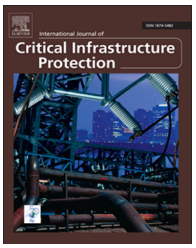
Available online at www.sciencedirect.com

ScienceDirect

www.elsevier.com/locate/ijcip



An evaluation of modification attacks on programmable logic controllers

Carl Schuett, Jonathan Butts, Stephen Dunlap*

Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio 45433, USA

ARTICLE INFO

Article history:
Received 18 October 2013
Accepted 29 January 2014
Available online 5 February 2014

Keywords:
Industrial control systems
Programmable logic controllers
Firmware
Modification attacks
Reverse engineering
ARM

ABSTRACT

Unprotected supervisory control and data acquisition (SCADA) systems offer promising targets to potential attackers. Field devices, such as programmable logic controllers (PLCs), are of particular concern because they directly monitor and control industrial processes. Although attacks targeting SCADA systems have increased, relatively little research has focused on exploring the vulnerabilities directly associated with the exploitation of field devices. Attacks such as Stuxnet have targeted operating characteristics, but not low-level firmware code. As attacks increase in sophistication, it is reasonable to expect increased exploitation of the field device firmware.

This paper examines the feasibility of modifying PLC firmware to execute remotely-triggered attacks. A general method is used to reverse engineer the firmware to determine its structure. After the structure is understood, the firmware is modified to add an exploitable feature that can remotely disable a PLC. The attacks described in this paper utilize a variety of triggers and leverage existing functions to exploit PLCs. Important segments of the firmware are described to demonstrate how they can be used in attack development. Finally, design recommendations are suggested to help mitigate potential weaknesses in future firmware development.

Published by Elsevier B.V.

1. Introduction

Supervisory control and data acquisition (SCADA) systems monitor and remotely control vital infrastructures such as oil and gas pipelines, electric power transmission networks and potable water distribution systems [16]. In recent years, attacks targeting SCADA systems have increased in scope and magnitude [17]. Aging equipment, proprietary hardware, limited processing capabilities and geographical distance are factors that hinder the implementation of low-cost and viable protection solutions [9].

To date, attacks on SCADA systems have primarily focused on high-level systems (e.g., human machine interfaces) or network protocols (e.g., Ethernet and MODBUS) [12]. Even Stuxnet, one of the most sophisticated cyber weapons, targeted high-level application software and did not directly exploit low-level field device code [7]. Surprisingly, little research has focused on the exploitation of field device firmware [3].

Programmable logic controllers (PLCs) collect data and interact with sensors, motors, valves and other devices positioned throughout vast industrial systems to streamline process automation and control [4]. By seizing control of a PLC, an attacker can directly affect the outcome of, or interfere with, the underlying industrial process. As attacks increase in sophistication, it is likely that attackers will target the PLC firmware for exploitation. Such attacks could have devastating consequences.

*Corresponding author.
E-mail address: jonathan.butts@afit.edu (J. Butts).

http://dx.doi.org/10.1016/j.ijcip.2014.01.004
1874-5482 Published by Elsevier B.V.

This paper focuses on the exploitation of field devices. The goal is to determine the feasibility of developing firmware-based attacks that specifically target PLCs. The attacks are intended to demonstrate the ability to disable PLC functionality while remaining undetected. Having understood the nature and scope of the attacks, solutions and strategies can be developed to mitigate the threats.

2. Background

SCADA systems comprise three basic components: (i) a central control station; (ii) field devices; and (iii) communications links between the control station and the field devices [5]. PLC field devices located at the edge of a SCADA network are the focus of this paper. A PLC is an embedded device that contains programmable memory for executing sequences of instructions that collect data from attached sensors and transmit the data to a central operations center. A PLC can also translate instructions into actuator movement based on the inputs from the attached sensors or a direct command from the operations center.

As shown in Fig. 1, the PLC architecture incorporates three notional layers: (i) hardware; (ii) firmware; and (iii) ladder logic or programmable area [3]. The intermediate firmware layer provides the interface between the hardware and programming layers. Note that all the functions made available in the firmware must map to a hardware implementation [8].

Embedded devices such as PLCs use the firmware as the operating system because the size and memory capacity of the devices render it impractical to implement an additional software-based operating system. Because of this design, the PLC firmware is often full-featured and offers a variety of services, including remote administration via a web server. Note that many PLCs allow remote firmware updates, providing convenience to end-users as well as attack vectors to adversaries.

3. Exploitation methodology

This section describes the PLC firmware exploitation methodology along with the experimental testbed used in the research.

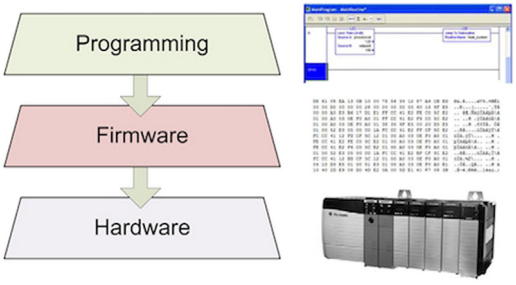


Fig. 1 – PLC architecture layers [5].

3.1. Approach

A key vulnerability associated with PLCs is the inherent trust of the firmware verification process, which relies on a CRC and a checksum as a validity mechanism [10]. The CRC and the checksum are both tested to verify that the firmware has not been corrupted, but this mechanism cannot detect intentional tampering [3].

This paper focuses on attacks targeting PLCs where the PLC functionality is not adversely impacted until desired by the attacker. The primary goal of the research was to test the feasibility of developing, deploying and concealing PLC firmware repackaging attacks. This was accomplished by first reverse engineering the PLC disassembled code to match known device functions. Next, instructions were inserted into the firmware without adversely affecting the stability of the software. Following this, the firmware was repackaged to execute denial-of-service (DoS) attacks under the following conditions:

- Force the PLC to terminate operations after a predetermined amount of time.
- Force the PLC to terminate operations upon receiving a control signal.
- Force the PLC to terminate operations upon receiving a control signal and make a permanent modification to the device that prevents its owner from regaining control of the device.

3.2. Experimental testbed

Fig. 2 shows the experimental testbed configuration. The test equipment comprised a Controllogix 1756-L61 controller with a 1756-ENBT Ethernet module connected via a 1756 PA75-B chassis. A Windows XP workstation served as the management and engineering console, which contained Allen-Bradley

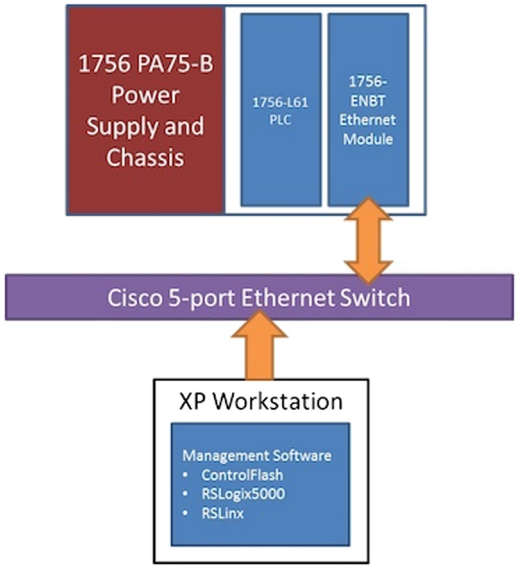


Fig. 2 – Experimental testbed configuration.