1. DllMain is found at 0x1000D02E in the .text section.

2. The import for gethostbyname is found at 0x100163CC in the .idata section.

3. The gethostbyname import is called nine times by five different functions throughout the malware.

4. A DNS request for pics.practicalmalwareanalysis.com will be made by the malware if the call to gethostbyname at 0x10001757 succeeds.

5. IDA Pro has recognized 23 local variables for the function at 0x10001656.

6. IDA Pro has recognized one parameter for the function at 0x10001656.

7. The string \cmd.exe /c is located at 0x10095B34.

8. That area of code appears to be creating a remote shell session for the attacker.

9. The OS version is stored in the global variable dword_1008E5C4.

10. TheregistryvalueslocatedatHKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WorkTime and WorkTimes are queried and sent over the remote shell connection.

11. The PSLIST export sends a process listing across the network or finds a particular process name in the listing and gets information about it.

12. GetSystemDefaultLangID, send, and sprintf are API calls made from sub_10004E79. This function could be renamed to something useful like GetSystemLanguage.

13. DllMain calls strncpy, strnicmp, CreateThread, and strlen directly. At a depth of 2, it calls a variety of API calls, including Sleep, WinExec, gethostbyname, and many other networking function calls.

14. The malware will sleep for 30 seconds.

15. The arguments are 6, 1, and 2.

16. These arguments correspond to three symbolic constants: IPPROTO_TCP, SOCK_STREAM, and AF_INET.

17. The in instruction is used for virtual machine detection at 0x100061DB, and the 0x564D5868h corresponds to the VMXh string. Using the cross- reference, we see the string Found Virtual Machine in the caller function.

18. Random data appears to exist at 0x1001D988.

19. If you run *Lab05-01.py*, the random data is unobfuscated to reveal a string.

20. By pressing the A key on the keyboard, we can turn this into the readable string: xdoor is this backdoor, string decoded for Practical Malware Analysis Lab :)1234.

21. The script works by XOR'ing 0x50 bytes of data with 0x55 and modifying the bytes in IDA Pro using PatchByte.