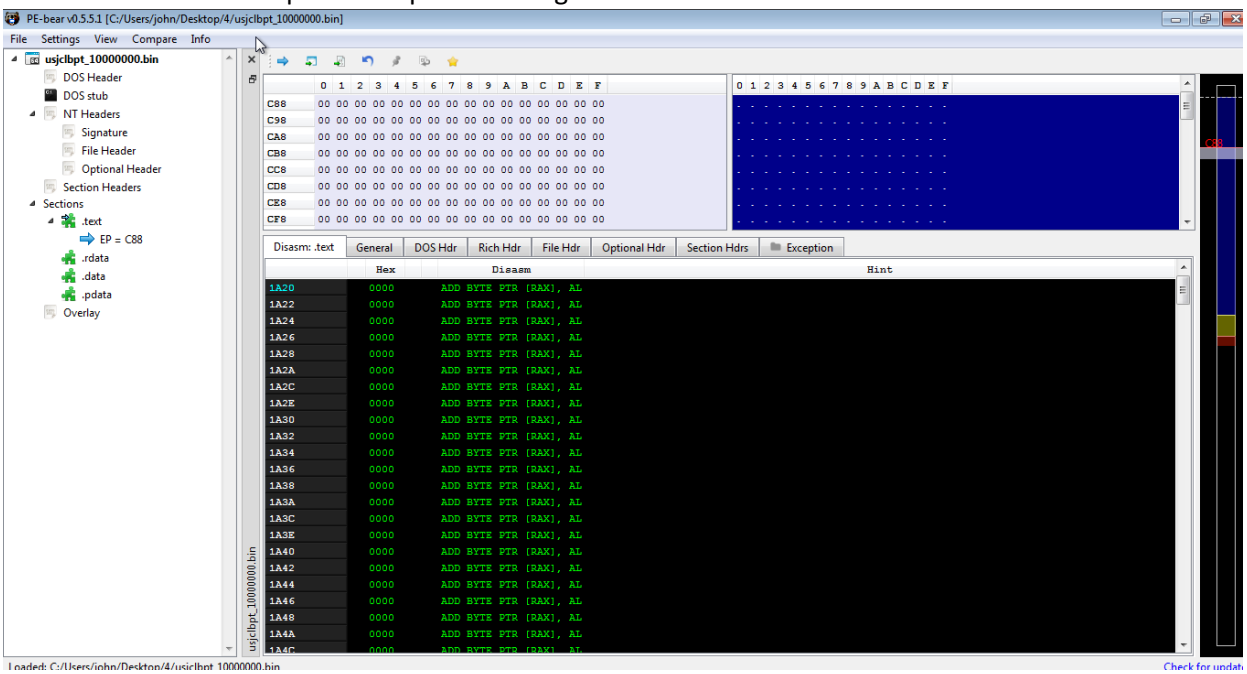
Now if one presses f9 after sometime it will get terminated and a new svchost will start.



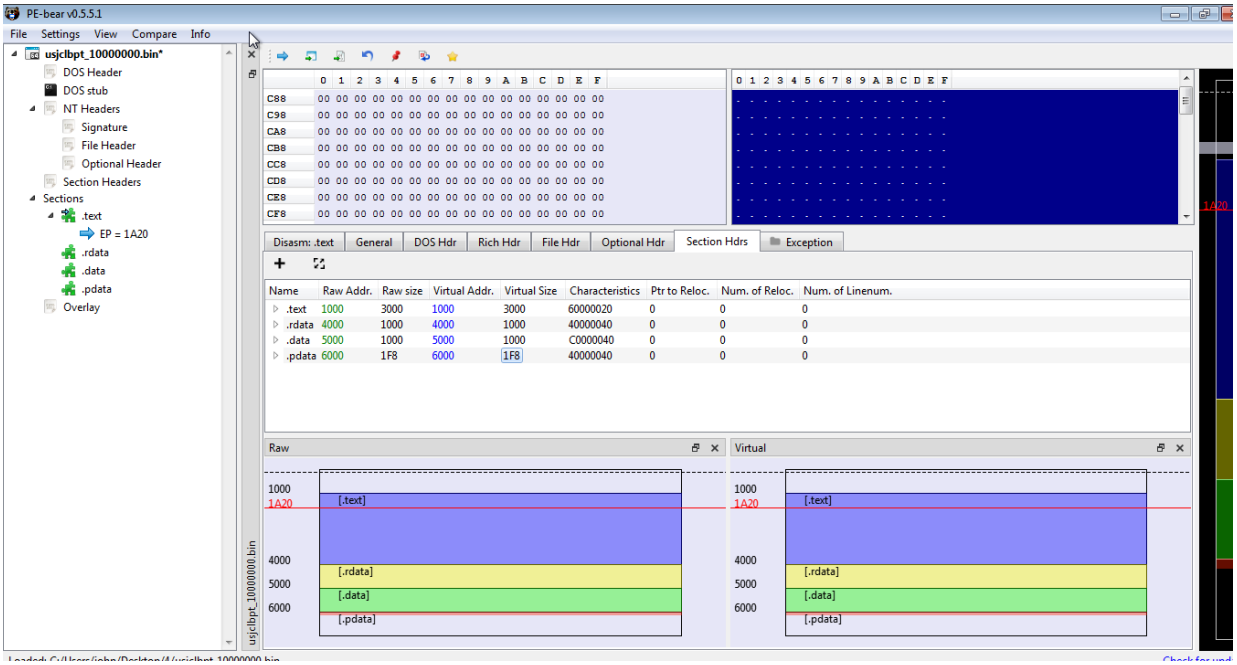
Here the new svchost has started.



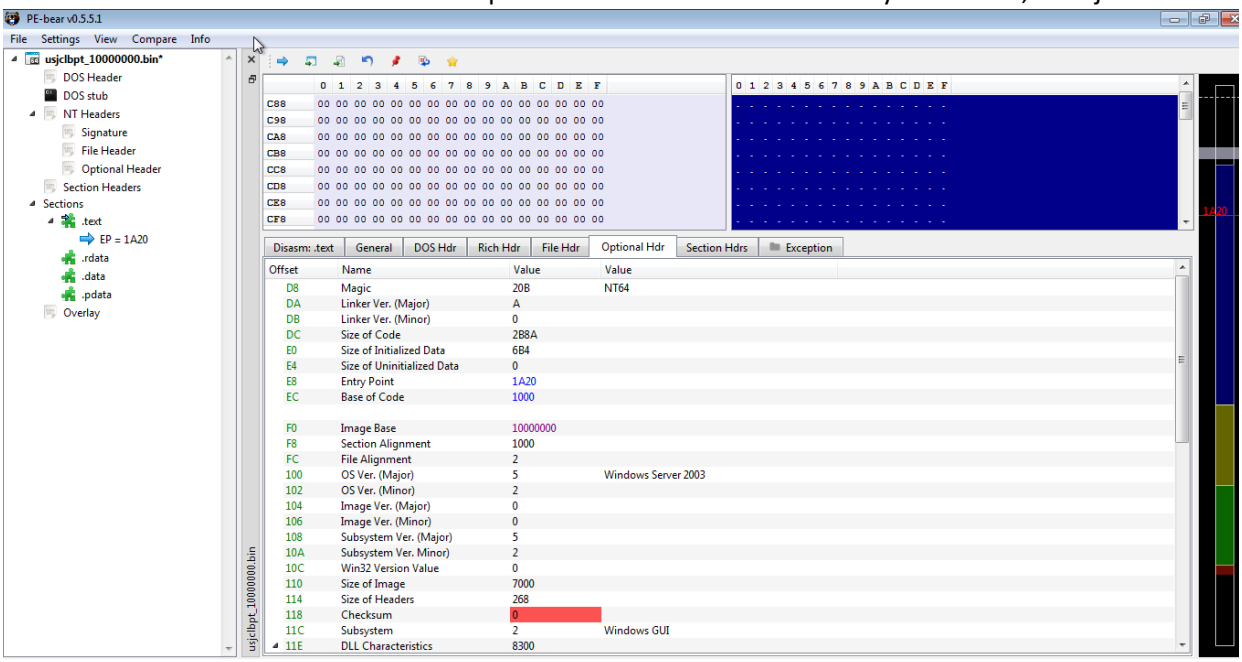
Now here we will unmap the dumped file using PE-bear.



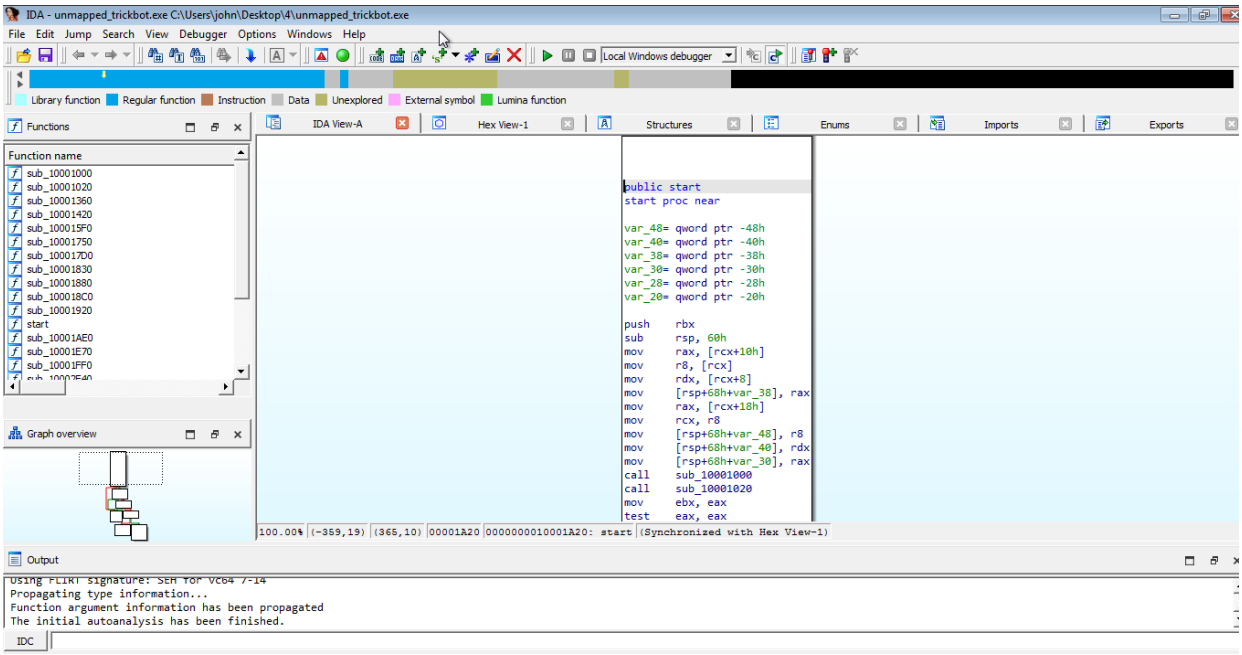
Now here in Section Hdrs tab we need to make raw address same as virtual address , and then calculate the raw size, and then make the virtual size same as raw size.



Now we will check the base address in Optional Hdr tab which was already 10000000, now just save it.



Load the newly dumped file in IDA.



There are no imports here as this trojan will dynamically imports the library as it runs.