**Reverse Engineering Overview**

Reverse engineering is the process of taking compiled machine code program and converting it back into more human readable code. Essentially, we are trying to understand the functionality of the program.

Machine code or assembly code have been formatted for direct execution of a CPU. In general, converting to machine code is a one-way process for compiled languages. Machine code can be converted back into assembly however, it is more difficult to read and requires practice.

Assembly instructions perform various actions on registers. These include data movement, arithmetic operations and control-flow. Instructions are pieces of memory executed based on its address. Control flow is achieved by jumping or accessing instructions stored at a certain address. Addresses itself can be likened to the indices of an array, where the array is memory and memory addresses are the indices acting as a reference to an instruction.

The language to be used will be C and C++.

Disassemblers are tools which revert a machine code into assembly code. Various disassemblers are IDA, Binary Ninja and GNU Debugger (GDB). Binary Ninja was chosen mostly due to its aesthetics in presenting assembly code and it has a free version.

Decompilers attempt to convert compiled code back into pseudocode for further reconstruction. This gives a general outline of what the original source code looked like.