CMPSC 443 Lab 8: Dynamic Analysis

# Lab Description

Note: you are supposed to use a Windows VM to finish all these labs

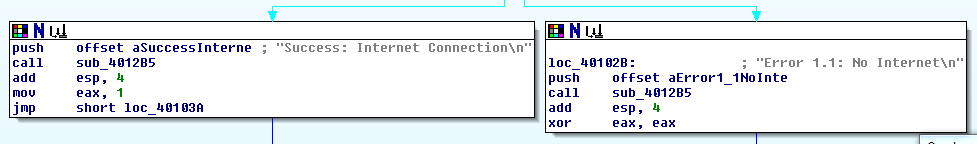
**Part I: basic dynamic analysis**

## Task 1

Analyze the malware found in the file *Lab06-04.exe*. You need to use ApateDNS to setup a DNS server that directs all DNS request to the local IP. Meanwhile, you may use python simplehttpserver (the command is “python –m SimpleHTTPServer 80”) to launch a web server that handles any HTTP request from the malware.

You need to explore the malware behavior based on the prior setup. Please provide a detailed paragraph of malware actions with screenshots for submission.

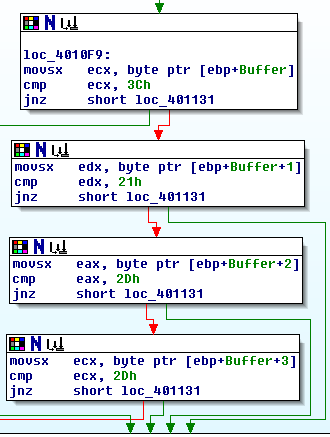
Examining *Lab06-04.exe* in IDA Pro tells us almost everything we need to know about this malware. For starters, it checks to see if it can connect to the internet:



Assuming it does, next it attempts to retrieve a file from “http://www.practicalmalwareanalysis.com/cc.htm”.

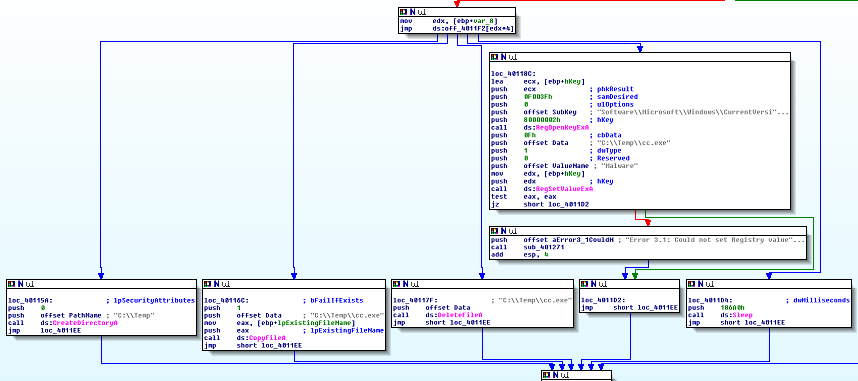
P:\CMPSC443\Git\CMPSC443\Lab08\Lab06-URL.PNG

Once again assuming that it retrieves the “.htm” file correctly, it then checks to see if the first bytes of data are “0x3C 21 2D 2D”, which correspond to “<!--", which is the beginning of an HTML document comment.



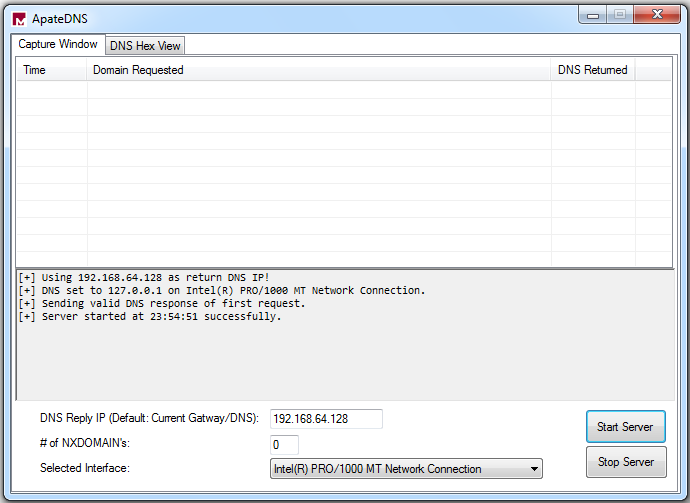
Assuming that the document starts with the above “<!--", the malware then takes the next byte of the HTML document, and subtracts 0x61 = 97 from it. 97 is coincidentally ASCII for ‘a’. As long as the resulting subtraction is less than or equal to 4 (letters that would be acceptable here are ‘a’ = 97-97 = 0, ‘b’ = 98-97 = 1, ‘c’ = 99-97 = 2, ‘d’ = 100-97 = 3, ‘e’ = 101-97 = 4), the program will jump to a memory address of that character’s respective subroutine.

Still only using IDA Pro, it seems as if ‘a’ creates a directory in “C:\Temp”, ‘b’ copies a file to “C:\Temp\cc.exe”, ‘c’ deletes the file at “C:\Temp\cc.exe”, ‘d’ sets a registry value at 'Software\Microsoft\Windows\CurrentVersion\Run', and ‘e’ sleeps for 100,000ms = 100 seconds.

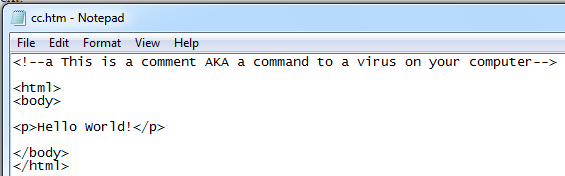


Once it’s done with the current command, it will then sleep for 0xEA60 = 60,000ms = 1 minute. It will run this loop of parsing a command and sleeping for 1 minute for 1440 times. All in total, upon execution of this program, it will run for 1440 minutes = 24 hours = 1 day.

Now, time to verify that the above hypothesis is correct. In order to do that, we will need to be able to “spoof” commands to the malware. We will do this through ApateDNS to spoof the url of “www.practicalmalwareanalysis.com” to our own computer, then by using Python to run a simple HTML server to ultimately host our own “cc.htm” file.

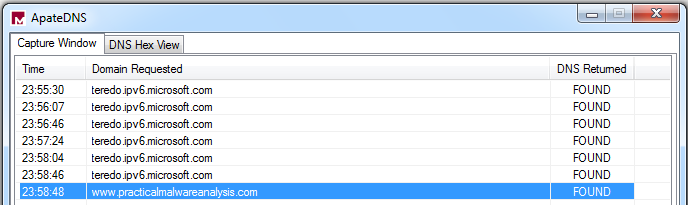


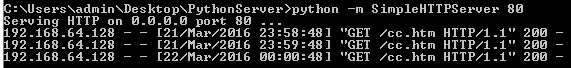
P:\CMPSC443\Git\CMPSC443\Lab08\PythonSimpleServer.PNG



Now that we have everything set up, it is time to run the malware! The results of running the malware are below:

P:\CMPSC443\Git\CMPSC443\Lab08\Lab06-04SuccessCommandA.PNG





Just as expected, “C:\Temp” now exists!



For the sake of brevity, the rest of the commands are much less in depth.

Running command ‘b’ results in “cc.exe” being copied to “C:\Temp”.



Running command ‘c’ deletes “cc.exe” in “C:\Temp”.



Lastly, command ‘d’ creates registry files and command ‘e’ sleeps for 100 seconds.

IDA Pro’s static analysis was proven correct!

**Part II: advanced dynamic analysis**

## Task 2:

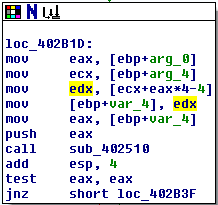
Analyze the malware found in the file *Lab09-01.exe* using OllyDbg and IDA Pro to answer the following questions.

0. Summary of the malware. (Not an actual question, just something I typed up.)

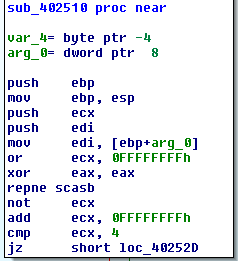
The first thing this program checks for is that there is not one argument being passed to it. If there is, it will delete itself by calling “sub\_402410”. Arg\_0 is defined earlier in the program and is equal to 8. [ebp+arg\_0] is the first argument being passed to main, which is “int argc”, which is the number of arguments being passed to this program.

**P:\CMPSC443\Git\CMPSC443\Lab08\Images\Lab09-Args.PNG**

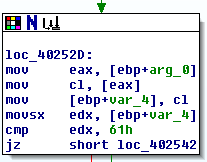
Then, the program gets the pointer to the char array of the arguments being passed to it (ebp+arg\_4), gets the last argument in that array (ecx+eax\*4-4), and passes it to function “sub\_402510”.



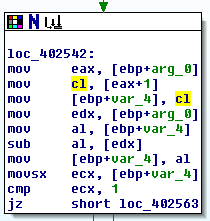
With that last argument, the program checks to make sure that it only has 4 characters. “Repne scasb” is an instruction to **sc**an **a** **s**tring in **b**inary. The number of bytes that it went through is left in a counting register, ecx, and compared with 4.



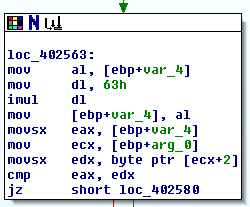
Assuming that there are four characters in the last argument, it then checks if the first byte is equal to 0x61 = ‘a’.



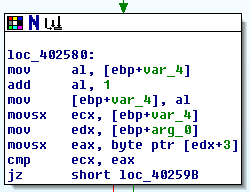
The program then compares the second byte of the argument with the first byte of the argument. If the difference is 1 (second byte = first byte + 1 = ‘a’ + 1 = ‘b’), the program continues.



Next, it compares the third byte of the argument with 0x63 = ‘c’.

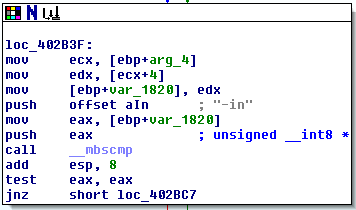


Lastly, it sees if the fourth byte of the argument is the same as the third byte + 1 = ‘c’ + 1 = ‘d’.



This is the end of sub\_402510. If any of the above turn out to be false, the program will delete itself by calling “sub\_402410”.

It now does a string comparison (\_\_mbscmp) with the second argument ( [\*\*argv + 4] ) passed to it (the first argument is the name of the virus itself, Lab09-01.exe) and “-in”. If they are the same, it will then look to see if an additional argument is passed and install to that location. Otherwise, it will install to a default location of “C:\Windows\System32” (as long as you are running with administrator privileges.



If the second argument wasn’t “-in”, it looks to see if the argument was “-re”. If so, it will do the same thing as above but remove the malware.

If the second argument is “-c”, you will need 7 arguments total or the malware will delete itself. If you specify 7 arguments, the four after “-c” will be set into the registry configuration data at “Computer\HKEY\_LOCAL\_MACHINE\SOFTWARE\MICROSOFT \XPS”.

If the second argument is “-cc”, it will display the current registry configuration data at “Computer\HKEY\_LOCAL\_MACHINE\SOFTWARE\MICROSOFT \XPS”. If the malware is not installed, it will not show anything. If it is installed, it will show the configuration set previously with “-c”. If no configuration was previously set, it will show “k: ups h:http://www.practicalmalwareanaysis.com p:80 per:60”.

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

Run the malware at the command line with administrative privileges: “Lab09-01.exe -in abcd”.

2. What are the command-line options for this program? What is the password requirement?

Command line options are:

* -in
  + Install
* -re
  + Remove
* -c
  + Set registry configuration
* -cc
  + Display configuration

3. What are the host-based indicators of this malware?

If the malware is installed, it will be located at “C:\Windows\System32\Lab09-01.exe”. The registry may be modified at “Computer\HKEY\_LOCAL\_MACHINE\SOFTWARE\MICROSOFT \XPS”, notice the space after MICROSOFT.

4. What are the different actions this malware can be instructed to take via the network?

The network can send it the standard commands of “-in”, “-re”, “-c”, and “-cc”.

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

By default, the registry stores a value of “http://www.practicalmalwareanalysis.com” over port 80. However, this is modifiable to any URL with the “-c” command.

# Submission

|  |
| --- |
| * Please write a report to answer all questions. * Screenshots needed for task 1. |