Reflective Loading

Achieving stealth by creating processes without any trace

What is Reflective Loading?

Reflective loading is where we try to obfuscate a PE executable file by building it piece-by-piece dynamically on the fly using a special DLL called Reflective DLL.

So, the existence of the PE file is completely unknown by AV engines, since it is non-existent at the beginning and only brought into existence dynamically.

Stephen Fewer created a special library called the Reflective DLL Library. To turn a normal DLL into a Reflective DLL, all we need to do is to include StephenFewer's Reflective DLL library when compiling it.

Basic Concepts

- Creating processes directly from memory without using files
- Load a PE library directly from memory without using any files on disk
- Payload does not have to reside on disk and can be loaded and live only in memory
- As such it bypasses any AV engines that are scanning files
- The Reflective DLL does not register itself with the OS and also does not exist in the PEB of the target process.

Steps to create a reflective-loaded Trojan

1. You will need to put whatever you want to do in a DLL file
2. Then add Stephen Fewer’s library to it
3. Compile and build the DLL (it will be a Reflective DLL)
4. Then, embed the DLL as a shellcode into any Trojan (you may encrypt it first, if you want to add another layer of obfuscation)
5. Run the Trojan
6. The Trojan will allocate memory and run the Reflective DLL which will the call its ReflectiveLoader() function to dynamically construct a PE executable on the fly and execute it

Thank you