Analyze the malware found in the file *Lab05-01.dll* using only IDA Pro. The goal of this lab is to give you hands-on experience with IDA Pro.

Questions

1. What is the address of DllMain?
2. Use the Imports window to browse to gethostbyname. Where is the import   located?
3. How many functions call gethostbyname?
4. Focusing on the call to gethostbyname located at 0x10001757, can you fig- ure out which DNS request will be made?
5. How many local variables has IDA Pro recognized for the subroutine at 0x10001656?
6. How many parameters has IDA Pro recognized for the subroutine at 0x10001656?
7. Use the Strings window to locate the string \cmd.exe /c in the disassembly. Where is it located?
8. What is happening in the area of code that references \cmd.exe /c?
9. In the same area, at 0x100101C8, it looks like dword_1008E5C4 is a global variable that helps decide which path to take. How does the malware set dword_1008E5C4? (Hint: Use dword_1008E5C4's cross-references.)
10. A few hundred lines into the subroutine at 0x1000FF58, a series of comparisons use memcmp to compare strings. What happens if the string comparison to robotwork is successful (when memcmp returns 0)?
11. What does the export PSLIST do?
12. Use the graph mode to graph the cross-references from sub_10004E79. Which API functions could be called by entering this function? Based on the API functions alone, what could you rename this function?
13. How many Windows API functions does DllMain call directly? How many at a depth of 2?
14. At 0x10001358, there is a call to Sleep (an API function that takes one parameter containing the number of milliseconds to sleep). Looking backward through the code, how long will the program sleep if this code executes?
15. At 0x10001701 is a call to socket. What are the three parameters?
16. Using the MSDN page for socket and the named symbolic constants functionality in IDA Pro, can you make the parameters more meaningful? What are the parameters after you apply changes?
17. Search for usage of the in instruction (opcode 0xED). This instruction is used with a magic string VMXh to perform VMware detection. Is that in use in this malware? Using the cross-references to the function that executes the in instruction, is there further evidence of VMware detection?
18. Jump your cursor to 0x1001D988. What do you find?
19. If you have the IDA Python plug-in installed (included with the commercial version of IDA Pro), run *Lab05-01.py*, an IDA Pro Python script provided with the malware for this book. (Make sure the cursor is at 0x1001D988.) What happens after you run the script?

20. With the cursor in the same location, how do you turn this data into a single ASCII string?
21. Open the script with a text editor. How does it work?