Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

An unsupervised anomaly-based detection approach for integrity attacks on SCADA systems

Abdulmohsen Almalawi [a,c], Xinghuo Yu [b], Zahir Tari [a], Adil Fahad [a,d,*], Ibrahim Khalil [a]

[a] School of Computer Science and Information Technology, RMIT University, Melbourne, Vic. 3001, Australia
[b] School of Electrical and Computer Engineering, RMIT University, Melbourne, Vic. 3001, Australia
[c] Faculty of Computing and IT King Abdulaziz University, Jeddah, Saudi Arabia
[d] Department of Computer Science, Al-Baha University, Al-Baha City, Saudi Arabia

ARTICLE INFO

Article history:
Received 7 November 2013
Received in revised form
25 May 2014
Accepted 13 July 2014
Available online 29 July 2014

Keywords:
Unsupervised detection
Cyber-warfare
SCADA systems
Intrusion Detection System
Consistent/Inconsistent SCADA Patterns

ABSTRACT

Supervisory Control and Data Acquisition (SCADA) systems are a core part of industrial systems, such as smart grid power and water distribution systems. In recent years, such systems become highly vulnerable to cyber attacks. The design of efficient and accurate data-driven anomaly detection models become an important topic of interest relating to the development of SCADA-specific Intrusion Detection Systems (IDSs) to counter cyber attacks. This paper proposes two novel techniques: (i) an automatic identification of consistent and inconsistent states of SCADA data for any given system, and (ii) an automatic extraction of proximity detection rules from identified states. During the identification phase, the density factor for the *k*-nearest neighbours of an observation is adapted to compute its inconsistency score. Then, an optimal inconsistency threshold is calculated to separate inconsistent from consistent observations. During the extraction phase, the well-known fixed-width clustering technique is extended to extract proximity-detection rules, which forms a small and most-representative data set for both inconsistent and consistent behaviours in the training data set. Extensive experiments were carried out both on real as well as simulated data sets, and we show that the proposed techniques provide significant accuracy and efficiency in detecting cyber attacks, compared to three well-known anomaly detection approaches.

1. Introduction

SCADA systems control and monitor industrial and infrastructure processes such as transportation, oil and gas refining and energy and water distribution networks (Yu et al., 2011; Fahad et al., 2013). In recent years, the incorporation of Commercial-Off-The-Shelf (COTS) products such as standard hardware and software platforms have begun to be used in SCADA systems. This incorporation allowed various products from different vendors to be integrated with each other to build a SCADA system at low cost. In addition, the integration of standard protocols (e.g. TCP/IP) into COTS products has increased their connectivity, thereby increasing productivity and profitability. However, this shift from proprietary and customized products to standard ones exposes these systems to cyber threats (Oman et al., 2000). Undoubtedly, any attack targeting SCADA systems could lead to high financial losses

* *Corresponding author.*
E-mail addresses: alharthi.adil@gmail.com, dr.alharthi.fahad@gmail.com (A. Fahad).

http://dx.doi.org/10.1016/j.cose.2014.07.005
0167-4048/Crown Copyright © 2014 Published by Elsevier Ltd. All rights reserved.

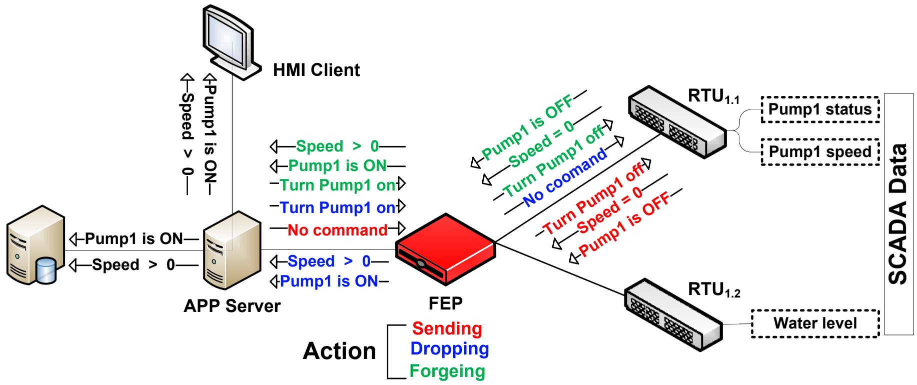

Fig. 1 – Compromised FEP sends undesired command and falsifies the feedback information.

and serious impacts on public safety and the environment. The attack on the sewage treatment system in Maroochy Shire (Australia) is an example of such attacks on critical infrastructures (Slay and Miller, 2007), where the attacker took over the control devices of a SCADA system. The Stuxnet (Falliere et al., 2011) worm, which was designed to damage nuclear power plants in Iran, is a recent example of threats targeting control systems. Both of the aforementioned attacks are classified as man-in-the-middle (MITM) attacks, where control devices are compromised to perform malicious actions, and meanwhile false information is sent to the Master Terminal Unit (MTU) to avoid detection. Such cyber threats allow attackers to perform high-level control actions (Wei et al., 2011; Queiroz et al., 2011; Nicholson et al., 2012), and pose potential threats to SCADA systems.

An awareness of the potential threats to, as well as the need to reduce the various vulnerabilities of SCADA systems have recently become an important research focus in the area of security. A number of (security) measures have been used in traditional IT systems, including management, filtering, encryption and intrusion detection. However, such measures cannot be directly applied to SCADA systems without considering their specific characteristics. Additionally, none of these traditional IT security solutions can completely protect SCADA systems from potential cyber attacks. However, properly adapting/extending such IT solutions can create robust protection of SCADA systems against cyber attacks. IDS (Intrusion Detection System) is one of the security solutions that has showed promising results in detecting malicious activities in traditional IT systems, and this is one of the reasons for using and adapting it to SCADA environments.

1.1. *Problem statement*

To illustrate the intrusion detection problem, two well-known scenarios (Verba and Milvich, 2008) are considered. Fig. 1 illustrates an attacker compromising the front end processor (FEP) by carrying out three actions: (i) initialising a connection with a remote terminal unit ($RTU_{1.1}$) and sending a command without receiving a corresponding command from the application server; (ii) dropping the command sent from the application server to $RTU_{1.1}$, and frogging feedback information sent back to the application server to meet the attack; and (iii) frogging the command sent from the application server to $RTU_{1.1}$, as well as frogging feedback information sent back from $RTU_{1.1}$ to the application server. All commands sent to $RTU_{1.1}$ will be trusted, as they are syntactically valid and sent from an FEP.

Two inconsistent data can be identified in this scenario: an **inconsistent network traffic pattern** and (ii) an *inconsistent SCADA data*. The former relates to the following: (i) an FEP is not an intelligent device that can make a decision and send a command to $RTU_{1.1}$ without receiving a corresponding command; (ii) and the dropped command at FEP will be shown up in the network stream from the application server to the FEP, but not in the network stream from the FEP to the $RTU_{1.1}$, while the frogged commands between the application server and $RTU_{1.1}$ can be identified by the inconsistent SCADA data.

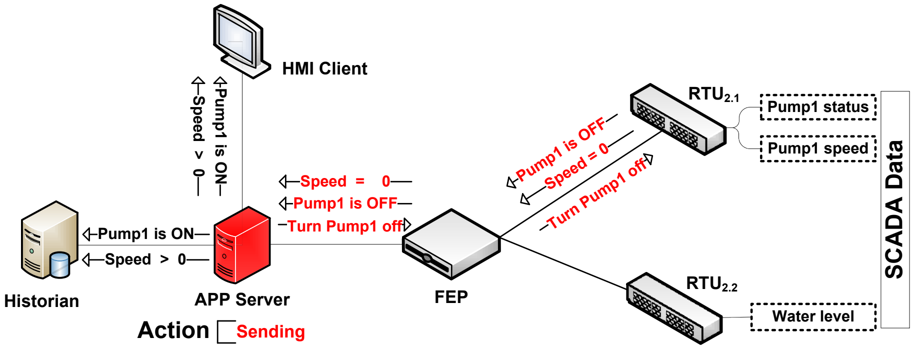

Fig. 2 – Compromised application server sending false information.