Anaisy Garcia PID: 5734454

Lab 3-2Analyze the malware found in the file Lab03-02.dll using basic dynamic analysis tools.

Questions1. How can you get this malware to install itself?

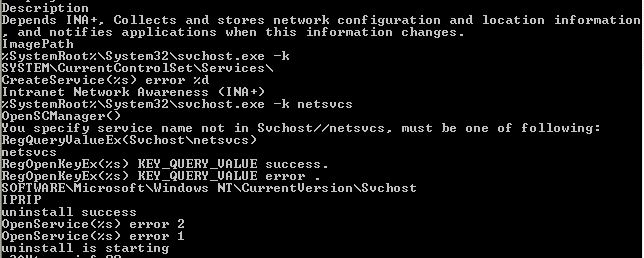


Using PEview we see various imports and exports. The exports tell us what this malware can do. Here we see it is a service that exports installs, uninstalls, and servicemain. Using cmd and the command line found in the book rundll32.exe Lab03-02.dll followed by one of the install exports. I tried first with install and got no results. With installA, I was able to make it install.

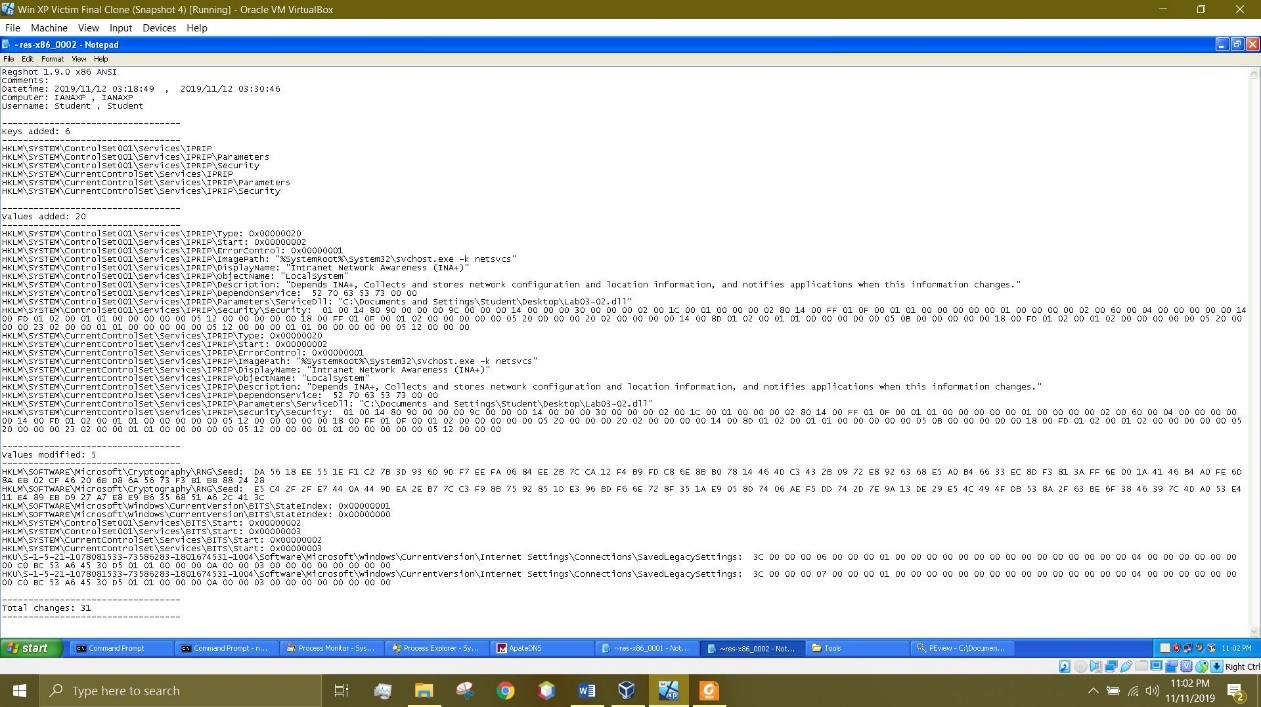


2. How would you get this malware to run after installation?

Using strings and analyzing the dll, we can see several paths and an executable called svchost.exe. Here we can see under one of the paths a keyword “IPRIP”.



After using the command line “rundll32.exe Lab03-02.dll, installA”, we use regshot to compare a shot before and after the malware was run. Here we can see a new key called “IPRIP” was added.



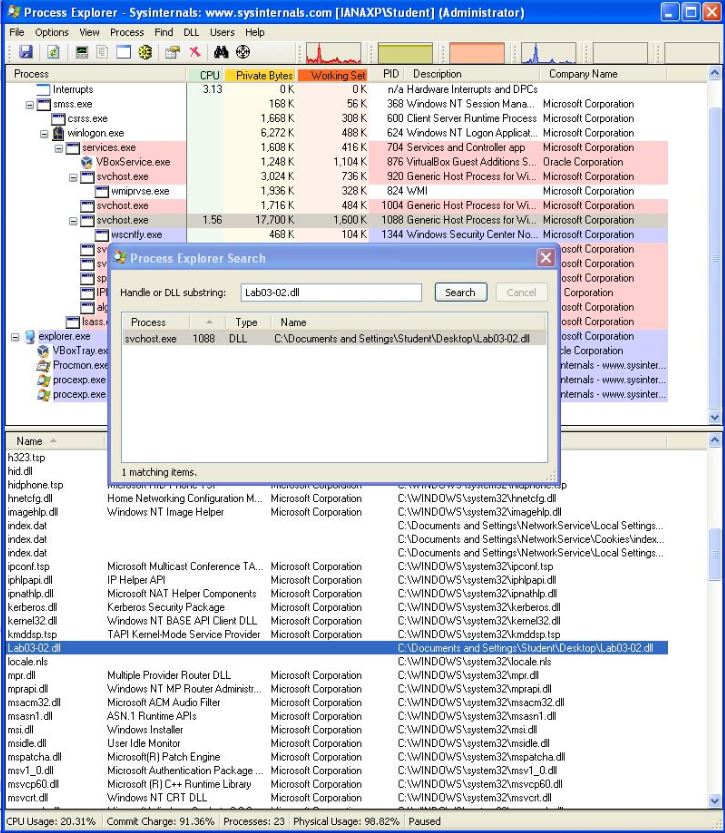
To run the malware, we use the command found in the book called net start.



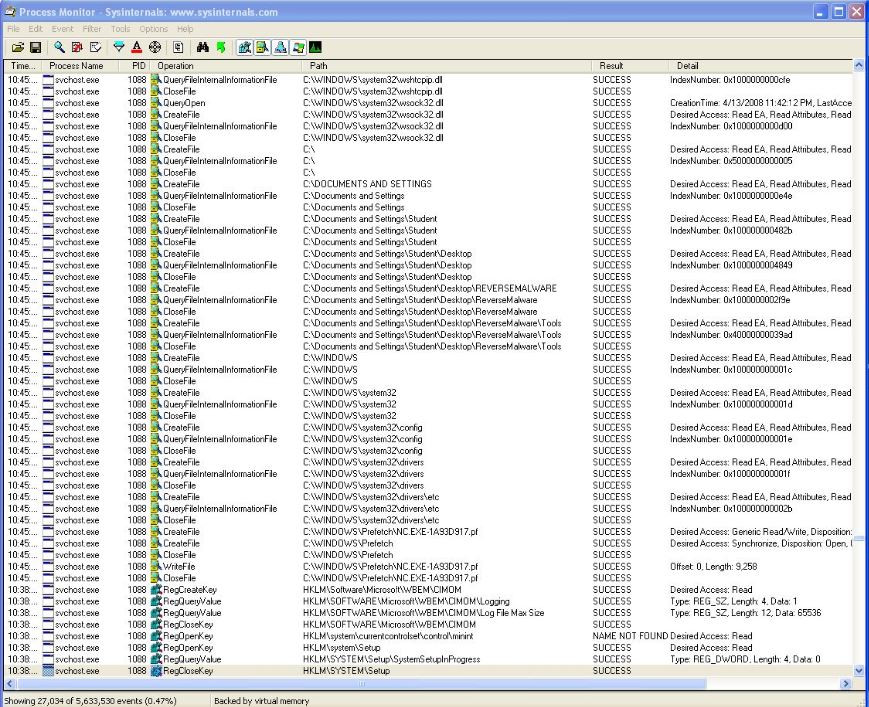
Here we see it started.

3. How can you find the process under which this malware is running?

To find the process it is under we can use process explorer to search the dll. When we search Lab03-02.dll, we see the ID for it being 1088. Which is using the process svchost.exe from the strings we did before.



When using process monitor and filtering by the ID we see that the process is creating and closing a file as well as creating a key. This is happening multiple times probably meaning the program refreshes.



4. Which filters could you set in order to use procmon to glean information?

The filter from the above picture shows I filtered by the PID 1088.

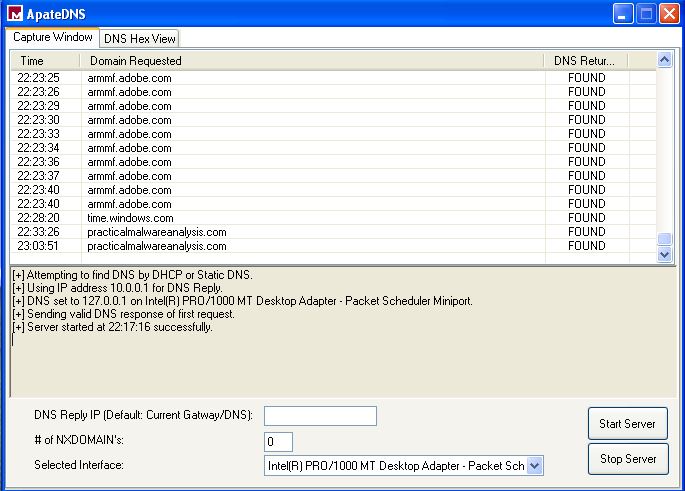
5. What are the malware’s host-based indicators?

Since the malware installs itself as a service on the registry called IPRIP which we see under question 2.

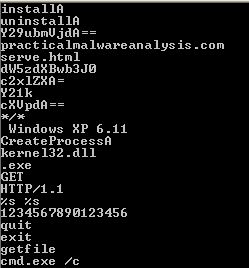


6. Are there any useful network-based signatures for this malware?

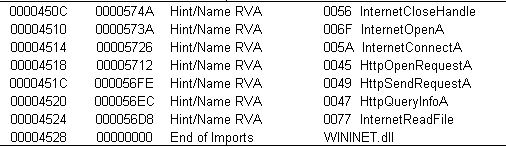
Using apateDNS, we saw that the malware does a DNS request for the website www.practicalmalwareanalysis.com.



We can also see in strings that there is a string that says GET HTTP/1.1.



We can also see that it has multiple imports from WININET.dll called InternetCloseHandle, InternetOpenA, InternetConnect, etc.



Lab 3-4Analyze the malware found in the file Lab03-04.exe using basic dynamic analysis tools. (This program is analyzed further in the Chapter 9 labs.)

Questions1. What happens when you run this file?

When this file is run by double clicking it briefly shows up on process explorer then disappears. Nothing shows up on netcat, apateDNS, or process monitor. 

2. What is causing the roadblock in dynamic analysis?

The roadblock is being caused by the program deleting itself. This probably means the program is missing something or the installation might not be correct.

3. Are there other ways to run this program?

We could try using a command like we did on lab 3-2 using one of the strings. When trying -cc, -re, -in, PST, or PDT nothing works and the program just continues to delete itself.



Lab 5-1Analyze 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. If you’ve already worked with IDA Pro, you may choose to ignore these questions and focus on reverse-engineering the malware.

Questions1. What is the address of DllMain?



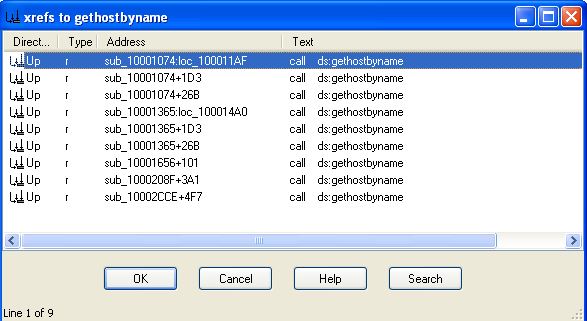
Address is 0x1000D02E.

2. Use the Imports window to browse to gethostbyname. Where is the import located?



Imported located at 0x100163CC.

3. How many functions call gethostbyname?



Called 9 times by 5 different functions.

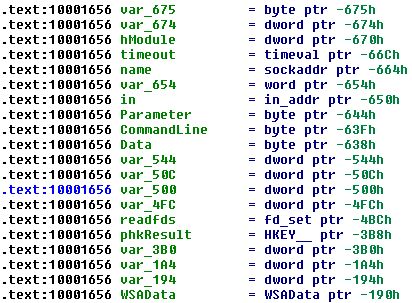
4. Focusing on the call to gethostbyname located at 0x10001757, can you figure out which DNS request will be made?



DNS request for Pics.practicalmaalwareanalysis.com

5. How many local variables has IDA Pro recognized for the subroutine at 0x10001656?

20 local variables found below in address 0x10001656.



6. How many parameters has IDA Pro recognized for the subroutine at 0x10001656?

Only one parameter found below in parenthesis.



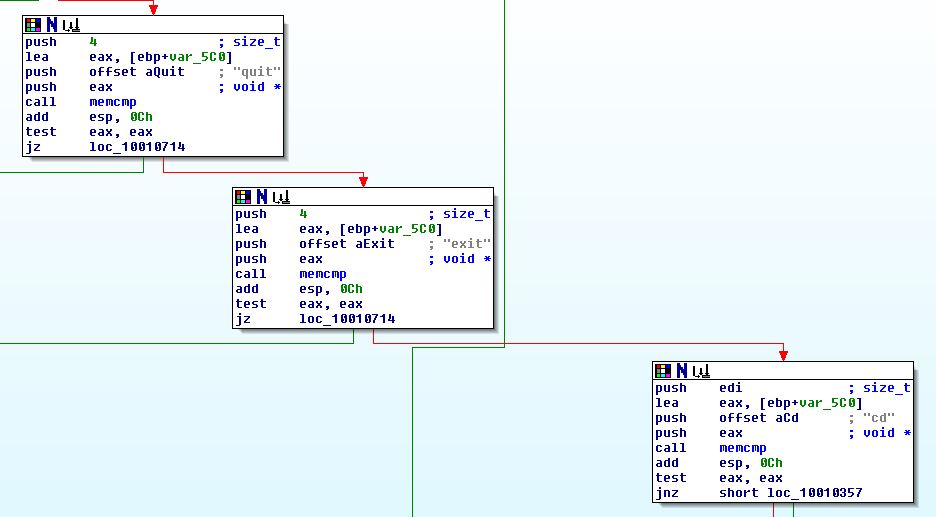
7. Use the Strings window to locate the string \cmd.exe /c in the disassembly. Where is it located?

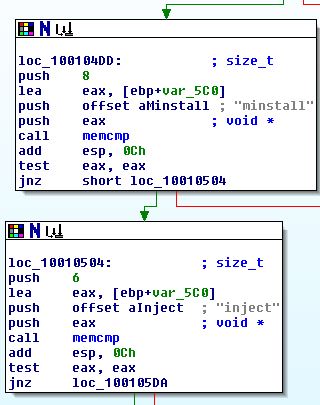
Located at 0x10096B34.



8. What is happening in the area of code that references \cmd.exe /c?

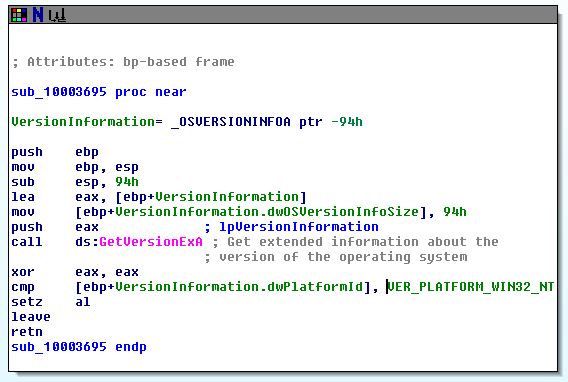
The code can execute shell commands which leads to believe its for remote shelling.





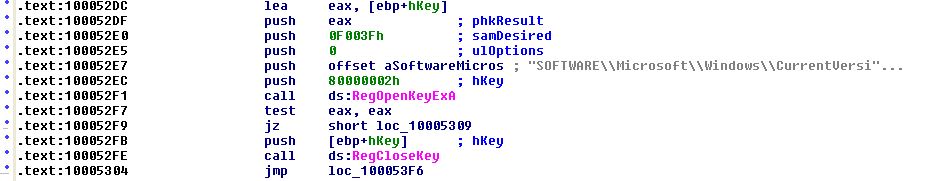
9. In the same area, at 0x100101C8, it looks like dword\_1008E5C4 is a global variable that helps decide which path to take. How does the malware set dword\_1008E5C4? (Hint: Use dword\_1008E5C4’s cross-references.)

This is were the version of the OS is stored.

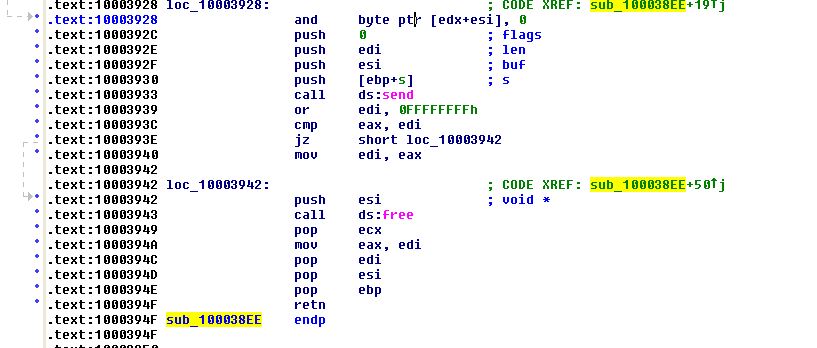


10. A few hundred lines into the subroutine at 0x1000FF58, a series of comparisons use memcmp to compare strings. What happens if the string comparison to robotwork is successful (when memcmp returns 0)?

When successful the registry values are put in query and sent over remote connection.

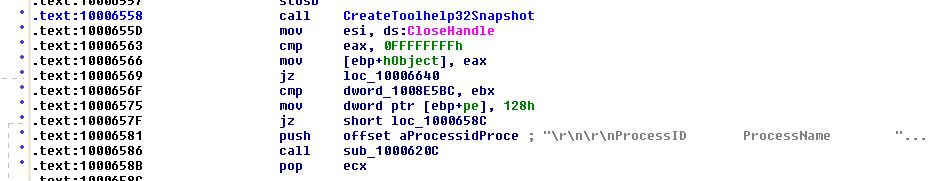






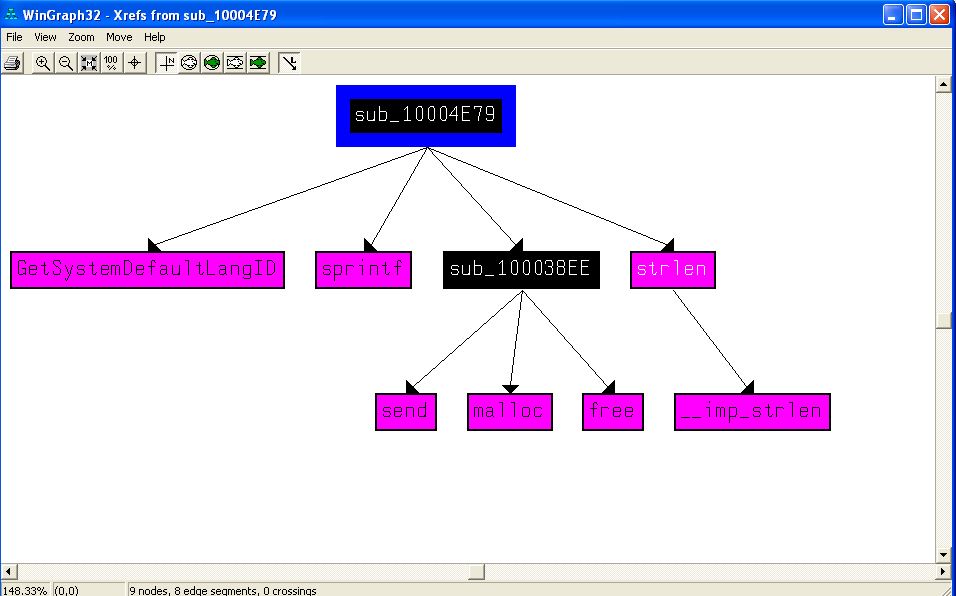
11. What does the export PSLIST do?

Finds or sends a process name and shows information on it such as processID.



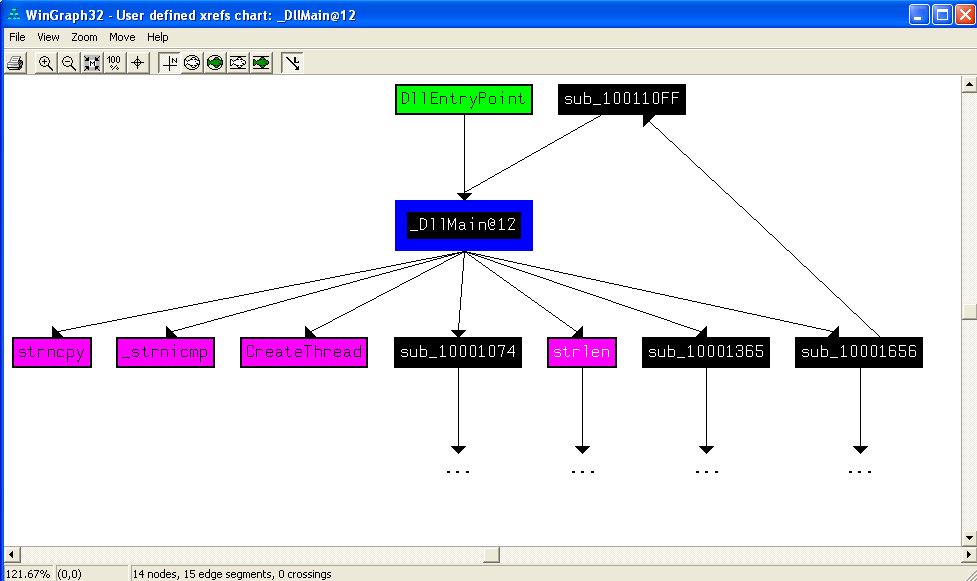
12. Use the graph mode to graph the cross-references from sub\_10004E79. Which API functions could be called by entering this function? Based on the API functions alone, what could you rename this function?

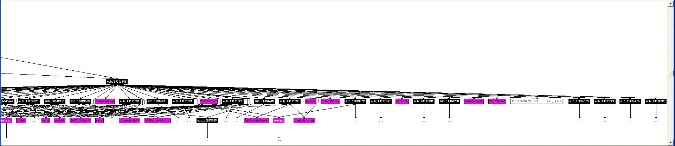
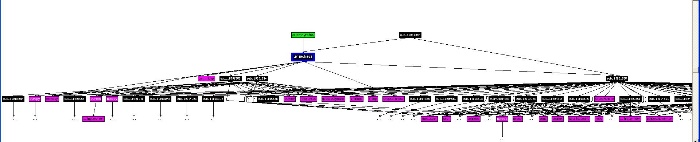
Since the first call is GetSystemDefaultLanguageID, we can rename it GetDefaultLanguage.



13. How many Windows API functions does DllMain call directly? How many at a depth of 2?

It calls 4 functions directly. At depth 2 it calls a lot more including Sleep.

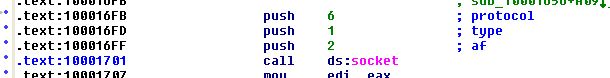




14. At 0x10001358, there is a call to Sleep (an API function that takes one parameter containing the number of milliseconds to sleep). Looking backward through the code, how long will the program sleep if this code executes?

30 seconds.

15. At 0x10001701 is a call to socket. What are the three parameters?



6, 1 , 2

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?

6: IPPROTO\_TCP

1: SOCK\_STREAM

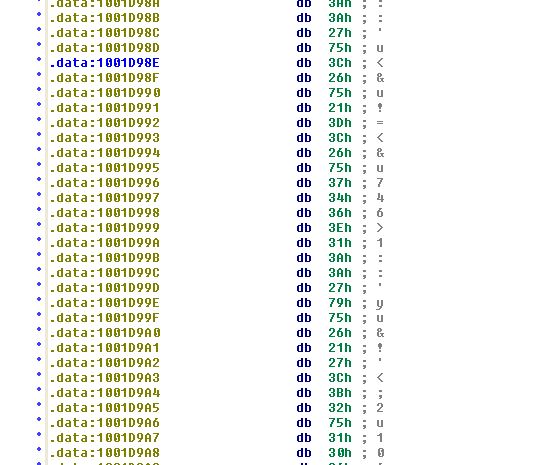
2: AF\_NET

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?

564D5868h = VMXh. When cross refrenced it is a Found Virtual Machine function. It can identify if it is running on a VM or not.

18. Jump your cursor to 0x1001D988. What do you find?

Random Data.



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?

Random data shows a string.

20. With the cursor in the same location, how do you turn this data into a single ASCII string?

To turn the data into a single string press A. This will show the string “xdoor is this backdoor, string decoded for practical malware analysis….”.

21. Open the script with a text editor. How does it work?

The script XORs the bytes of data 0x500 with 0x55. Then modifies using IDA pro and PatchByte.