# LAB 5

## Malware Analysis

Under the direction of
Dr. Samuel Liles

**Table of Contents**

Abstract

This lab is focused on Malware Analysis. The lab is going to use tools and application to do Static/Dynamic analysis of the malware while being isolated from the internet. The Practical Lab 7.1 to Lab 7.3 will be carried out to answer the questions provided.

The Computer Anti-virus was disabled as part of the instructions to enable the download and extract of the files being used. This lab is intended to lay grounds for further labs in the course.

*Keywords:* Digital Investigation, Forensic Evidence, Malware Analysis.

# Lab 5 Malware Analysis

## Steps of the process

### Preparing the LAB

The Computer was rebooted, anti-virus was disabled, and the appropriate files were downloaded. Different Images of VM were installed. Installation of different windows environment such as XP, 7 and 8.1. Programs needed have been downloaded and snapshots of the process have been taken.

### LAB 7-1, 7-3

### Applications & Tools

The following applications are used to forensically examine the files. The following descriptions have been captured from the developer's website and manuals.

**PEiD**," is an intuitive application that relies on its user-friendly interface to detect packers, cryptors and compilers found in PE executable files – its detection rate is higher than that of other similar tools since the app packs more than 600 different signatures in PE files" (Gröbert, 2010).

**Resource Hacker**,"is a freeware utility to view, modify, rename, add, delete and extract resources in 32bit & 64bit Windows executables and resource files (*.res). It incorporates an internal resource script compiler and decompiler and works on all (Win95 - Win7) Windows operating systems" (Johnson, 2011).

**PE Explorer**"provides powerful tools for disassembly and inspection of unknown binaries, editing the properties of 32-bit executable files and customizing and translating their resources. Use this product to do reverse engineering, analyze the procedures and libraries an executable uses." (Heaventools Software, 2009).

**Process Monitor** is an advanced monitoring tool for Windows that shows real-time file system, Registry and process/thread activity. It combines the features of two legacy Sysinternals utilities, Filemon and Regmon, and adds an extensive list of enhancements including rich and non-destructive filtering, comprehensive event properties such session IDs and user names, reliable process information, full thread stacks with integrated symbol support for each operation, simultaneous logging to a file, and much more. Its uniquely powerful features will make Process Monitor a core utility in your system troubleshooting and malware hunting toolkit (Russinovich & Cogswell, 2014).

**ApateDNS**, is a tool for controlling DNS responses though an easy to use GUI. As a phony DNS server, ApateDNS spoofs DNS responses to a user-specified IP address by listening on UDP port 53 on the local machine. It responds to DNS requests with the response set to any IP address you specify. The tool logs and timestamps any DNS request it receives. You may specify a number of non-existent domain (NXDOMAIN) responses to send before returning a valid response. ApateDNS also automatically sets the local DNS to localhost. By default, it will use either the set DNS or default gateway settings as an IP address to use for DNS responses. Upon exiting the tool, it sets back the original local DNS settings (Davis, 2011).

**Regshot**, is a small, free and open-source registry compare utility that allows you to quickly take a snapshot of your registry and then compare it with a second one - done after doing system changes or installing a new software product. The changes report can be produced in text or HTML format and contains a list of all modifications that have taken place between the two snapshots. In addition, you can also specify folders (with subfolders) to be scanned for changes as well (Regshot Team, 2013).

**IDA** is the Interactive DisAssembler: the world's smartest and most feature-full disassembler, which many software security specialists are familiar with (Hex-Rays SA, 2014).

## Issues or problems

Nothing So far.

## Conclusions

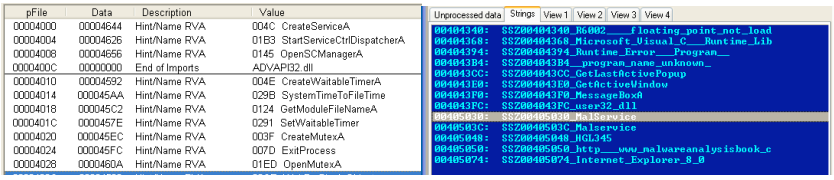
The Lab identified several programs that helps explore the malwares. The tools showed if the files being used are infected or packed. The tools used also showed the resources on the system that is being utilized such as privilege, CPU usage, Network communication.

## Case studies

No Case studies was given with this lab.

## Review questions

### Lab 7-1

| Answers | Lab07-01. exe |
|---|---|
| 1 | Static Analysis shows the following<br>Using PE Explorer several strings of interest was found such<br>http://www.malwareanalysisbook.com & MalService<br>Furthermore using PEview some functions would raise an alarm such as CreateServiceA & StartServiceCtrlDispatcharA<br>  <br>From this we can say that the malware creates services and uses internet connection.<br>Dynamic Analysis<br>We see that the file creates several threads and registry keys<br>**Keys added:4**<br>HKLM\SYSTEM\ControlSet001\Services\Malservice<br>HKLM\SYSTEM\ControlSet001\Services\Malservice\Security<br>HKLM\SYSTEM\CurrentControlSet\Services\Malservice<br>HKLM\SYSTEM\CurrentControlSet\Services\Malservice\Security<br>**Values added:16**<br>HKLM\SYSTEM\ControlSet001\Services\Malservice\Security\Security: 0<br>HKLM\SYSTEM\ControlSet001\Services\Malservice\Type: 0x00000010<br>HKLM\SYSTEM\ControlSet001\Services\Malservice\Start: 0x00000002<br>As we see above the malservice is being created with the following options Type is set to independent process, Start is set to auto start as shown in MSDN Microsoft<br>Which mean that this Malservice runs in its own process every time the operating system is restarted. |
| 2 | Using Google to look up mutex we found that it explains that it's an object that has multiple threads that share the same resources consecutively while maintaining a unique presence for the Main Object. Which probably means that the author of the program intends to have a unique single instance running and not multiple. |
| 3 | From the information above we can find the Malservice created as well as the registry keys set for it. |
| 4 | From the DNS server we see that the file has been trying to communicate with the website shown in the graphs and reflected in the Process Monitor software.<br>5:08:07.9739709 PM Lab07_01.exe 2956 TCP Reconnect WSUCCESS<br>5:08:07.9740871 PM Lab07_01.exe 2956 TCP Reconnect WSUCCESS 17:08:07 www.malwareanalysisbook.com |
| 5 | The program do sleep using the wait function WaitForSingleObject however I could not find out how long will it be sleeping |
| 6 | Going over the code their is no ending for the program the only exit is done when the |

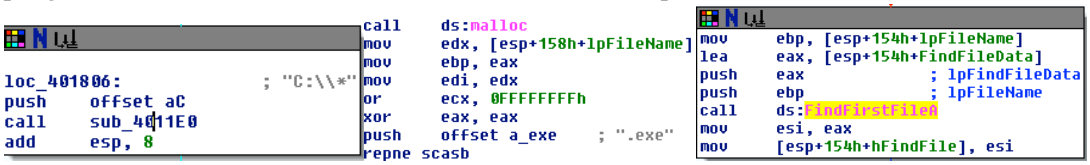
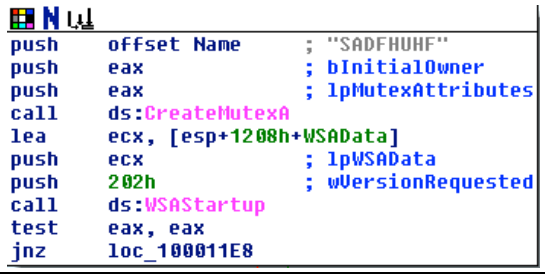
| | |
|---|---|
| | program fails to find an internet connection other than that it will wait the amount of time needed and then run continually trying to connect to the malware website given above. |

**Lab 7-2**

| Answers | Lab07-02. exe |
|---|---|
| 1 | Using static analysis we find that the program uses 3 Dll files with nothing alarming except the OLEAUT32.DLL that has the following function OleInitialize CoCreateInstance, OleUninitialize which for some reason does not feel right. We also find the following string in the program http://www.malwareanalysisbook.com/ad.html Ad could hint that an advertisement of some sort might be used. However, it also could be something else.<br>Next we move to Dynamic analysis.<br>After running the program we can find that the program uses IE to go to the website provided above. No other activity is evident.<br>Using IDA Pro, we can see that the flow is one way and no loops are used therefore the program seems non persistence and it run once only. |
| 2 | From what was said before it seems like the program opens the website and displays it only. 18:25:49 www.malwareanalysisbook.com |
| 3 | As soon as the program is finished displaying the ad the program exits |

**Lab 7-3**

| Answers | Lab07-03.exe; Lab07-03.dll |
|---|---|
| 1 | We start by Static Analysis of both files to find out the following functions<br>That creates Process and Create Mutex we also see file creating and handling.<br>Value<br>0296 Sleep<br>0044 CreateProcessA<br>003F CreateMutexA<br>01ED OpenMutexA<br>001B CloseHandle<br>001B CloseHandle<br>02B0 UnmapViewOfFile<br>01B5 IsBadReadPtr<br>01D6 MapViewOfFile<br>0035 CreateFileMappingA<br>0034 CreateFileA<br>0090 FindClose<br>009D FindNextFileA<br>0094 FindFirstFileA<br>0028 CopyFileA<br>We can also find using strings the following information: 172.26.152.13, hello, C:\Windows\System32\Kernel32.dll (note 132 not L32) , SADFHUHF, WARNING_THIS_WILL_DESTROY_YOUR_MACHINE<br>Next we move to dynamic analysis.<br>Nothing tangible was added to the registry. And not so many activities happen.<br>Next step would be to use IDA Pro to learn more about the program and library. |

| | |
|---|---|
| | Reading over we find out that to run the program we need two arguments one of which is the string WARNING_THIS_WILL_DESTROY_YOUR_MACHINE. If the argument is not correct or there are no two arguments the program does nothing and exits.<br>Using IDA Pro we can see lots of Find functions that search the C directory for files and those files are being modified by adding the kerne132.dll file which indicates that the Malware is trying to persistence by adding itself to all executables so next time the program runs the Dll is used and loaded into the computer .<br>  |
| 2 | The file created Kernel32.dll is one clear signature as well as the mutex created named SADFHUHF. Also the network connection logs created when communicating using 172.26.152.13 could be used as a network indicator.<br>  |
| 3 | From the Static analysis we found out that there is a hello message as well as a sleep function in the Address table of PE view, with the IP address it would say that the malware creates a session for a remote host to access the pc and execute commands or sleep the malware which is verified by the flow in IDA pro showing that it will either continue with the program or simply sleep based on the input provided from the remote session.<br>  |
| 4 | Since the Program modifies exe files and it will be very hard to track them and fix them one by one. A simple restore from backup or reinstall would be the longer and safer process. However, if that is not possible simply modifying the fake kerna132 Dll to reflect the original kernal32.dll would also be advisable. If modifying system files is not an option I would block all outgoing connections to the ip address provided. Such |

| | |
|---|---|
| | malware are very hard to handle and therefore it would be safer to simply format and reinstall the system from scratch. |

References

Davis, S. (2011, October). *ApateDNS*. Retrieved from https://www.mandiant.com/blog/research-tool-release-apatedns/

Gröbert, F. (2010, 02 07). *PEiD*. Retrieved 02 18, 2014, from https://code.google.com/p/kerckhoffs/downloads/

Heaventools Software. (2009, 10 14). *Heaventools*. Retrieved from http://heaventools.com/download.htm

Hex-Rays SA. (2014, July). *Freeware Download Page*. Retrieved from https://www.hex-rays.com/index.shtml

Johnson, A. (2011, 09 16). *Resource Hacker*. Retrieved from http://www.angusj.com/resourcehacker/

Regshot Team. (2013, August). *Regshot*. Retrieved from http://sourceforge.net/projects/regshot/

Russinovich, M., & Cogswell, B. (2014, March). *Process Monitor v3.1*. Retrieved from http://technet.microsoft.com/en-us/sysinternals/bb896645.aspx