Module A3: FORWARD ENGINEERING

# Exploring PE HEADERS

**Lab Description:** The objective of this lab is to actively investigate a malicious binary to understand how binaries are built and the resulting structures in assembly language.

**Lab Environment:** This is intended to be an interactive lab with the instructor providing guidance to extend student demonstrations of the prerequisite knowledge of assembly language. The exercise should be run in a protected environment as it deals with REAL malware. It can be run in the SRE class virtual environment, which would contain the malware and allow instructors to view/step-in to the student environment even when not co-located.

**LAB EXERCISE/STEP 1**

The instructor needs to clearly express the following warning:

WARNING: The binary can be found in your VM (/code/malware/binary.exe), but can also be downloaded from https://nullify.cc/binary.exe NOTE: this is not a benign binary, and may trigger Anti-Virus or other similar security tools. DO NOT EXECUTE THIS BINARY ON ANY SYSTEM. The safest approach is to examine this within the environment you have been provided.

**LAB EXERCISE/STEP 2**

The purpose of this lab is to familiarize you with the use of file format specifications to understand the structure of binary files. In this lab we will stick to the widely used PE file format, but you will examine the file using nothing more than a hex editor / hex viewer.

**LAB EXERCISE/STEP 3**

Begin by opening the file pecoff\_v8.pdf. This is Microsoft’s PE file format specification. It can be found in your VM in the /docs/SRE directory. Next, open the assigned binary in a hex editor. This file can be found on your VM in the /code/sre/malware/ directory.

**LAB EXERCISE/STEP 4**

Explore the hexdump using the page up and page down functionality. Use the Intel manual to answer the following questions. Be prepared to defend your answer.

1. What is the file offset of the PE signature?

2. How many sections does the file contain?

3. What is the hex value of the File Header’s TimeDateStamp field?

4. What is the size of the optional header?

5. At what file offset does the Optional Header begin?

6. Is the file a PE32 or a PE32+ ?

7. What is the value of ImageBase for this file?

8. What is the Virtual Address (NOT RVA) of the AddressOfEntryPoint?

9. What is the RVA of the Import Table?

10. What is the file offset of the start of the section header array?

11. What are the names of each section?

12. Which section contains the import table?

13. What is the file offset of the import table?

14. How many libraries are references by the import table?

15. What are the names of these libraries?

# What to submit

This is intended to be an in-class exercise with the instructor and students walking through the solutions together as the process of arriving at the answer is more important than the answer and is easiest to capture through a walk-through. It is important to learn that there are multiple ways to arrive at these answers and to share these methods for better reverse engineering. Instructors should encourage creativity and diversity in solutions and allow students to walk through unique solutions for the entire class.

**Options –**

Instructors can assign individual students to individual questions.

Instructors can select a different file to examine. Some of the listed questions will not apply to other files.

Instructors could add extension assignments for students who have mastered this material providing additional files to evaluate.