**ITT-210 SolarWinds Guide**

## Prerequisites

These prerequisites will help you understand the SolarWinds activity.

1. Research the SolarWinds hack.
   1. Define the term "supply chain attack."
2. Research CRC (cyclical redundancy check) and how it is used to verify software.
   1. You do not need to get deep in the math weeds.
      1. Explain the purpose of CRC.
      2. How CRC is used to validate software.
3. Research SDLC (software development life cycle).
   1. Define each SDLC stage.
   2. Address cybersecurity issues that should be considered at each stage.
4. Research, compile, and link stages of building software. (Assembly and C use the same process.)

## The SolarWinds Lab

Visual Studio will be used in a different way for this activity.

1. The activity requires you to compare good source code vs. bad source code. To better see the differences, install a file compare utility that will show you the differences between two text files. Download and install [WinMerge](https://winmerge.org/?lang=en) and read the documentation on the website to understand how to apply this to the lab.
2. From the assignment, download the SolarWinds.zip and expand its contents in a project folder.
3. In Visual Studio, open the folder where you have expanded SolarWinds:

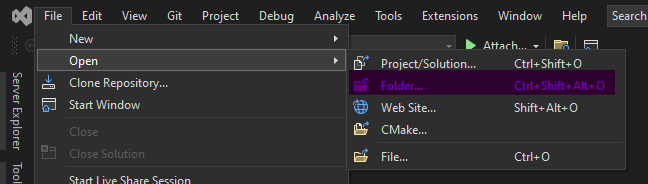


Figure 1: Visual Studio File Open

1. Your Solution Explorer view should look like this:

Figure 2 shows the Solution Explorer Folder View
Search Solution Explorer - Folder View
Topic 1 - SolarWinds (E:\projects\
crc8.asm
crc8.h
crc8-malware.asm
main.c

Figure 2: SolarWinds Files View from Visual Studio

1. The Visual Studio compiler and assembler are usually used by pressing the Run or Build buttons in Visual Studio. This assignment has you use the command-line version of these tools. To access these tools open in terminal:

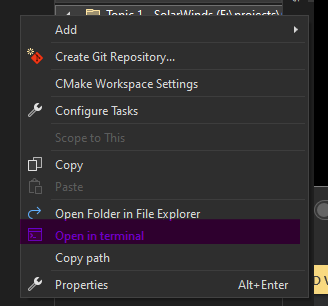


Figure 3: Open Visual Studio Terminal Window

1. From this terminal (Developer PowerShell), you will be able to run the command line versions of the C compiler and assembler:

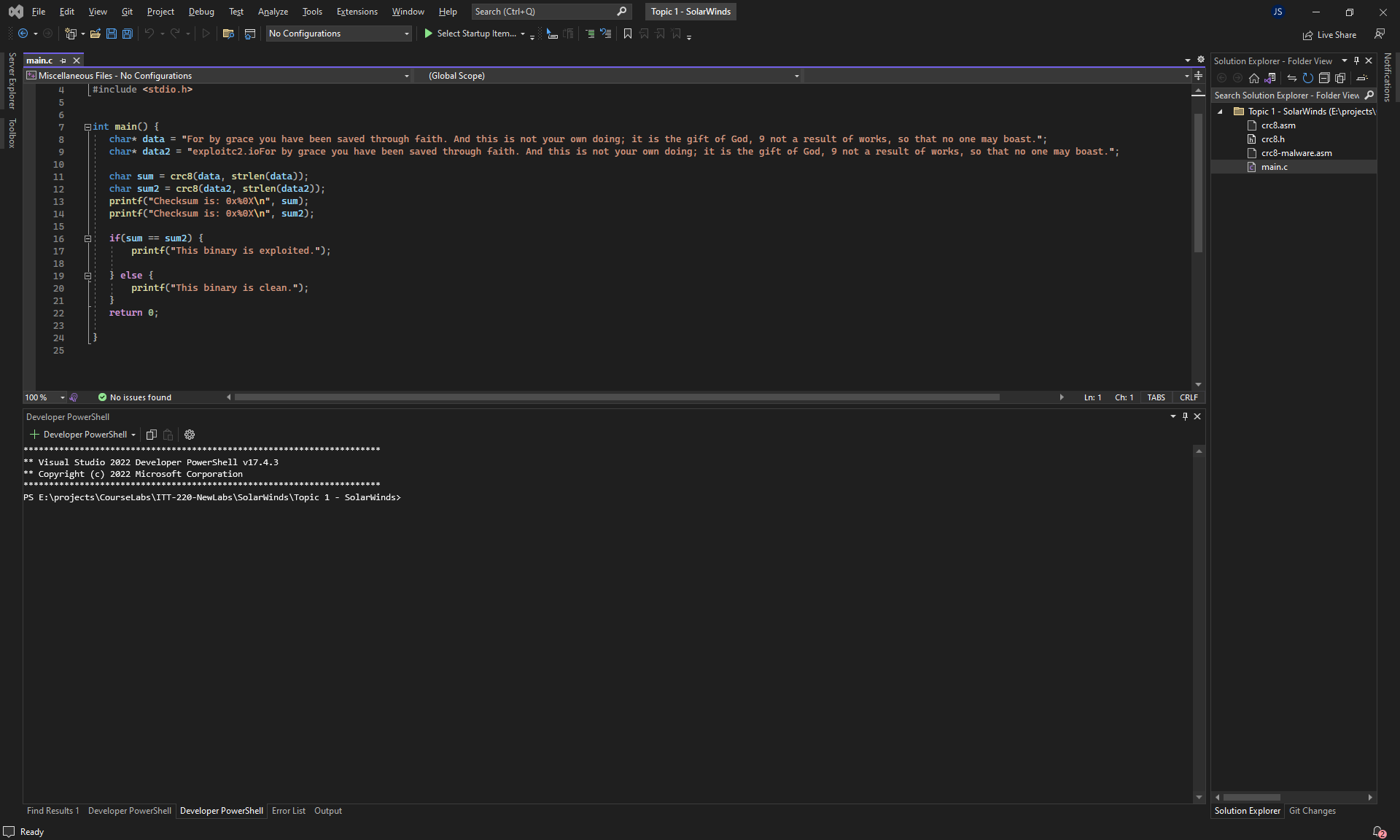


Figure 4: Visual Studio Terminal Window

* 1. cl is the c compiler and linker
  2. ml is the (macro) assembler and linker
     1. cl main.c /c
        1. compiles main.c
        2. produces main.obj (machine code)
        3. /c à compiler only, no link
     2. ml crc8.asm /c
        1. assembles crc8.asm
        2. produces crc8.obj (machine code)
        3. /c à assembler only, no link
     3. cl crc8.obj main.obj /Fecrc.exe
        1. links crc8.obj and main.obj à crc.exe

1. When you modify a file in Visual Studio, you will see an asterisk by the filename until you save the file. You must save (ctrl+s) the file before you compile (or assemble) since the compiler works with the copy on disk.

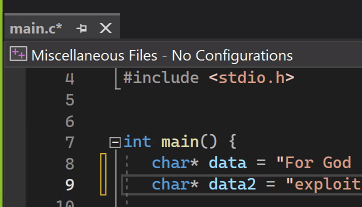


Figure 5: A modified file shown in Visual Studio

1. On some machines, the anti-virus will alert on these files. This is a false positive. If this happens to you, you can submit screenshots of the anti-virus message. You still must understand what the code is illustrating.