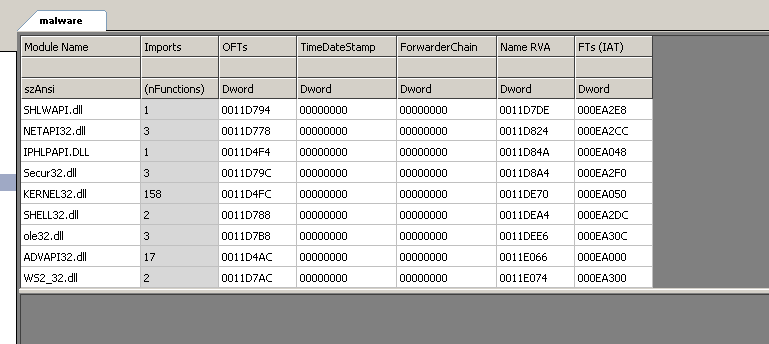
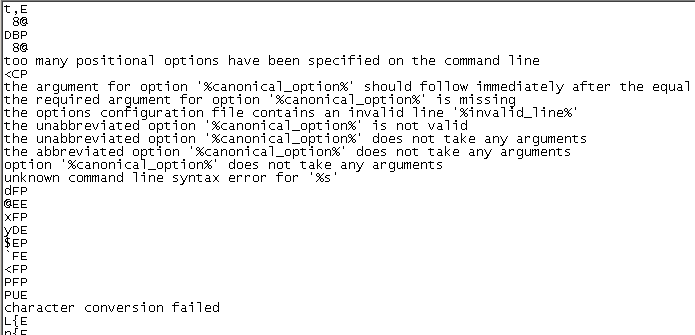
# Malware Analysis: LockerGoa.exe

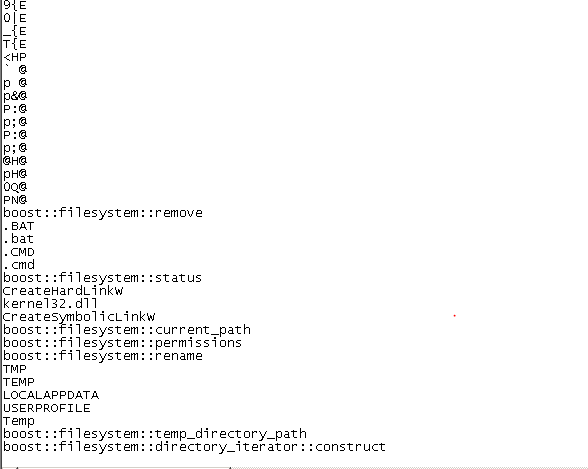
**Static Analysis:**

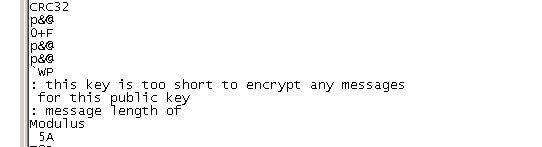
* Pe32
* Visual C++
* Alisa LTD
* Many Imports so may not be packed, possible web, registry access

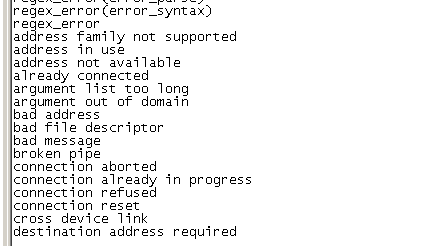


* Strings, may take command line arguments, filesystem interaction, cryptography related messages, may bind to address







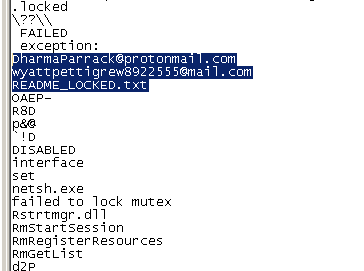


Some interesting email:

DharmaParrack@protonmail.com

wyattpettigrew8922555@mail.com

README\_LOCKED.txt



**Dynamic:**

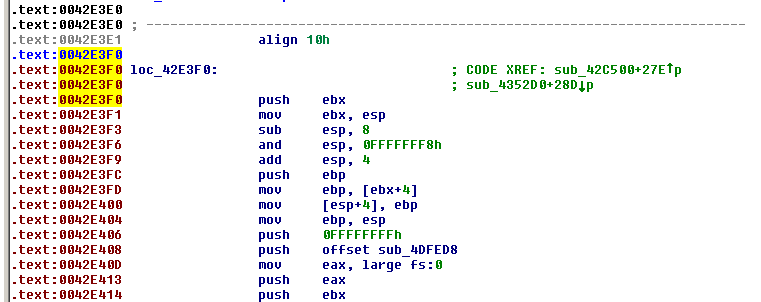
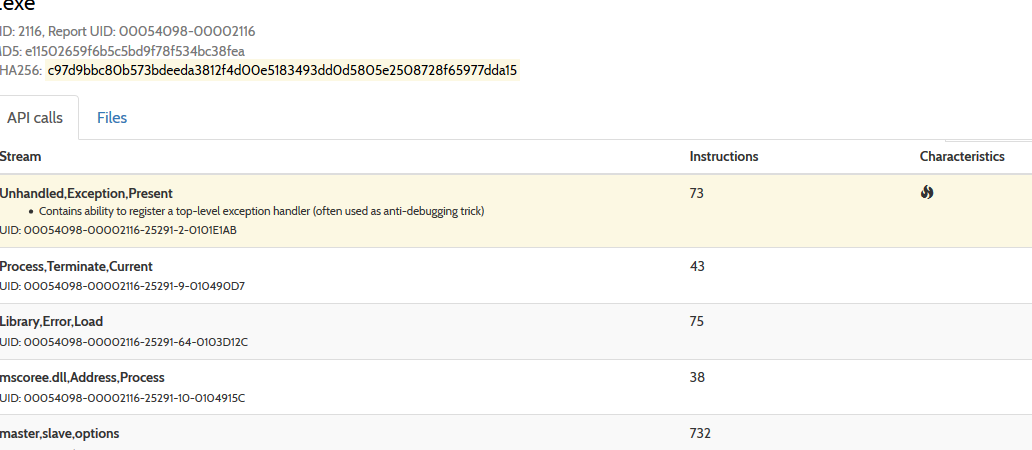
* LockerGoa.exe starts LockerGoa.exe again *CreatePRocess?*
* Then the spawnd process starts cmd C:\Windows\system32\cmd.exe /c move /y C:\Users\admin\Desktop\LockerGoga.exe C:\Users\admin\AppData\Local\Temp\tgytutrc8131.exe (*shellexecute?*)
* Started the tgyutrc8131.exe process (*CreateProcess?*)
* The tgy.. process starts logoff.exe,net.exe (adds new user admin I guess) again starts itself again, and finally notepad.exe readme file (*CreateProcess? And ShellExecute?*)
* No network connection and registry changes

**Reversing Malware:**

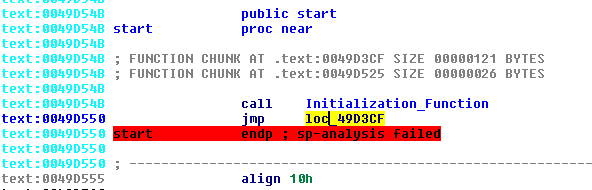
So we know that the first thing the malware does is create process, lets search for createprocessA/w

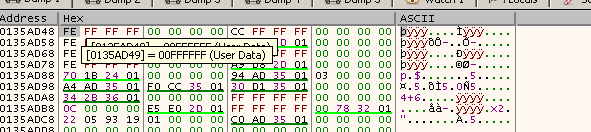
A lot of places. Also when I ran the malware, it just crashed

In start first analysis failed, changed to code and then just jump

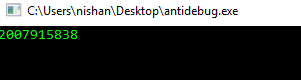
* 0042E3F0 treated as data
* I need to probably run a debugger and see where it goes
* Already analysis in hybrid, SO I look here, there is antibugging trick (I thought initially), IsProcessorFeaturePresent, (What I though it seems the malware just crashed because it could not copu lul)

I searched in IDA and went up, In start first analysis failed, changed to code and then just jump

* 
* Then the address 0051AD48 is being pushed to a function, so I debugged to see whats happening, address different in debugger, IDK why?



Before and after no change, function called at 0049D8C3 which calls IsProcessFeaturePresent.

* I used In a sample program, and both returned same value in host, but when debuggint sample in guest vm returned 1, os then I wrote the program and compared the value that I got outside vm with inside vm
* Outside vm 
* This function is used to check if a process feature is present, the values passed are 10

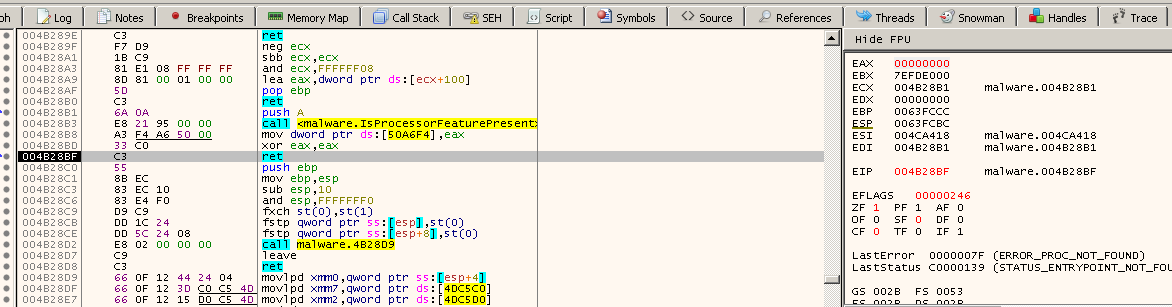
|  |  |
| --- | --- |
| **PF\_XMMI64\_INSTRUCTIONS\_AVAILABLE**  10 | The SSE2 instruction set is available.  **Windows 2000:**This feature is not supported. |

And 23 **PF\_FASTFAIL\_AVAILABLE**

23

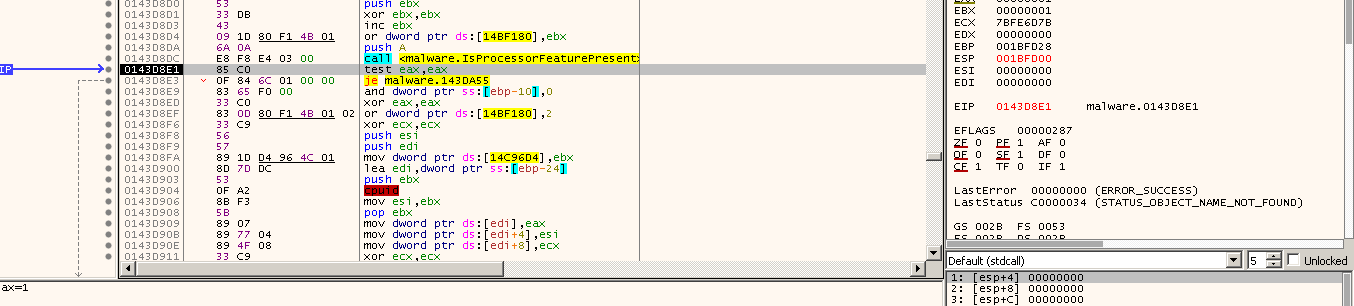
If the feature is supported, the return value is a nonzero value.

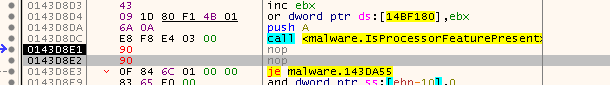
If the feature is not supported, the return value is zero.

* <https://docs.microsoft.com/en-us/windows/desktop/api/processthreadsapi/nf-processthreadsapi-isprocessorfeaturepresent>
* It seems when in vm () feature set is not supported
* When 0xa passed returned 0

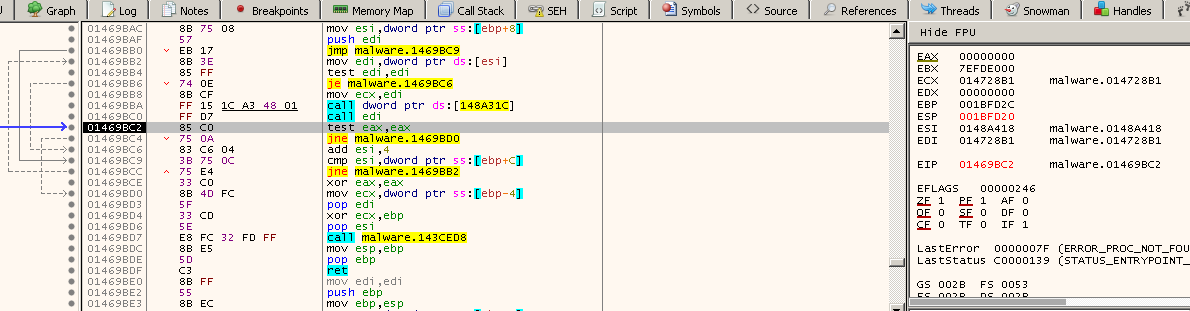
0041F6A0 0041F7F8 L"C:\\Windows\\system32\\rsaenh.dll, I noped it, now it works I guess

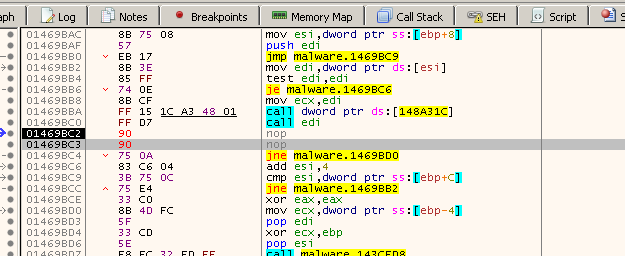
Before:

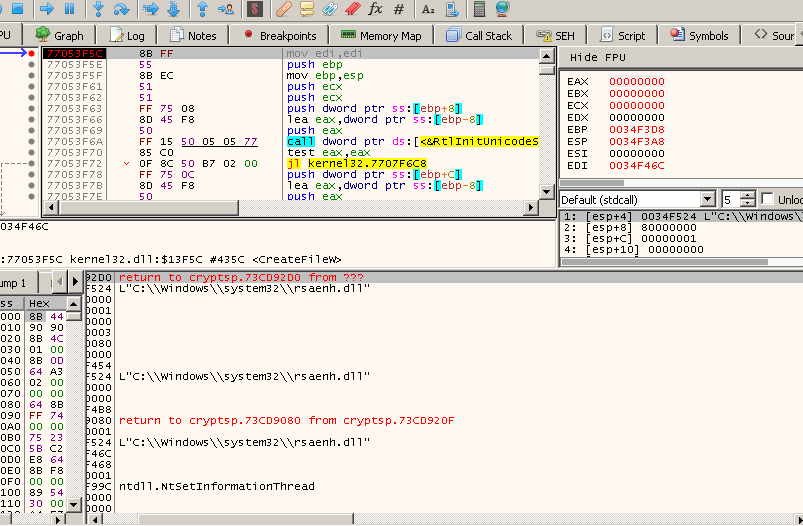




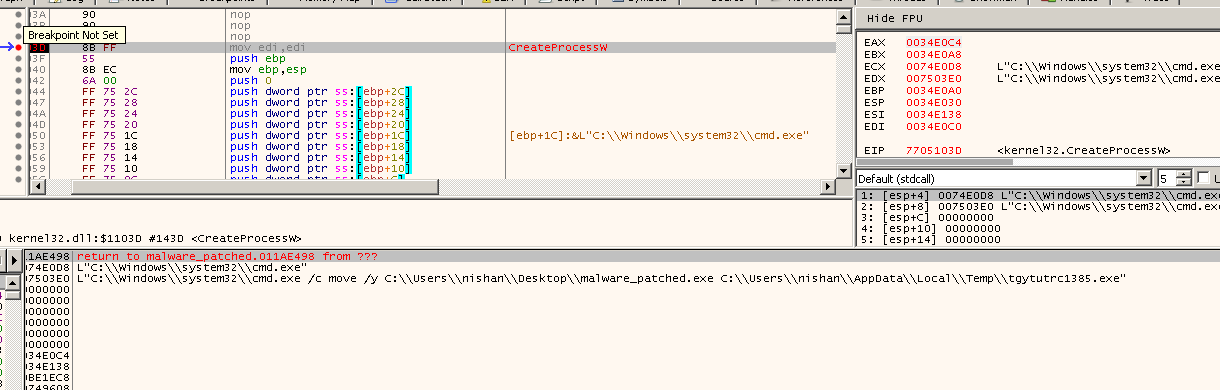
Before:







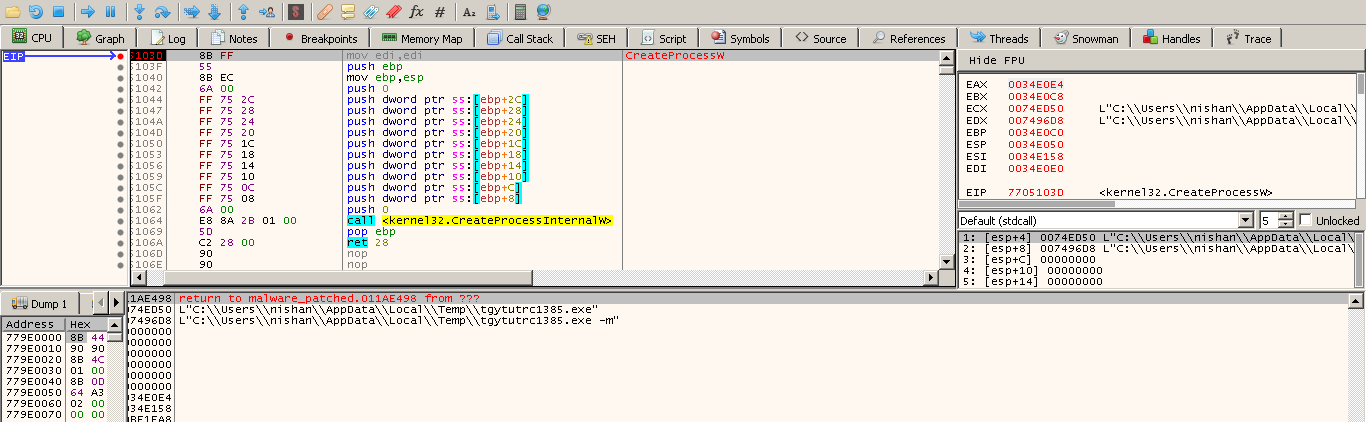
And process executed

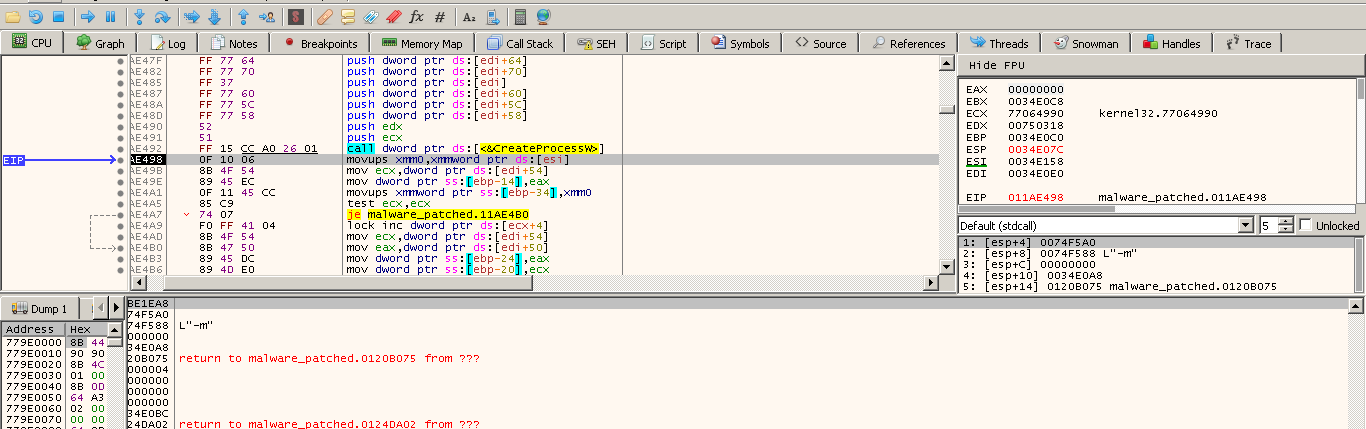


/c execute and terminate /y means backward compatibility

However the file is not created.

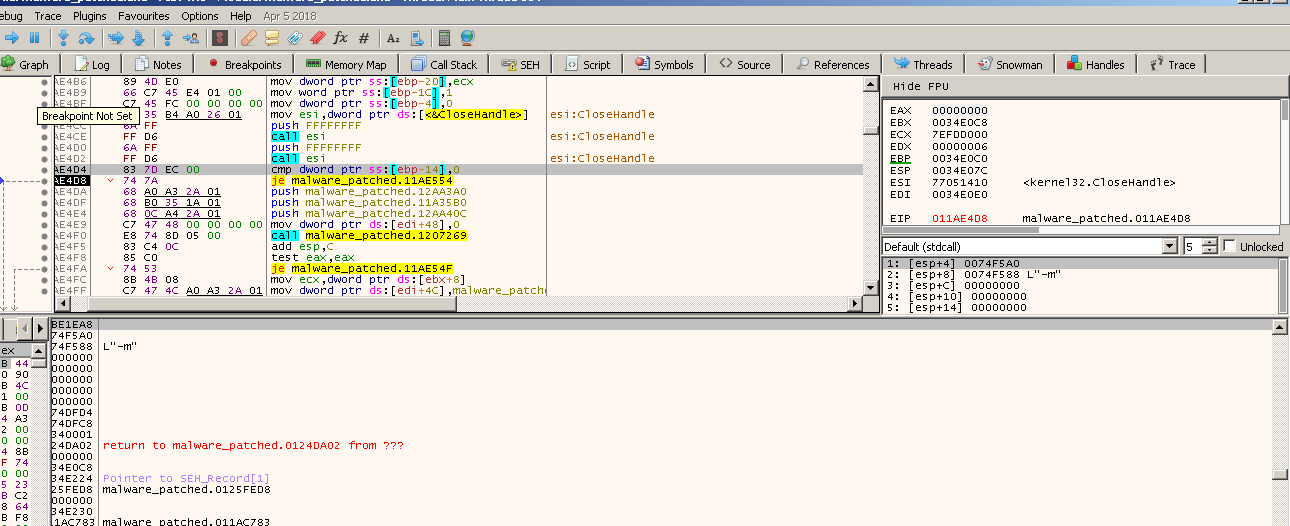
We see that the malware wants to execute

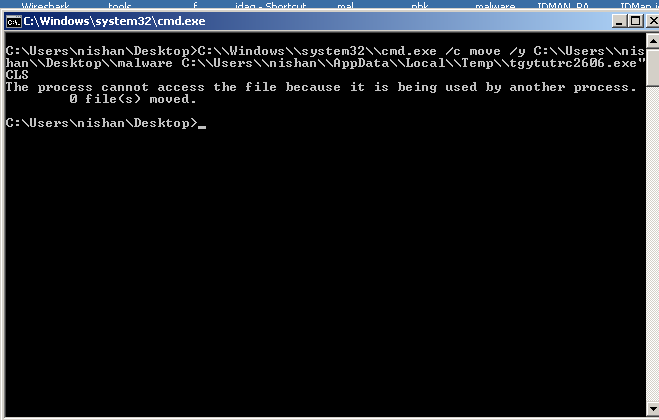
We can see the return value is zero, the function will return zero when could not create process



But the malware seems to be unware of the value in eax is never checked,

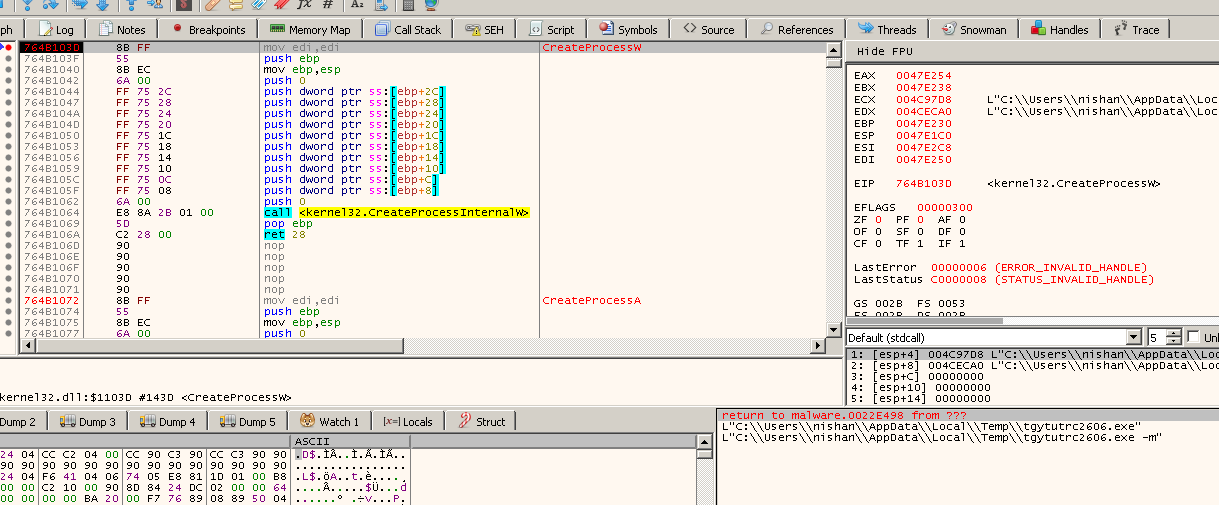
However checks at closehandle

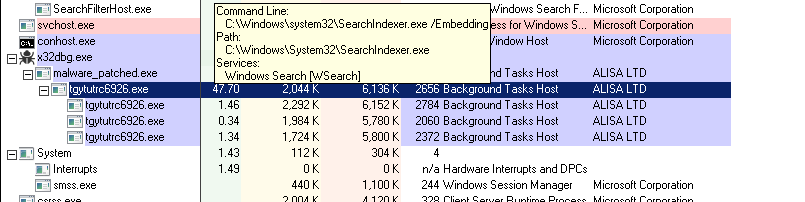




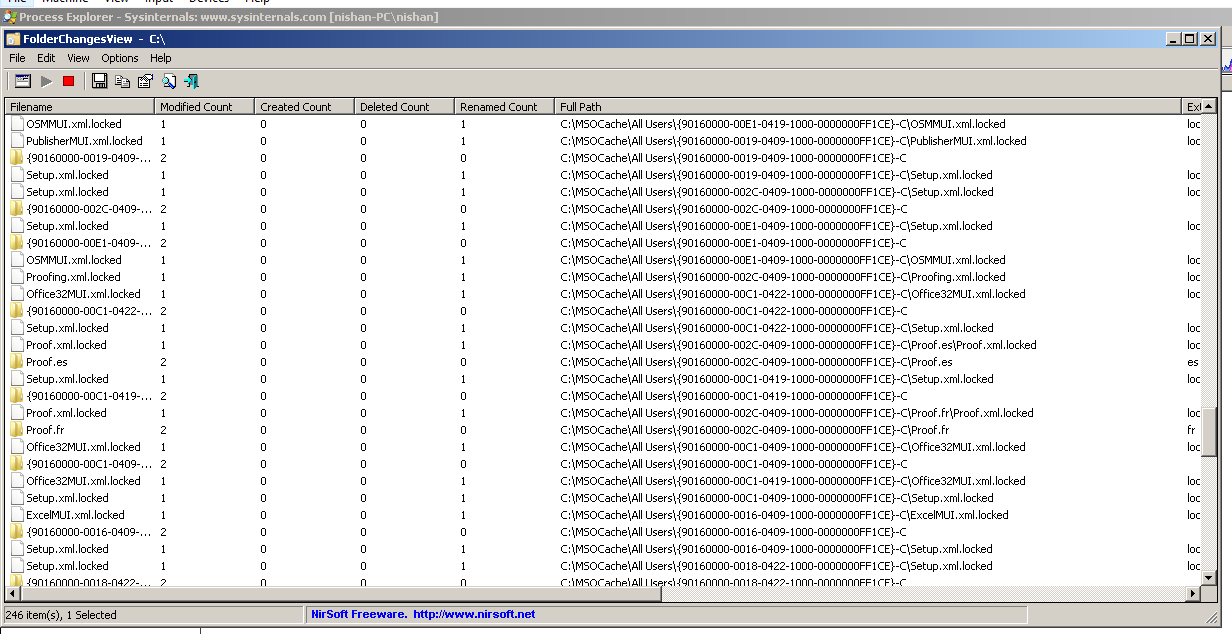
And that is what exits the program, Copied the file myself. THe file is identical, same hash, no change

Executes with –m argument

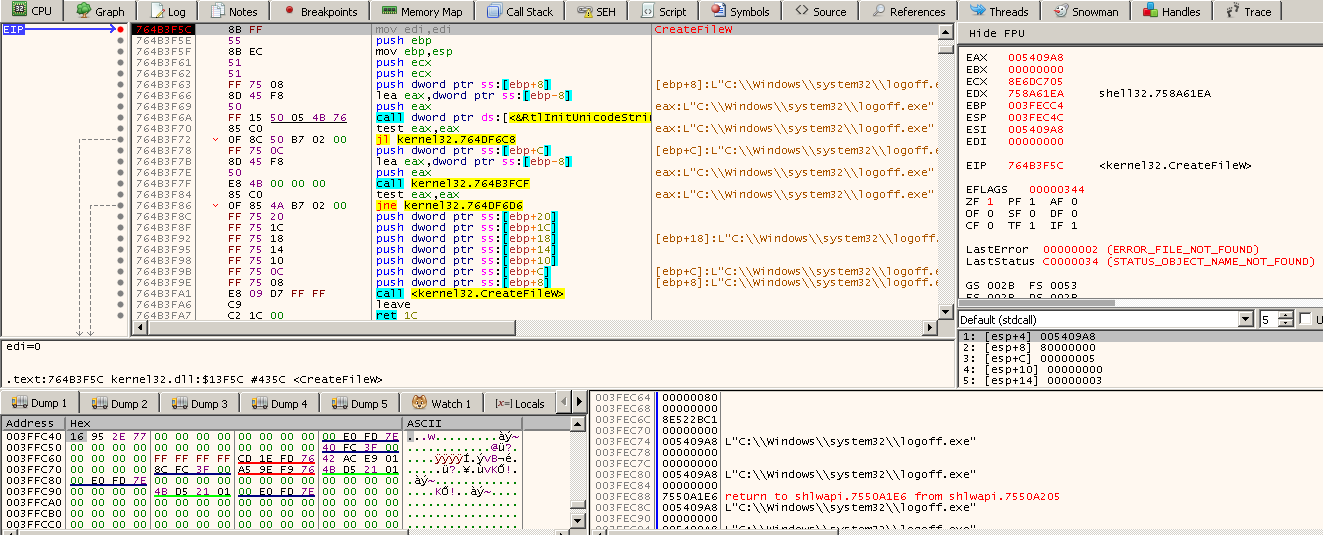




Then the process Creates ReadMe.txt, We can see .lockde Folder being created



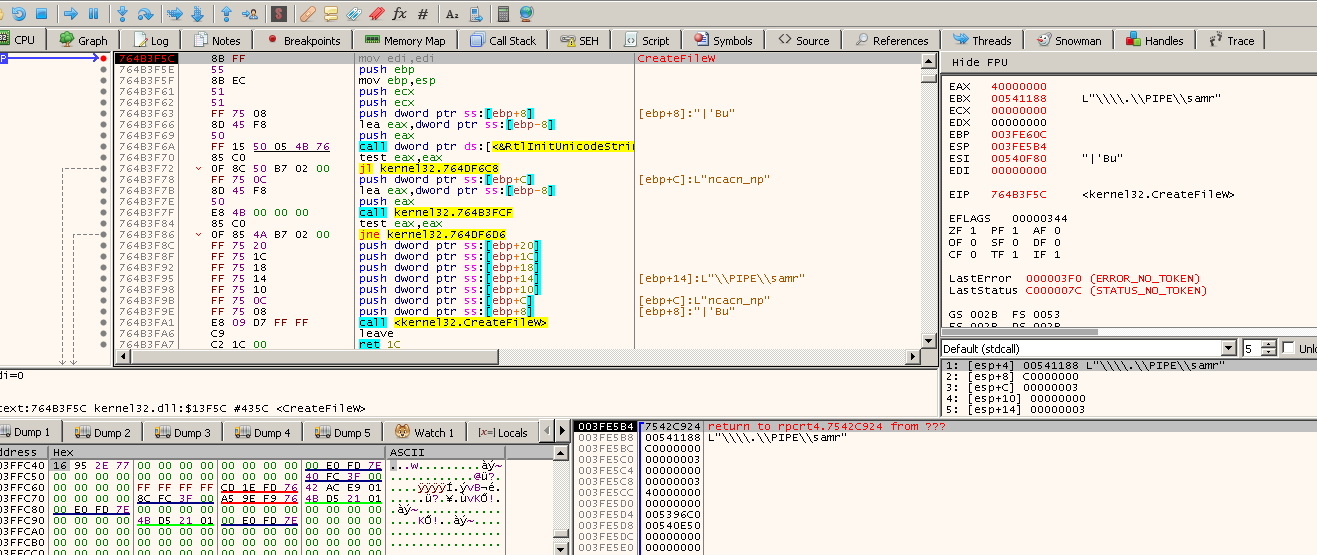
When –m flag is provided logoff.exe createfilew



File was legit.

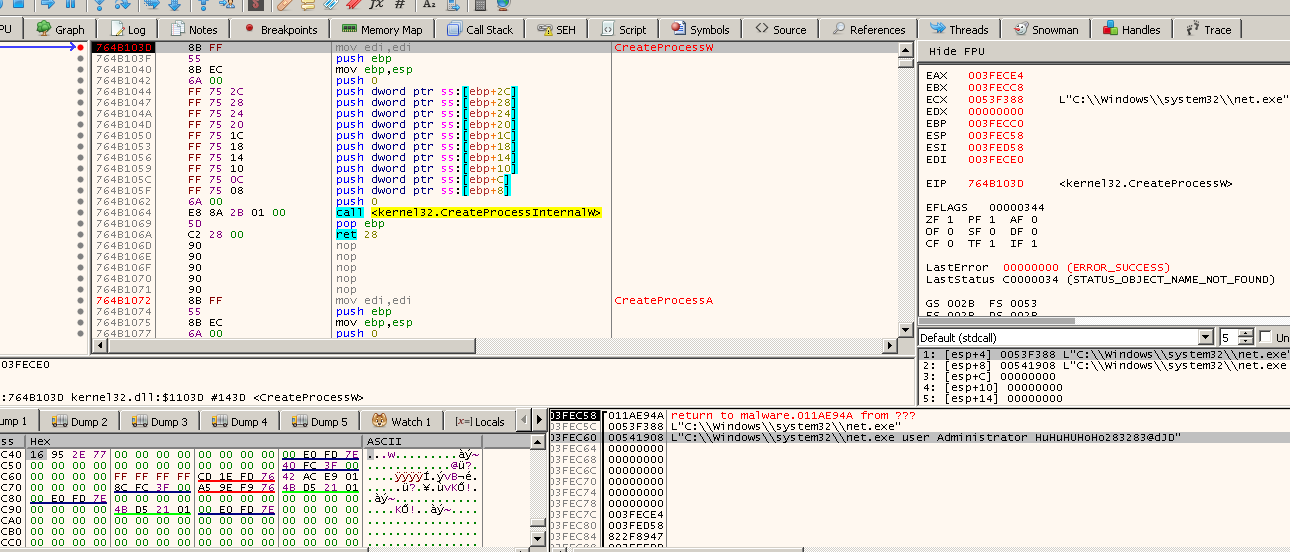
Logoff 0 -> probably session id.

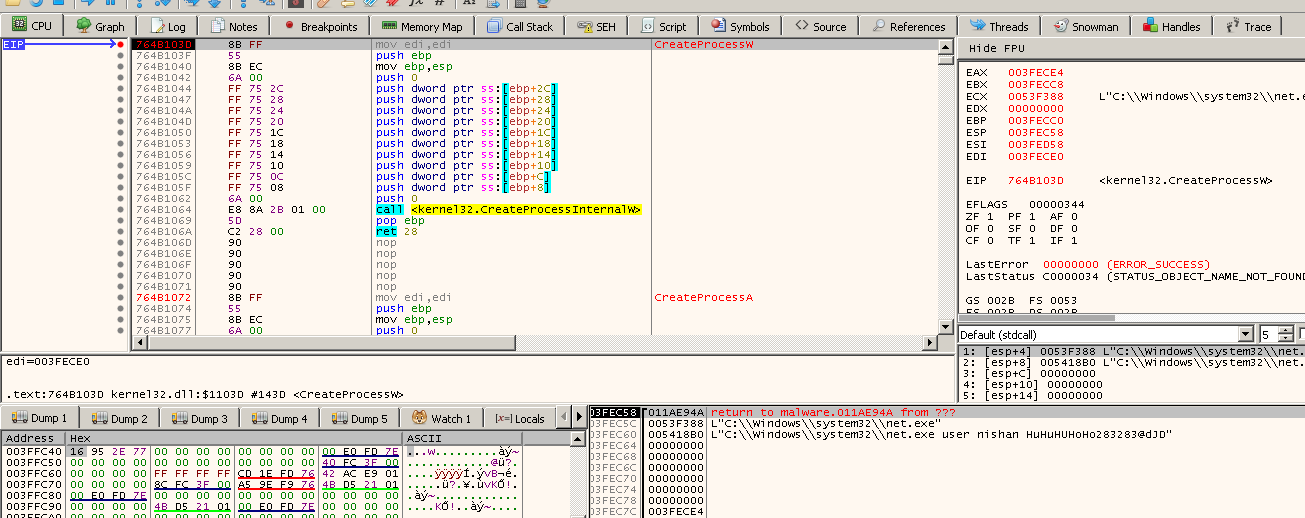
Creates file again



Pipe iA **pipe** is a section of shared memory that processes use for communication. The process that creates a **pipe** is the **pipe** server.

Create file for net.exe

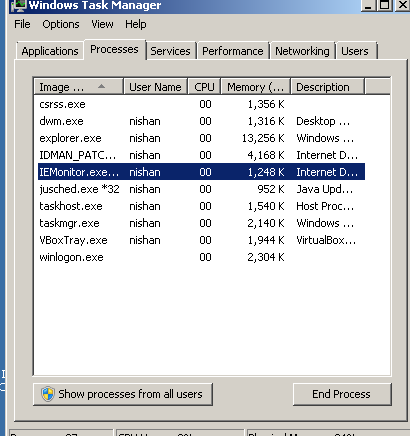




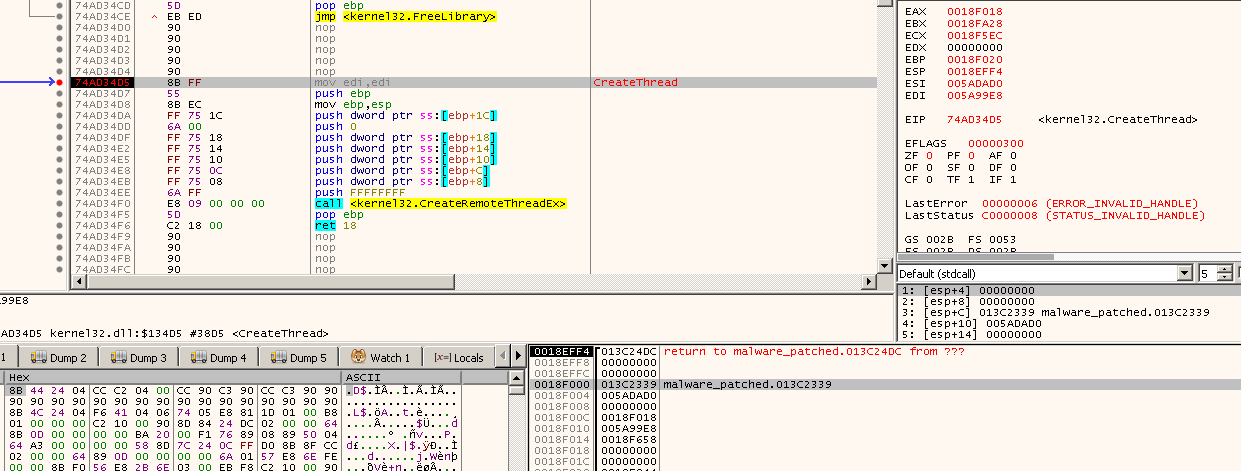
003FEC60 00541908 L"C:\\Windows\\system32\\net.exe user Administrator, also for the user HuHuHUHoHo283283@dJD"

003FEC60 005418B0 L"C:\\Windows\\system32\\net.exe user nishan HuHuHUHoHo283283@dJD".

And yes the user password was changed, but the thing is the malware did not write some program to run at startup, so that is wiered.

7

In the process, IT called Create thread



CreateThread(arg1,arg2,arg3(startaddress))