CMPSC 443 Lab 9: OllyDbg

# Lab Description

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

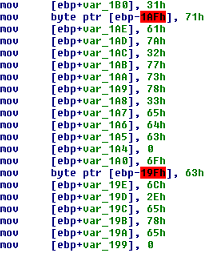
## Task 1

Analyze the malware found in the file Lab09-02.exe using OllyDbg to answer the following questions.

Questions

1. What strings do you see statically in the binary?

At the very beginning of the program, it puts 0x3171617a327773783365646300 = “1qaz2wsx3edc” and 0x6f636c2e65786500 = “ocl.exe” onto the stack.



In sub\_401000, it pushes the string “cmd” onto the stack.

P:\CMPSC443\Git\CMPSC443\Lab09\cmd.PNG

2. What happens when you run this binary?

Nothing noticeable.

3. How can you get this sample to run its malicious payload?

At one point, it obtains the current location of the file:

P:\CMPSC443\Git\CMPSC443\Lab09\Filename.PNG

It then later has both the filename and “ocl.exe” pointers contained in registers. P:\CMPSC443\Git\CMPSC443\Lab09\FilenamesOnReg.PNG

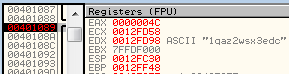
After making a function call to “\_strcmp” with both names on the registers, the program shortly thereafter exits. This leads me to believe that in order to continue the program execution, the filename needs to be changed.

Changing the filename to “ocl.exe” results in the executable running much more code after the string comparison.

4. What is happening at 0x00401133?

Just as I mentioned in Task 1.1, the program is putting strings onto the stack. It first adds “1qaz2wsx3edc” followed by “ocl.exe”.

5. What arguments are being passed to subroutine 0x00401089?



Once in subroutine 0x00401089 the only argument being passed to it is the same string that is referenced above in Task 1.4 and Task 1.1: “1qaz2wsx3edc”.

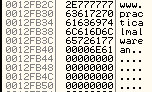
6. What domain name does this malware use?

Looking at IDA Pro, “WSASocketA” is called at address 0x00401290. After that it will (if successfully opened a socket) call “gethostbyname” which is at “0x004012CC”. When calling this function, it passes “www.practicalmalwareanalysis.com” as an argument.

P:\CMPSC443\Git\CMPSC443\Lab09\0x4012CC.PNG

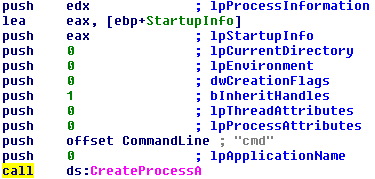
7. What encoding routine is being used to obfuscate the domain name?

In “sub\_401089”, the program will get stuck in a loop at 0x4010E3. This loop continually uses the “XOR” command which slowly begins to decode the obfuscation. This is process is paused and shown in the image below:



8. What is the significance of the CreateProcessA call at 0x0040106E?

This creates a process running the command line with the current environment variables.



## Task 2:

Analyze the malware found in the file Lab09-03.exe using OllyDbg and IDA Pro. This malware loads three included DLLs (DLL1.dll, DLL2.dll, and DLL3.dll) that are all built to request the same memory load location. Therefore, when viewing these DLLs in OllyDbg versus IDA Pro, code may appear at different memory locations. The purpose of this lab is to make you comfortable with finding the correct location of code within IDA Pro when you are looking at code in OllyDbg.

Questions

1. What DLLs are imported by Lab09-03.exe?

According to IDA Pro, Lab09-03.exe has many imports. They are from libraries DLL1, DLL2, KERNEL32, and NETAPI32.

2. What is the base address requested by DLL1.dll, DLL2.dll, and DLL3.dll?

DLL1.dll requests an Imagebase of 0x1000000.

P:\CMPSC443\Git\CMPSC443\Lab09\BaseAddrDLL1.PNG

DLL2.dll also requests an Imagebase of 0x10000000.

P:\CMPSC443\Git\CMPSC443\Lab09\BaseAddrDLL2.PNG

DLL3.dll does the same, requesting an Imagebase of 0x10000000.

P:\CMPSC443\Git\CMPSC443\Lab09\BaseAddrDLL3.PNG

3. When you use OllyDbg to debug Lab09-03.exe, what is the assigned based address for: DLL1.dll, DLL2.dll, and DLL3.dll?

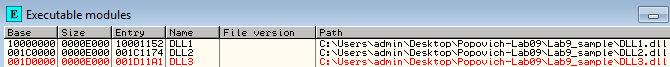
Because they all were requesting the same base address, it is impossible that they all get the same one if we are using more than one at a time.

Viewing “executable modules” in OllyDbg, we see that DLL1 and DLL2 are imported as soon as the malware executes, but DLL3 isn’t imported until runtime.

DLL1 = 0x10000000.

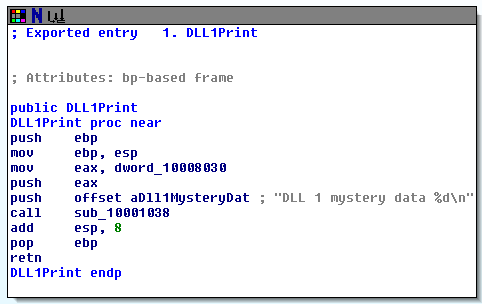
DLL2 = 0x001C0000.

DLL3 = 0x001D0000.



4. When Lab09-03.exe calls an import function from DLL1.dll, what does this import function do?

Looking at Lab09-03.exe in IDA Pro, the only function imported from DLL1 is “DLL1Print”. Opening up DLL1.dll in IDA Pro and looking at its exported functions, we can find the “DLL1Print” function.



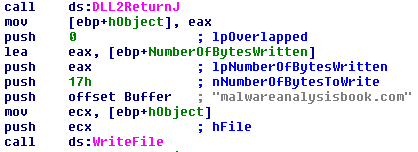
This function prints “DLL 1 mystery data [dword\_10008030]\n”.

We see in DLLMain() that “GetCurrentProcessId” is called and then a value is taken and stored at “dword\_10008030”, leading me to believe that DLL1 prints out “DLL 1 mystery data [DLL1 process ID]\n”.

P:\CMPSC443\Git\CMPSC443\Lab09\0x10008030.PNG

5. When Lab09-03.exe calls WriteFile, what is the filename it writes to?

The process WriteFile is sent [ebp+hObject] as an argument. [ebp+hObject] is stored by the subroutine DLL2ReturnJ.



DLL2ReturnJ is imported via DLL2.dll. Looking at DLL2ReeturnJ in DLL2.dll, we see that it returns dword\_1000B078. Looking at DLL2’s main(), we see that CreateFileA is called and then the return variable is stored to dword\_1000B078. 

Looks like it writes to a file named “temp.txt”.

6. When Lab09-03.exe creates a job using NetScheduleJobAdd, where does it get the data for the second parameter?

The second parameter to NetScheduleJobAdd is “LPBYTE Buffer”.

P:\CMPSC443\Git\CMPSC443\Lab09\NetScheduleJobAdd.PNG

Lab09-03 calls NetScheduleJobAdd at 0x00401085. The buffer is taken from [ebp+Buffer].

7. While running or debugging the program, you will see that it prints out three pieces of mystery data. What are the following: DLL 1 mystery data 1, DLL 2 mystery data 2, and DLL 3 mystery data 3?

As said in Task2.4, DLL 1 mystery data is the Process ID.

DLL 2 mystery data is the HANDLE WINAPI return value from CreateFileA().

DLL 3 mystery data is address to “ping www.malwareanalysisbook.com”.

8. How can you load DLL2.dll into IDA Pro so that it matches the load address used by OllyDbg?

When loading a new file in IDA Pro you can check a checkbox to manually load the base address. You can do this to match the address in OllyDbg.

# Submission

|  |
| --- |
| * Please write a report to answer all questions. |