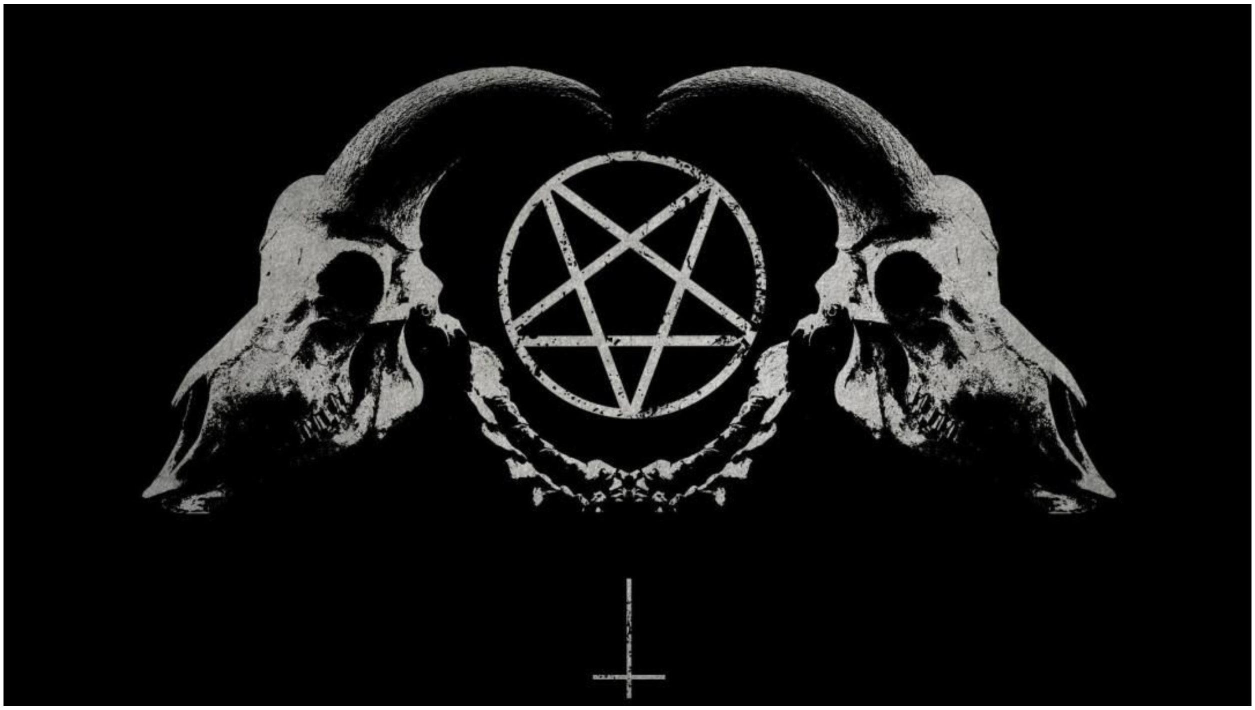
Hooking WinNT/2K/XP API

vx-underground.org archive // z0mbie



v0.01
(x) 2002
http://z0mbie.host.sk/

Our task is to hook some API functions in all existing processes, and in all new processes which may be created, under NT/2K/XP operating systems.

Patching existing processes can be done by means of the following:

1. Try to adjust privileges (some people believe it can help in some cases, but i'm still not sure) -- using OpenProcessToken, LookupPrivilegeValue("SeDebugPrivilege") and AdjustTokenPrivileges.
2. Get process list, using NtQuerySystemInformation
3. Try to open each process using OpenProcess()
4. Write own code into process' context, using VirtualProtectEx and WriteProcessMemory

Patching processes on creation is a bit more complex task. This is because the moment, at which process is loaded, is undefined. First, the main program file and NTDLL.DLL are loaded, and then control is returned into the parent process, which calls NtResumeThread. After that, other DLL's, such as KERNEL32.DLL are loaded.

So, if you want to hook API functions within NTDLL.DLL, there is no problem: just hook NtResumeThread in all existing processes (i've selected this function because it is always called after new process is loaded), and then (re)patch all existing processes, including new one, as it was described before.

But if you want to hook functions within other DLL's, you should first insert your code into NTDLL.DLL of the created process, then wait until other DLL's are loaded, and only after that install additional hooks.

This can be done by means of the following:

1. Hook NtResumeThread and LdrGetDllHandle within NTDLL.DLL, in all existing contexts.
2. On both functions called, (re)patch all existing processes and any loaded DLL's in these processes.

Here LdrGetDllHandle is just a function within NTDLL.DLL, which is called when other DLLs are loaded, which gives us an event to patch'em.

As you can see, the API hooking method which is described here, is just a virus, which lives only in the computer's memory.

Well, this is the simplest way I found in my first research of API hooking in NT-based systems.

Alternative is to patch all existing processes, then disable sfc, ename/copy some dll's, and patch them on disk; and i think this task is harder, however, it has its own specifics, good and bad sides.

Another way is to hook API functions within NTDLL.DLL only. But it is impossible for some functions, and very hard for other ones, because KERNEL32.DLL for example is not an empty space, and it performs different complex tasks. Yes, it is possible to hook NTDLL.NtQueryDirectoryFile instead of KERNEL32.FindNextFileW, but who knows what is simpler?