Lab 02: Environment Set-up

In this lab, you will prepare a reverse engineering environment that you will be using throughout the semester. For the most part, you will be free to decide what software your prefer, but there are a couple of cases where you will need access to specific programs.

Purpose

To create a development environment suitable for most reverse engineering projects. This environment would ideally be ready to be used for all of the remaining labs this semester. Do not worry about missing one thing or another; most of the required tools are listed below. If something is required for a lab later that was not installed here, just install it and document the fact in the applicable lab report.

Procedure

You will need access to the following materials:

- A relatively powerful computer
  - Intel Core i5 with virtualization support, equivalent, or better
  - 4GB or more RAM
  - 250GB or more hard disk space
- PC virtualization software (e.g., VirtualBox)
- A 64-bit distribution of Linux that is actively maintained
  - Include packages for 32-bit compatibility
  - Include packages for a C development environment (i.e., gcc and binutils)
  - Include packages for debugging (i.e., gdb)
- A licensed copy of 64-bit Windows 10 or better
  - WoW64 should be included
- A copy of Visual Studio that can compile C/C++ to native (Community Edition is available free of charge)
- A copy of the Windows SDK with Windows Debugging Tools

The host OS may be chosen to suit your preference. You may set up the required components however you like, but your set up must provide the following capabilities. Please include a screenshot showing each of the 4 configurations launched in a debugger:

- Build and debug a native C application for 32-bit Windows
- Build and debug a native C application for 64-bit Windows
- Build and debug a native C application for 32-bit Linux
- Build and debug a native C application for 64-bit Linux

Tips and Caveats

- I recommend a laptop that you can bring to class, but a powerful desktop with remote access and a lesser laptop may suffice.
- Microsoft makes test images of Windows 10 available for free, which can be licensed a month or so at a time.
- MinGW is not a suitable substitute for Visual Studio or the Windows SDK for this lab. Microsoft compilers produce different assembly code than GNU compilers.
- Take a snapshot or clone your VMs after completing the lab. It's usually a good idea to have a known baseline from which to begin each new lab.

- You may choose to use your host OS as one of the required platforms. Just beware that environment may not be safe for the security-focused labs.

Report

Completion of this lab must be substantiated by a report. For this lab, work is to be completed individually. The report should follow a suitable structure. The Scientific Method and The Engineering Design Process are good sources of inspiration. In practice, the exact nature of the report will depend on the purpose of the project. For example, a reverse engineering project for developing a compatible program might place emphasis on the results, i.e., the derived documentation of applicable APIs. In a forensics project, the procedure and conclusion would likely be most important. For this class, the report should provide sufficient detail that another person could reasonably duplicate your results. Please include screenshots demonstrating completion of the requirements.

Grading

This first report will be graded on completion; however, feedback will be provided to assist you in future reports. All future reports will be graded more precisely. At a minimum, each report should describe the purpose of the lab, the procedure, any assumptions, problems, or hypotheses, and the final results. Mostly, I'll be looking for a proper problem statement, a structured and detailed procedure, and for a reasonably correct solution substantiated by the experimental results.

This lab is due midnight next week.