



C41 A Ladder Logic Decompiler for Supervisory Control and Data Acquisition (SCADA) Network Forensics

Irfan Ahmed, PhD, Virginia Commonwealth University, Richmond, VA 23284*

Learning Overview: After attending this presentation, attendees will understand a decompilation process of a binary ladder logic program under three cyberattack scenarios to support the network forensic analysis of SCADA systems.

Impact on the Forensic Science Community: This presentation will impact the forensic science community by discussing a new decompiler for a ladder logic program, which is useful for SCADA network forensics, and allows a forensic investigator to scan the network traffic dump of a SCADA network, identify any transfer of a ladder logic, further retrieve the logic from the dump, and analyze it using the decompiler.

An attacker can transfer a malicious program over the network to compromise a given target Programmable Logic Controller (PLC). PLC is an essential and critical component for the automation of industrial processes, such as gas pipeline, chemical and nuclear plants, and power generation and distribution. It has a control logic that defines how a PLC controls the actuators based on input devices, such as sensors. For instance, a ladder logic program can communicate to a PLC to turn off a water pump when water reaches a given level in a tank. Ladder logic is a popular programming language for PLCs. It consists of graphical symbols, which are placed together in AND/OR logic sequence to write a control logic. During an investigation, if forensic investigators find a ladder logic program in a network traffic dump, they can further extract the program from the dump and then decompile it for further forensic analysis.

Interestingly, when a program is compiled, RSLogix does not create its low-level representation of a ladder logic program on disk (such as an executable) that is used to run on a PLC. However, when RSLogix downloads a program to a PLC, it transfers the low-level representation that can be captured and extracted from the network traffic, which can be decompiled to understand the logic actually being transferred to a PLC. This approach is useful for SCADA network forensics. In a case of an incident, a forensic investigator can scan the network traffic dump of a SCADA network, identify any transfer of a ladder logic, further retrieve the logic from the dump, and analyze it using a decompiler.

In particular, the decompiler first identifies the start and end of each rung in a program and, further, parses each rung to identify a sequence of instructions and their AND/OR relationship. Differential analysis is used to map low-level representation with higher level, such as unique sequence of bytes that represent each graphical symbol. The decompiler is equipped with a comprehensive database of the mapping. This presentation also discusses a number of challenges in developing the decompiler, including developing signatures to identify the start and end of a rung, format of each instruction, which varies in size and is analogous to opcode and operand of Intel Assembly. Opcode in a ladder logic instruction has a specific byte-sequence; however, the operand may have a different representation with respect to the opcode.

A prototype implementation (tool) of the decompiler will be discussed under three cyberattack scenarios in which an attacker either hides a malicious control logic from RSLogix or compromises the RSLogix's ability to retrieve the logic from a compromised PLC remotely. The tool can analyze the network traffic and reveal any control logic (at source code level) present within the traffic.

Ladder Logic, SCADA Forensics, Network Forensics