TinyLoad Malware Analysis

Andrew Glaz

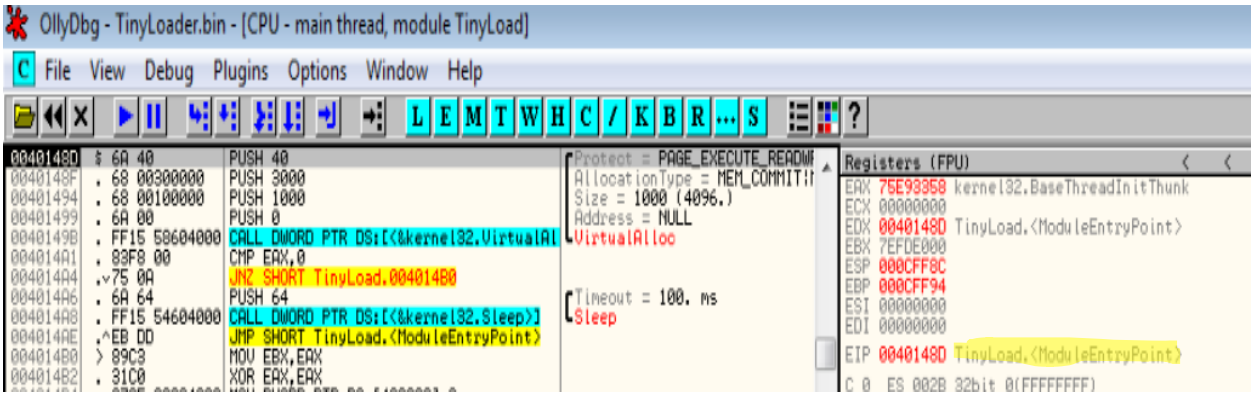
For the TinyLoader Malware, I will be using Call Stack Backtracing techniques to decrypt the malware.

This is a backdoor banking malware that delivers malware on point-of-sale and banking malware trojan. The TinyLoader malware runs on Windows for C2C. (Fidelissecurity.com) The malware uses a Kernel32.dll.

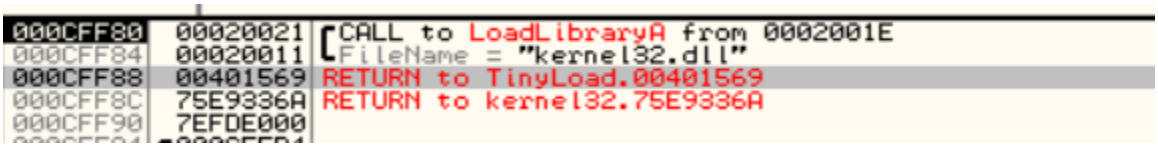
The malware spreads by “running a custom bytecode directly into the memory of the bot which makes it easier to load malware or augment malicious behavior”. (Fidelissecurity.com) The malware appears to maintain its persistence of using various commands from its C2C.

For this technique, I use OllyDBG for debugging and to find the entry point, which is the address 0040148D. I’m going to use the Call Stack Backtracing technique to identify the decrypted code.

I set breakpoints on multiple known API’s, such as VirtualAlloc, VirtualProtect, GetProcessAddress, LoadLibrary, GetModuleFileNameA and others.

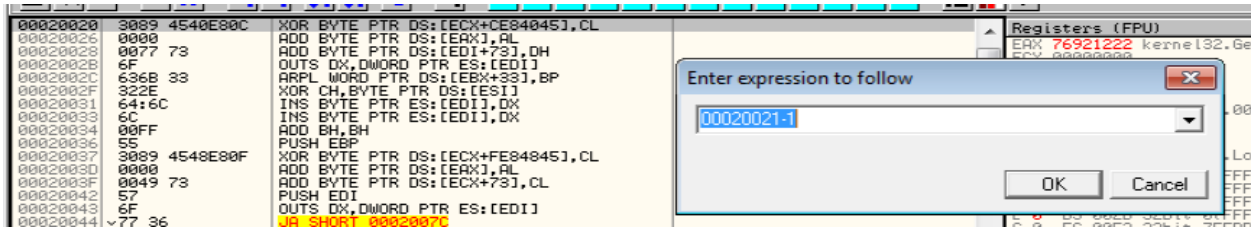


Now, I will run the malware until I get to the LoadLibraryA API.

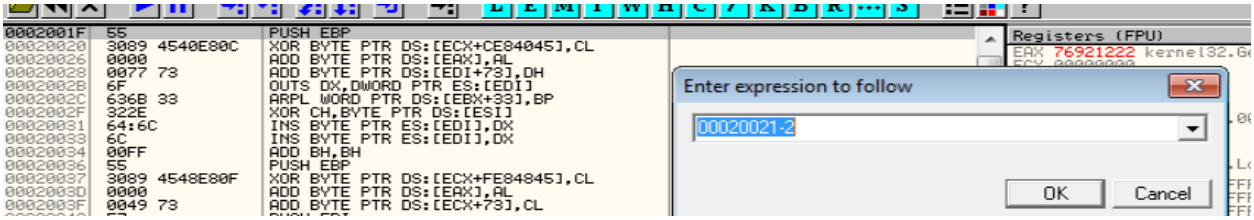


Now, I have a return address x00401569. Let’s check it out. This address points out after the entry function. I removed the analyses from the module

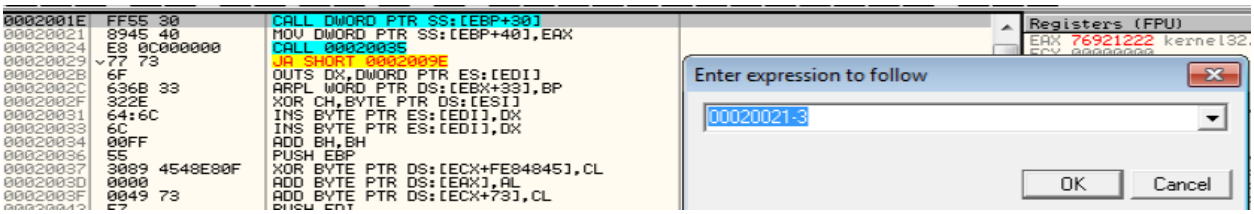
This is first the first address shift for 1 byte



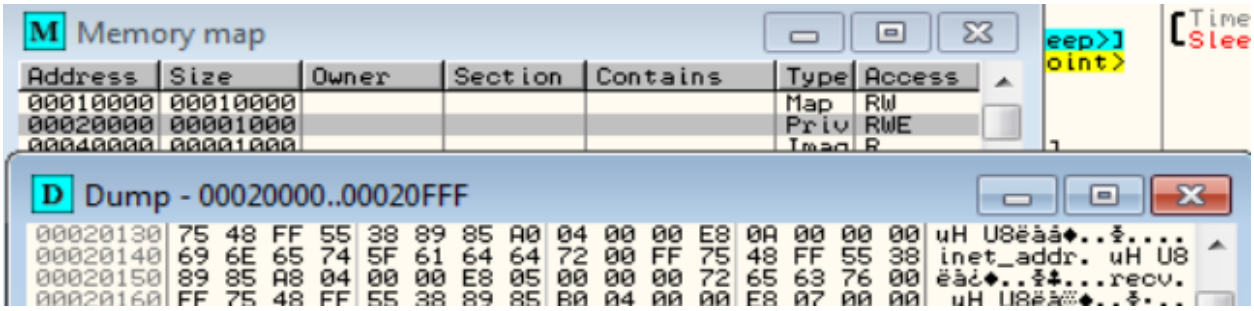
The second shift, 2 bytes



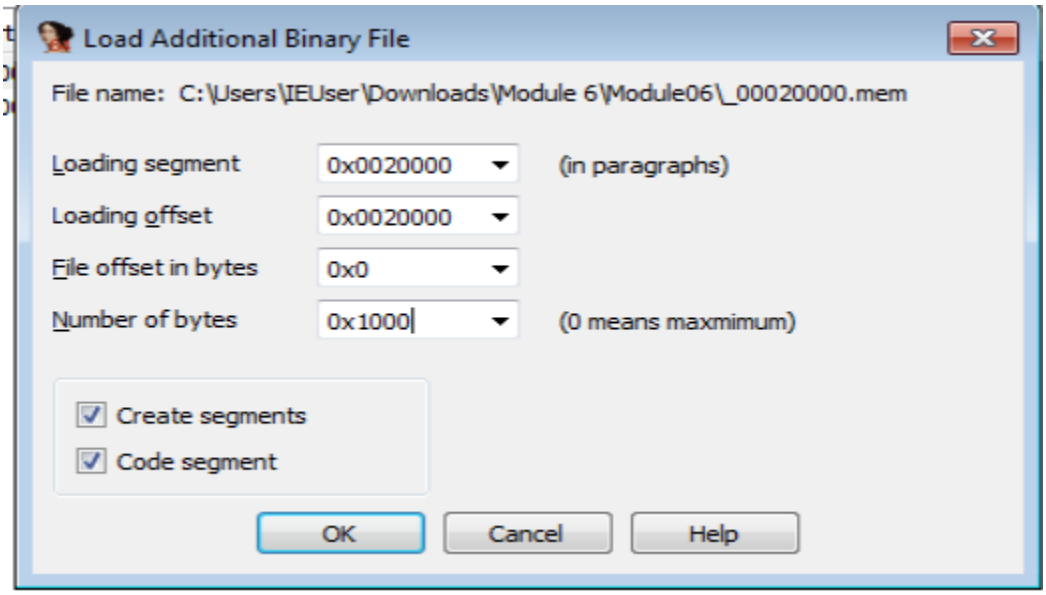
Third shift, 3 bytes



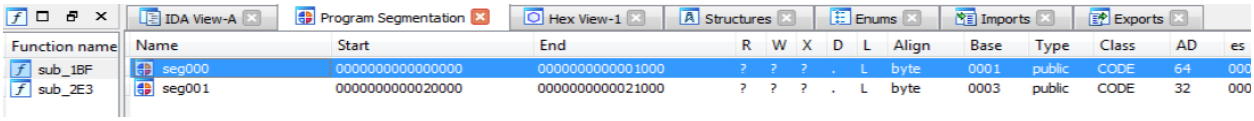
After this, I dump the memory by including the decrypted code from the memory map.



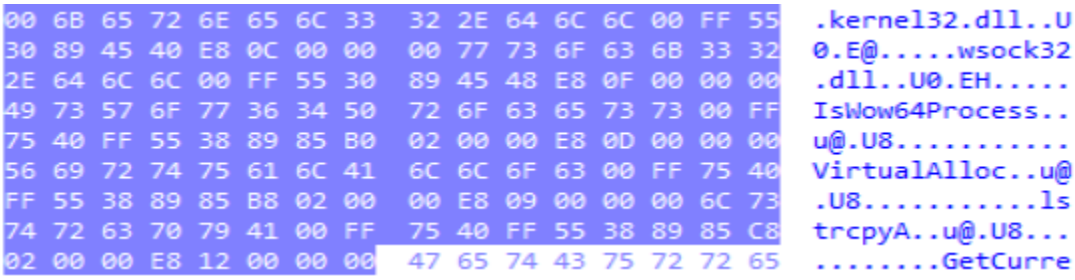
Now, I'm going to going to set the dumped memory file offset, _00020000.mem in the correct location in IDA Pro. I am using 1 byte here.



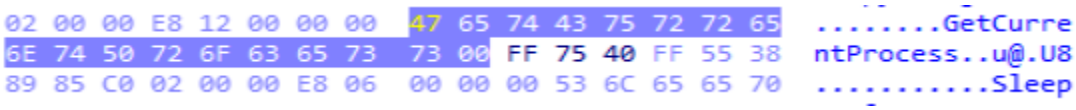
With this file loaded, I can now see the loaded file view in the segments.



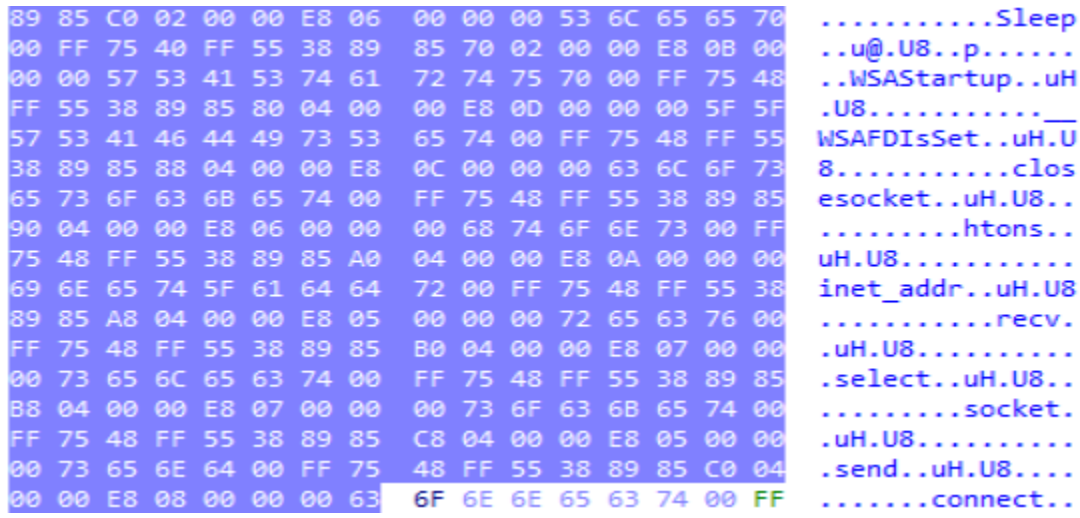
Now, I can go into the hex view, and I can see some separate strings, including the kernel32.dll



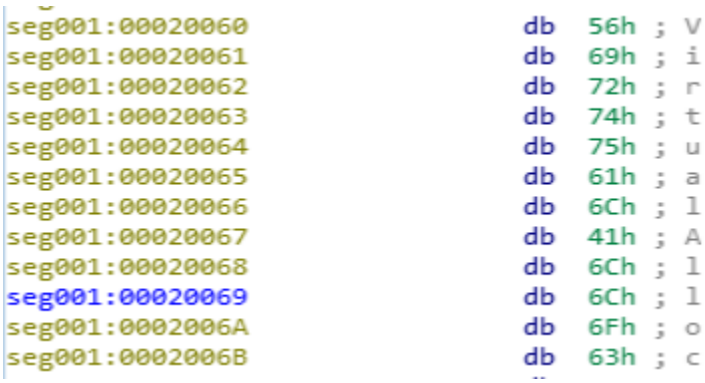
GetCurrentProcess is here, and then there is a sleep function called. Would like to know why this is?



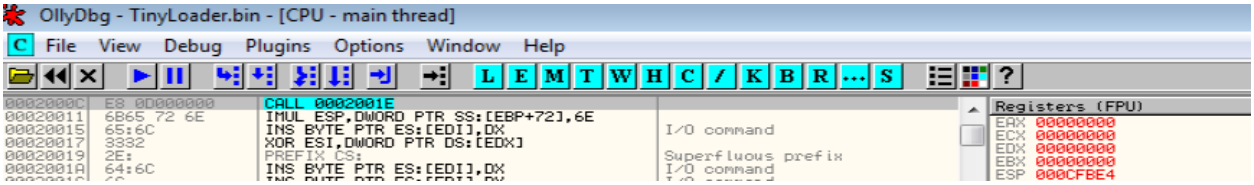
Then in the next data of the dump, this appears to be where the C2C is located, and possibly an IP address, with the inet_addr API.



You can also begin to see where the strings are stored in IDA Pro view. Still would like to know what the hotkey is to make the view better.



Finally, I go back to OllyDBG. I go and set a breakpoint on the outside of the entry point, 401569, and I rerun the application to get to 20021E and the call of LoadLibaryA. At this point, I got a little lost in the OllyDBG, but was able to find the entrypoint of the decryption code at 2000C



Thanks again for reading my analysis on the TinyLoaderMalware.

References:

https://fidelissecurity.com/threatgeek/threat-intelligence/deconstructing-tinyloader/

Https://Maltrak.com