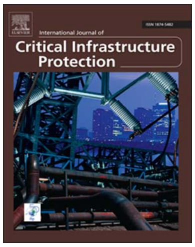
Available online at www.sciencedirect.com

ScienceDirect

www.elsevier.com/locate/ijcip



Using timing-based side channels for anomaly detection in industrial control systems

Stephen Dunlap[a], *Jonathan Butts*[b], *Juan Lopez*[c], *Mason Rice*[a,*], *Barry Mullins*[a]

[a]*Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio 45433, USA*

[b]*QED Secure Solutions, 417 Forest Ridge Drive, Coppell, Texas 75019, USA*

[c]*Applied Research Solutions, 51 Plum Street, Beavercreek, Ohio 45440, USA*

ARTICLE INFO

Article history:
Received 4 January 2016
Received in revised form
3 April 2016
Accepted 21 July 2016

Keywords:
industrial control systems
Programmable logic controllers
Modification attacks
Side channels
Anomaly detection

ABSTRACT

The critical infrastructure, which includes the electric power grid, railroads and water treatment facilities, is dependent on the proper operation of industrial control systems. However, malware such as Stuxnet has demonstrated the ability to alter industrial control system parameters to create physical effects. Of particular concern is malware that targets embedded devices that monitor and control system functionality, while masking the actions from plant operators and security analysts. Indeed, system security relies on guarantees that the assurance of these devices can be maintained throughout their lifetimes. This paper presents a novel approach that uses timing-based side channel analysis to establish a unique device fingerprint that helps detect unauthorized modifications of the device. The approach is applied to an Allen Bradley ControlLogix programmable logic controller where execution time measurements are collected and analyzed by a custom anomaly detection system to detect abnormal behavior. The anomaly detection system achieves true positive rates of 0.978–1.000 with false positive rates of 0.033–0.044. The test results demonstrate the feasibility of using timing-based side channel analysis to detect anomalous behavior in programmable logic controllers.

1. Introduction

During a siege in 590 BCE, Solon of Athens poisoned the water supply of the town of Cirrha using hellebore roots [20]. The contaminated water incapacitated the unsuspecting Cirrhaeans with uncontrollable diarrhea and the Athenians quickly overwhelmed the city [20]. In 2000, Vitek Boden leveraged unauthorized wireless access to a sewage treatment plant to release 800,000 l of raw sewage into public water supplies in Australia [22]. In this incident, one malicious actor without direct physical access was able to cause a significant environmental impact. Due to increasing network connectivity, critical infrastructure systems are more susceptible to malicious attacks than ever before.

Modern society relies on industrial control systems (ICSs) to automate the operation of the critical infrastructure. Historically, industrial control systems were considered to be secure due to their isolation from external networks. Recently, however, industrial control systems have become less isolated as they incorporate commodity information technologies to improve efficiency and decrease costs [25]. The trend to interconnect industrial control devices exposes

*Corresponding author.
E-mail address: mason.rice@afit.edu (M. Rice).

http://dx.doi.org/10.1016/j.ijcip.2016.07.003
1874-5482/Published by Elsevier B.V.

them to external networks and potential threats. Common devices, such as programmable logic controllers (PLCs), often lack security mechanisms such as authentication and encryption [25]. Stuxnet [8] demonstrated that significant damage can occur if a programmable logic controller in an industrial facility is compromised. Stuxnet also demonstrated that network protection mechanisms such as intrusion detection and air gapping, while important and effective, are unable to protect sensitive systems from sophisticated attackers.

A programmable logic controller has three layers: (i) hardware layer; (ii) firmware layer; and (iii) programming layer [4]. Building and maintaining trust in a device requires validating all three layers against a known-good baseline. The hardware is the lowest layer that executes the firmware. The firmware handles functionality such as communications, programming layer execution and error handling. Compromising the firmware of a device enables an attacker to cause negative effects and hide them from operators. The programming layer, also called the application layer, is designed to perform high-level system tasks. The applications in this programmable logic controller layer are often implemented using ladder logic programs. Stuxnet [8] demonstrated that the application layer of a programmable logic controller can be modified to cause significant damage to the controlled physical systems.

2. Background

This paper describes a novel technique for fingerprinting a programmable logic controller after it has been deployed. The fingerprint is developed by monitoring the execution characteristics of the device while it is in operation. The device is then fingerprinted periodically to verify that it has not been modified intentionally or unintentionally. The timing characteristics of the programmable logic controller are determined by the firmware and ladder logic programs that execute on the device. A baseline is created based on these characteristics and deviations from the baseline are flagged as anomalies.

This research employed an Allen Bradley ControlLogix L61 CPU module. Allen Bradley is a brand of Rockwell Automation, the second largest global supplier of programmable logic controllers [28]. The ControlLogix series of programmable logic controllers includes a task monitor utility capable of remotely collecting CPU execution times. This utility was designed to enable engineers to determine the CPU resources used by the firmware and ladder logic programs. Execution times are reported in microseconds and segmented into individual user tasks and firmware services such as communications. The ControlLogix utility provides sufficiently accurate timing information with minimal invasiveness. Alternative methods of collecting timing characteristics must considered for other programmable logic controllers.

Note that the effect of conditionally executed subroutines was not explored in this study. Subroutines in a ladder logic program may be designed to execute when certain conditions are met. These conditions and subroutine size can vary drastically between applications. Future research will attempt to characterize the possible effects and determine if the technique should be improved to handle these issues.

2.1. Programmable logic controller security

Most industrial control system security efforts have focused on the traditional information technology aspects of control systems [18,25]. The information technology level refers to devices such as computers and firewalls, but it specifically excludes control system components and embedded devices such as programmable logic controllers. Cyber attacks on control systems are expected to increase in the future. The threats posed by cyber attacks are heightened by vulnerabilities in systems that support the critical infrastructure [27]. It is expected that adversaries will target the most vulnerable components in industrial control systems. Protecting the traditional information technology components of industrial control systems is necessary, albeit not sufficient, to secure critical infrastructure systems.

Programmable logic controllers are a major focus of industrial control system security efforts. A programmable logic controller is often a fragile device that is prone to failure from cyber-induced events, especially one that presents unexpected inputs such as malformed packets to the device [25]. Programmable logic controllers are designed to be reliable and easy to manage under normal conditions; unfortunately, most devices are not designed with security as a primary requirement. As such, simple security features such as authentication and encryption are often not implemented. Without these protection mechanisms, programmable logic controllers are highly vulnerable targets. Firmware, ladder logic programs and device configurations are also susceptible to manipulation by an adversary intent on disrupting control system operations [14].

Ensuring that a programmable logic controller has not been maliciously modified after its deployment is a major challenge. The device must be analyzed comprehensively to demonstrate that it can be trusted. This requires the validation of the hardware, firmware, ladder logic programs and configuration settings. Firmware validation is an important first step toward establishing trust [4]. After a device has been shown to be trustworthy, however, the trust must be revalidated at periodic intervals. Verifying that a device has not been modified is the most direct means of maintaining trust. Firewalls and intrusion detection systems are important, but they are not enough [3]. A general method for confirming the correct operation of a device must be developed to ensure that it has not been modified in a malicious manner.

2.2. Side channel analysis

A side channel is an unintended avenue by which information can be obtained by an observer; this is also referred to as an information leak [10]. Side channel information is typically leaked due to the physical requirements of devices [1]. Monitoring the physical aspects of a device can provide information about its internal state and operation [29]. A side channel analysis attack occurs when an adversary uses leaked information to learn secrets contained within a system [1,6,10]. The targets of these attacks are typically