Module C3: static analysis for binary sre - Disassemblers and Decompilers.

# LET’s START ANALYZING BINARIES!

**Lab Description:** The objective of this lab is to allow you to choose appropriate static analysis tools and techniques to safely examine a binary file to achieve a specified objective.

**Lab Environment:** This is intended to be an interactive lab with the instructor providing guidance to extend student demonstrations of very basic static analysis skills. The exercise should be run in a protected environment as it may deal with malware. It can be run in the SRE class virtual environment, which would contain the crackme files and tools for static analysis. It would also 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: This lab may expose you to malware. Malware needs to be handled carefully. Malware may trigger Anti-Virus or other similar security tools. DO NOT EXECUTE THESE BINARIES ON ANY SYSTEM. The safest approach is to examine this within the environment you have been provided.

**LAB EXERCISE/STEP 2**

Using a disassembler or decompiler, perform the following tasks:

1. Reverse engineer (showing your work) 3 of the 64-bit binaries from the 0320\_crackme\_exec directory (either linux\_x64 or osx versions) to clearly show how you determined the password to be entered. One of the three you choose should be from the first five (i.e., crackme\_01 through crackme\_05), 1 from the second (i.e., crackme\_06 through crackme\_10), and 1 from the third (i.e., crackme\_11 through crackme\_15). Note that these are just 64-bit builds of the x86 binaries from the previous module.
2. Reverse engineer two of the CPP files from 0330\_crackme\_cpp that we did not reverse engineer in class (i.e., not crackme\_103 or crackme\_105) to clearly show how you would recover the password to be entered.

# What to submit

Students should present their results to the class.

Instructor Notes: These are 64-bit builds of the x86 binaries from the previous lesson.

Results will be greatly varied as student select difference file and approach the RE process and documenting the RE process differently. Prompts you might use during presentations include:

Why did you choose that tool?

Would (alternative tool) have performed better/worse/differently?

Would results have been different if you had use different options with the command?

Using just the file and your documentation, could another analyst fully understand the results and be confident that you did not make an error?

Using your provided analysis, could another analyst repeat your process and obtain the same results?