**Overview**

The purpose of this assignment is to practice the skills taught in Chapter 1 of the Sikorski textbook. In order to simulate realistic malware analysis, you will be given little or no information about the program you are analyzing. The malware and most of the questions in this exercise are taken from Michael Sikorski. “Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software.”

This document outlines the major steps you will have to take, and you will need to provide answers to the questions asked. There are no restrictions on resources that can be used to complete the assignment. This includes consulting with other students and the instructor. We only ask that you will not share your report with others.

Please use this document as the template of your report, and remember to submit it in .pdf form by the deadline. This assignment will be graded for completion.

**Part 1 - Download and install the tools**

1. Download the lab files from Canvas. These files are taken from the “Practical Malware Analysis” book mentioned above. Please remember to make it a habit not to run any binary in your local environment even when practicing static analysis. Instead, move the files to your safe VM environment. The password is “infected”. [7-zip](https://www.7-zip.org/download.html) (32-bit) is a safe program for extracting files on Windows if you are in need of one.

2. Download a PE viewer tool. You can choose whichever one you want and trust. The book uses [PEview](http://wjradburn.com/software/), which we find safe although VirusTotal may report it as suspicious (false positive?). Make sure you can run your selected PE Viewer on your Windows VM. The instructions in this lab will follow the book which uses PEview. Based on the slides and/or other resources, what does this tool do?

PEView, as the name suggests, provides a quick and easy way to view the structure and contents of Portable Executable (PE) and Component Object File Format (COFF) files. PE (Portable Executable) file refers to the executable file format used in Windows OS and is based on COFF (Common Object File Format) of UNIX. Among the extensions we often encounter in Windows, PE files are EXE, SCR, DLL, OCX, SYS, and OBJ. In summary, it is a program that looks into the structure of a file running on Windows.

3. Download [PEiD](http://www.softpedia.com/get/Programming/Packers-Crypters-Protectors/PEiD-updated.shtml) (look for Download PEiD-0.95-20081103.zip. Make sure you can run it on your Windows VM.

Based on the slides and/or other resources, what does this tool do?

PEiD detects the most common packers, ciphers, and compilers for PE files. Before using this program, we need to know what a packed program is. Malware authors try to hide malicious malware programs. So they packed or obfuscated the program in order not to show the contents of the program. In short, the PEiD is used to determine which compiler or packer is being used.

4. Download [Dependency Walker](http://www.dependencywalker.com), and make sure you can run it in your Windows VM.

Based on the slides and/or other resources, what does this tool do?

Basically, it is a program that shows hierarchically all modules used in the relevant executable file. And you can see which functions are used in each module. For each module found, it lists all functions exported by that module and functions actually called by other modules. It can also detects many common application issues such as missing modules, invalid modules, import/export mismatches and etc.

**Part 2 – Textbook Lab 1-1**

1. Upload “Lab01-01.exe and Lab01-01.dll” to http://www.VirusTotal.com/ and view the reports. Does one of the files match existing antivirus signatures?

Add a screenshot and provide an explanation below:

2. When was ‘lab01-01.exe’ compiled?

Use the space below to explain where you found this header, and report the compile time.

3. Use PEiD to determine what tool was used to build the program. What is it? Are there any indications that either of these files is packed or obfuscated? If so, what are these indicators?

4. Use dependency Walker to and look for imports and exports of ‘lab01-01.exe’. Do any imports hint at what this program does? If so, which imports are they?

5. Use dependency Walker to and look for imports and exports of ‘Lab01-01.dll’. Does it import the same functions from kernel32.dll? What can you learn from this?

6. Are there any other files or host-based indicators that you could look for on infected systems?

7. What network-based indicators could be used to find this malware on infected machines?

8. When considering all your findings, can you make an educated guess about the purpose of these two files (.exe and .dll)?

**Part 3 – Textbook Lab 1-2**

In this section, we will analyze Lab01-02.exe.

1. Upload the Lab01-02.exe file to http://www.VirusTotal.com/. Does it match any existing antivirus definitions?

2. Are there any indications that this file is packed or obfuscated? If so, what are these indicators? If the file is packed, unpack it if possible.

3. Do any imports hint at this program’s functionality? If so, which imports are they and what do they tell you?

4. What host- or network-based indicators could be used to identify this malware on infected machines?