12-14-2013

Cyberthreats, Attacks and Intrusion Detection in Supervisory Control and Data Acquisition Networks

Wei Gao

Cyberthreats, attacks and intrusion detection

in supervisory control and data acquisition networks

By

Wei Gao

A Dissertation
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in Computer Engineering
in the Department of Electrical and Computer Engineering

Mississippi State, Mississippi

December 2013

Cyberthreats, attacks and intrusion detection

in supervisory control and data acquisition networks

By

Wei Gao

Approved:

_____

Thomas H. Morris
(Major Professor)

_____

Sherif Abdelwahed
(Committee Member)

_____

Donna Reese
(Committee Member)

_____

David Dampier
(Committee Member)

_____

James E. Fowler
(Committee Member)

_____

Achille Messac
Dean
James Worth Bagley College of Engineering

Name: Wei Gao

Date of Degree:December 14, 2013

Institution: Mississippi State University

Major Field: Computer Engineering

Major Professor:Thomas H. Morris

Title of Study: Cyberthreats, attacks and intrusion detection in supervisory control and data acquisition networks

Pages in Study:134

Candidate for Degree of Doctor of Philosophy

Supervisory Control and Data Acquisition (SCADA) systems are computer-based process control systems that interconnect and monitor remote physical processes. There have been many real world documented incidents and cyber-attacks affecting SCADA systems, which clearly illustrate critical infrastructure vulnerabilities. These reported incidents demonstrate that cyber-attacks against SCADA systems might produce a variety of financial damage and harmful events to humans and their environment. This dissertation documents four contributions towards increased security for SCADA systems. First, a set of cyber-attacks was developed. Second, each attack was executed against two fully functional SCADA systems in a laboratory environment; a gas pipeline and a water storage tank. Third, signature based intrusion detection system rules were developed and tested which can be used to generate alerts when the aforementioned attacks are executed against a SCADA system. Fourth, a set of features was developed for a decision tree based anomaly based intrusion detection system. The features were tested using the datasets developed for this work.

This dissertation documents cyber-attacks on both serial based and Ethernet based SCADA networks. Four categories of attacks against SCADA systems are discussed: reconnaissance, malicious response injection, malicious command injection and denial of service. In order to evaluate performance of data mining and machine learning algorithms for intrusion detection systems in SCADA systems, a network dataset to be used for benchmarking intrusion detection systemswas generated. This network dataset includes different classes of attacks that simulate different attack scenarios on process control systems. This dissertation describes four SCADA network intrusion detection datasets; a full and abbreviated dataset for both the gas pipeline and water storage tank systems. Each feature in the dataset is captured from network flow records. This dataset groups two different categories of features that can be used as input to an intrusion detection system. First, network traffic features describe the communication patterns in a SCADA system. This research developed both signature based IDS and anomaly based IDS for the gas pipeline and water storage tank serial based SCADA systems. The performance of both types of IDS were evaluates by measuring detection rate and the prevalence of false positives.

Keywords: Intrusion detection system, SCADA, Vulnerability, Dataset, Network Security

.

DEDICATION

To my family

ACKNOWLEDGEMENTS

I would like to thank Dr. Thomas H. Morris for giving me his generous assistance for my research. Thanks Dr. Sherif Abdelwahed, Dr.Donna Reese and Dr. David Dampier for their help and comments on my study and dissertation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Background

Supervisory Control and Data Acquisition (SCADA) systems are computer-based process control systems that interconnect and monitor remote physical processes. SCADA systems collect data from remotes facilities about the state of the physical process and send commands to control the physical process creating a feedback control loop. SCADA systems are widely used in chemical processing, petroleum refining, electrical power generation and distribution, water purification and distribution, intelligent buildings and nuclear plants.

Contemporary SCADA systems are distributed cyber-physical systems. These systems consist of Remote Terminal Units (RTU), Master Terminal Units (MTU), Human Machine Interface software (HMI) and the sensors and actuators that interface with the physical system. Remote terminal units are connected to sensors and actuators to interface directly with a physical process. RTU commonly store control parameters and execute programs that directly control the physical process. For instance, an RTU may be used to control the water level in a tank. It will be programmed with a high water level a low water level. The RTU continuously monitors the water level with a connected water level sensor. If the water reaches the programmed high level, the RTU turns off a pump that fills the tank. If the water level reaches the low level, the RTU turns on the pump to

add water to the system. The RTU, the ladder logic (or other programmation), and the attached sensors and actuators form a feedback control loop.

Master Terminal Units (MTU) are connected to the RTU via SCADA network communication links. The MTU polls the RTU periodically to read physical quantities of the controlled system such as a voltage, pressure, and water level. Typically this information is displayed on a Human Machine Interface (HMI) to allow operators to monitor the physical process. HMI typically allow the operator to monitor the physical process. HMI typically also allow the operator to interact with the physical process. Operators may change operating parameters. For example an operator may change the high and low water levels set points, from the previously mentioned water system example. The MTU, RTU, communication link, HMI, and operator form a second supervisory feedback control loop. The communication links in a SCADA network can be thought of as occupying layers 1, 2 and 7 of the OSI model; these are physical, data link (and media access control), and application layers respectively. The physical layer may be wired or wireless. Wired networks may use leased lines, category 5 or 6 cables, serial cable, and/or fiber optic cable. Wireless links may also use proprietary non-standard protocols. Finally, wireless links may include very long distance solutions such as satellite and microware links. SCADA communication network protocols include Fieldbus, Profibus, MODBUS, DeviceNet, Distributed Network Protocol version 3 (DNP3), EtherNet/IP and ControlNet. All of these protocols have versions that have been adapted to use TCP/IP as network and transport layers for use with commodity Internet equipment.

Fig 1 [i]illustrates a typical SCADA architecture. The Field devices consist of RTUs, the Programmable Logic Devices (PLCs) and the Intelligent Electronic Devices (IEDs). Field devices are connected to the SCADA network and the SCADA network are connected with the corporate intranets.

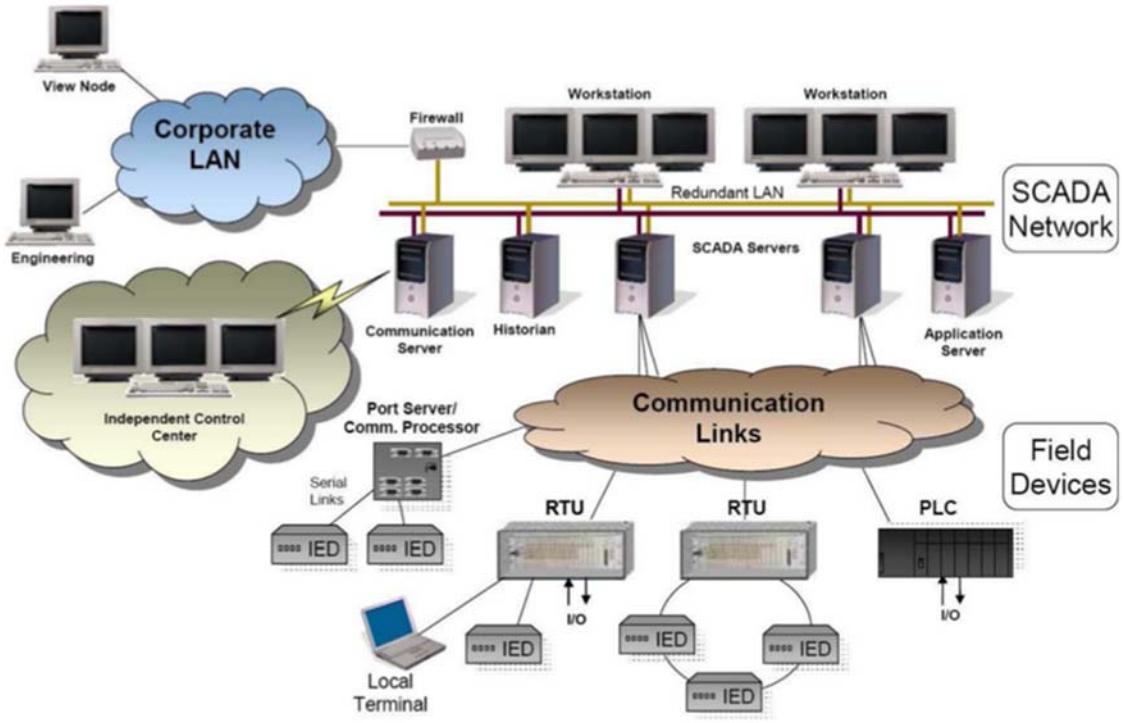

Figure 1 SCADA network architecture

SCADA systems have a strategic importance due to the fact that they are adopted by the Critical Infrastructure (CI) of nation. The President's Commission on Critical Infrastructure Protection (PCCIP) defined CI as "whose incapacity or destruction would have a debilitating impact on our defense and economic security" [ii]. CI is associated with energy, transport and utilities. Any damage to CI may impact on the economy of a country.

There have been several real-world documented incidents and cyber-attacks affecting SCADA systems, which clearly illustrate critical infrastructure vulnerabilities. These reported incidents demonstrate that cyber-attacks on SCADA systems might produce a variety of financial damage and harmful events to humans and their environment.

Team Cymru, a specialized Internet security research firm, released a briefing paper in 2008 [iii] which discussed malicious port scan activity against their DarkNet (a honey pot) searching for open ports on port numbers commonly associated with SCADA system network protocols. This report showed heavy scanning activity from four areas: Asia, North America, Western Europe and Eastern Europe. The report cited heavy scanning of DNP3 ports from Russia and Taiwan and heavy scanning activities for MODBUS related ports in Western Europe and China. This port scanning is potentially indicative of attackers searching for SCADA systems for later attacks.

Stuxnet [iv] is the first known worm to target an industrial control system. Stuxnet targeted PC's running the Siemens WinCC SCADA software product. Infected systems had a DLL replaced used by the WinCC Step7 tool. The worm then monitored communications between the WinCC tool and a remote terminal. If a specific signature related to the remote terminal was found firmware on the remote terminal was replaced with malicious code. The malicious code on the remote terminal caused the physical process to misoperate while continuing to inform operators that the system was functioning correctly.

On January 2000, an ex-employee of a contracting company attacked the Maroochy Shire Council's sewage control system in Queensland, Australia. A pump in

the control system failed to start or stop when specified and an alarm failed to alert. This attack made approximately 264, 000 gallons of raw sewage leak to nearby rivers [v].

In 2003, the Davis-Besse nuclear plant in Oak Harbor Ohio was attacked by the Slammer Worm which made a safety monitoring system of the plant offline for approximately five hours [vi].

The following sections provide motivation information on intrusion detection and vulnerabilities analysis of SCADA system, and list the contributions of this research.

Motivation

National Electric Reliability Council (NERC) Critical Infrastructure Protection (CIP) Standards 002-3 through 009-3 [vii] require utilities and other responsible entities to place critical cyber assets within an electronic security perimeter. The electronic security perimeters must be subjected to vulnerability analyses, use access control technologies, and include systems to monitor and log the electronic security perimeter access. The Federal Energy Regulatory Commission (FERC) requires responsible entities involved in bulk electricity transmission to adhere to the NERC CIP 002-3 through 009-3 standards. No such regulation exits for the electric distribution systems and other critical infrastructure, such as water treatment and distribution and gas distribution in the United States. Electronic perimeter security will minimize the threat of illicit network penetrations, however, persons with electronic access to SCADA systems within the electronic security perimeter still remain a threat.

There are many security mechanisms that can secure process control systems such as access control via passwords and firewalls, data diodes, virus scanning, encryption, and digital signatures. These solutions are required for defense in depth for SCADA

systems. However, none of them are completely secure. Attacks continue to defeat security mechanisms to penetrate systems and execute attacks against confidentiality, integrity and availability. Because current security mechanisms can never be 100% secure, intrusion detection systems are needed to monitor attacks in progress.

The original goals of SCADA network protocols were to provide reliable communication to transmit system information and control commands between MTU and RTU. Furthermore, RTU, programmable logic controllers (PLC), programmable automation controllers (PAC), and intelligent electronic devices (IED) have limited computing capability and network bandwidth which limits their ability to provide comprehensive cyber security features. Many widely used protocols in SCADA systems also lack security features [viii]. For instance, MODBUS[ix], DNP3[x] and Allen Bradley's Ethernet Industrial Protocol [xi]lack of authentication capability leads to the potential for network penetrators and disgruntled insiders to inject false command and false response network packets into a SCADA system either through direct creation of such packets or replay attacks. Due to the fact that the core of the control flow of every SCADA system is the communication protocols [xii], one possible solution to improve security is to extend the protocol to add authentication. There is a secure version of the MODBUS protocol [xiii] which can add integrity, authentication, no-repudiation and anti-replay mechanisms. In order to secure the DNP3 protocol based SCADA system, Flexi-DNP3 [xiv] (Flexible Distributed Network Protocol Version 3) is designed by using DNPSec [xv].

Patching each RTU in order to use modified protocols is difficult. In a large scale SCADA system, there are maybe thousands of remote devices such as PLC and IED.

These devices may be in remote locations. As such it will be very difficult to patch every device. Further a single network may include devices from multiple vendors.

One common important feature of SCADA systems is 24/7 operation. Upgrading communication protocols usually means downtime of the SCADA system, which should be considered before implementation of a modified protocol.

Similarly with IT security area, the new threats to SCADA system increase day after day. Even if a modified secure protocol can prevent already known attacks to process control system, it cannot ensure to stop new attack. The SCADA system still may require continual patching.

The interconnection of SCADA networks with corporate networks and with the Internet increases efficiency and productivity of the process control system. Operators or companies may collect real time system information by using corporate networks and take control of the devices remotely. However, this integration leads to more security issues [xvi]. It may open up the SCADA network to corporate network vulnerabilities. Attackers may gain access to the SCADA network from the corporate network. Possible attack processes are described as follow. Attackers may exploit existing vulnerabilities of the operating system of computers or servers in corporate network. From these computers or servers, attackers may gain user name and password credentials to gain access to the SCADA system. Malware may be installed through a web browser or email client on a computer, server, printer, and later gain access to the SCADA network to take control of remote devices. Malicious insiders who have full access authorization may access the SCADA system to change the configuration of the SCADA network or of individual SCADA system components.

Since the integration of a SCADA network and a corporate network may introduce more threats to SCADA systems, one solution to enhance security is to add firewalls, using access control lists and implement cryptographic protection to enhance system security. Firewalls are required between SCADA networks and corporate networks to block illegal access and malicious behavior targeting the SCADA network. Access Control Lists (ACL) will block unauthorized user who try to access remote devices in process control systems. However, ACLs also have some issues [xvii]: attackers may bypass access control in the gateway. Second, if the SCADA network has thousands of remote devices, the access control list will be very large which may lead to long delay which may be unacceptable for hard real time SCADA systems. Finally, malware may infect machines with allowed by ACL to communicate with SCADA system devices.

Applying cryptography [xviii][xix] is another alternative solution to improve SCADA system security. Communication time delay is an issue when implementing cryptographic methods for SCADA systems control systems [xx]. SCADA control systems have real time requirements. Cryptography should not add delay which causes real time requirements to not be met. Cryptography requires secure key distribution to be effective. Many papers, including [xxi][xxii][xxiii], have proposed key management systems for SCADA systems. Unfortunately, none of these solutions have led to actual deployments. Wright et al. [xxiv] present an encryption protocol for a SCADA system based on cyclic redundancy codes (CRC), however, there have been no progress updates on this research since 2006. Additionally, other researchers have criticized this encryption as unable to provide adequate protection against attacks. Kang and Kim [xxv] applied a symmetric encryption algorithm and added a device on the MTU side and RTU

side of the network to encrypt and decrypt network packets. Experimental results show the time delay from this method does not affect the communication. However, it may be difficult to implement this encryption methodology for SCADA systems with large numbers of remote devices due to key distribution issues and the cost of deployment.

The idea of intrusion detection is a well-established approach in the traditional IT security area. A SCADA network is different from conventional IT networks several ways [xxvi] [xxvii]. First, SCADA networks are real time systems. Second, network availability is critically important to allow operators to continually monitor and control SCADA systems. Third, computer capability and memory resources in SCADA devices is limited which means it is difficult for existing devices to be upgraded to add security enhancements. In order to monitor network traffic and detect malicious behaviors in SCADA systems a SCADA-specific Intrusion Detection System (IDS) is required.

The KDD Cup 1999 dataset [xc] (here after "KDD dataset") provides labeled dataset for intrusion detection researchers in the information technology (IT) area. The KDD dataset includes approximately 5 million connection records which are divided into four categories of attacks: probing attacks, denial of service (DOS) attacks, User to Remote (U2R) attacks, Remote to Local (R2L) attacks, and normal connection records. Each connection record includes 41 network related attributes. Probing attacks are information gathering attacks which map the network to gather information, like which services are available, which ports are open or what type of operating system runs in the target host. DOS attacks lead hosts to run out of computing capability or memory resources so that legitimate requests for service are refused. U2R attacks aim to provide an attacker with root or administrator privileges. R2L attacks aim to gain local access to

local machines from remote machines. Table 1 lists categories of attacks and attack names found in the KDD dataset.

Table 1 KDD99 Dataset Attacks by Category

NAME OF SYSTEM CATEGORIES OF ATTACKS	ATTACK NAME
Probing	ipsweep, nmap, portsweep, satan
Denial of Service (DoS)	back, land, neptune, pod, smurf,teardrop
User to Root (U2R)	buffer_overflow, loadmodule, perl, phf,rootkit
Remote to Local (R2L)	ftp_write, guess_passwd, imap

Many researchers in the SCADA security area use datasets developed in house to test intrusion detection systems. There is no uniform dataset which includes normal network traffic and attacks against a SCADA network which can be used by researchers as a benchmark to compare the IDS performance. In order to evaluate performance of data mining and machine learning algorithms for intrusion detection systems in SCADA systems, a network dataset used for benchmarking intrusion detection systems should be generated. This network dataset should include different classes of attacks which simulate different attack scenarios on process control system networks

Research Contribution

This dissertation includes four major contributions. First, a set of cyber-attacks against a gas pipeline, water storage tank, and chemical processing systems are described. Attacks are grouped into 4 categories: reconnaissance, malicious command injection, malicious response injection, and denial of service. Second, a dataset was created which

includes network traffic captured from gas pipeline and water storage tank networks during normal operation and during periods of attack. The datasets can be used to benchmark SCADA intrusion detection systems. Third, a set of signature based intrusion detection rules were developed to detect the aforementioned attacks against the gas pipeline and water storage tank systems. Fourth, a set of input features were developed for use with a decision tree anomaly based intrusion detection system. The new input features were leverage physical system state information found in SCADA network packets to predict future control actions and process measurements. Both the signature based and anomaly based intrusion detection systems were evaluated. Evaluation included calculation of accuracy using detection rate and false positive rates. Both instruction detection systems were found to have high accuracy and low false positive rates. Analysis of intrusion detection system accuracy is provided for each class of attack.

Flow records in SCADA network communication not only describe network traffic patterns such as remote device addresses, function codes etc, but the payload of the packets also provides much more information to describe the control flow or state of the control system.

This research develops SCADA network intrusion detection datasets and extracts features from network flow records. This dataset groups two different categories of features that can be used for input to the intrusion detection system. First, network traffic features describe the communication pattern in a SCADA system. Compared with enterprise IT network, process control system topologies are static and the services provided by the system are regular. Some attacks against control systems may change the communication pattern in network, so network traffic features can be used to describe

normal traffic patterns and detect malicious activity. Another category of features are payload features for process control systems including system measurements such as air pressure, water level and the system setting point.

The signature based intrusion detection approaches monitor the traffic record and match the traffic pattern to known misuse patterns or rule to detect rules violations. Snort [xxviii]is an open source signature based network intrusion detection system. This research applies Snort IDS with signature rules combined with system state prediction rules to detect malicious behaviors in SCADA networks.

Anomaly IDS uses statistical models or machine learning algorithms to classify network traffic as normal or abnormal (or into smaller sub-classes). Various model types or classifiers can be used to build the anomaly IDS. This research applies Bayesian network, neural network and decision tree to classify network transactions as normal or abnormal. Decision tree algorithm is selected due to the fact that it is one of most useful techniques in deterministic classification. Bayesian network is a fundamental statistical approach to the problem of pattern classification. Neural network is also a widely classification algorithm in intrusion detection system. So these three classification algorithms are selected in this research.

It is hypothesized that building a SCADA system intrusion detection dataset to evaluate SCADA IDS performance is possible through the SCADA system threat model analysis and different exploiting methods both serial based and Ethernet based SCADA network. Most SCADA systems are designed for using a long period, the topology of SCADA network is static and the network communication traffic pattern is regular. Therefore intrusion detection system can use anomaly traffic pattern to detect abnormal

communication traffic. Furthermore, process control systems are based on the feedback control loop, the system measurements are inputs to the controller such as PID controller. Based on the system current measurements, system setting parameters and controller parameters, the next measurements can be predicted.

Organization

The remainder of this dissertation is organized as follows. Chapter II provides a literature review. The literature review is broken into a threat analysis of SCADA networks, literature on attacks against SCADA systems, signature based intrusion detection systems, anomaly based intrusion detection systems, and SCADA specific intrusion detection systems. Chapter III discusses a set of cyber-attacks which target two serial based SCADA systems; the gas pipeline system and the water storage tank system. Attacks against an Ethernet based chemical processing system are also described. Attacks are grouped into four categories; reconnaissance, malicious response injection, malicious command injection, and denial of service. Chapter IV describes four data sets created from capturing network traffic from the gas pipeline and water storage tank systems during normal operation and when the attacks from Chapter III were executed against these systems. There are four datasets total. The first datasets contain the complete set of traffic. The second pair is a scaled down set to enable faster processing when evaluating intrusion detection systems. Chapter V discusses implementation of signature based and anomaly based intrusion detection. Chapter VI make the conclusion of this dissertation.

CHAPTER II

LITERATURE REVIEW

Threats Analysis of SCADA Networks

Many works were found which describe threats to SCADA systems and networks. SCADA systems are exposed to serious vulnerabilities that can be exploited by attackers. Research on cyber threats and vulnerabilities show the security challenges and the secure approaches in SCADA networks. In the [xxix][xxx][xxxi], the authors analyze potential cyber threats to SCADA systems and explain the technological challenges for SCADA systems. These papers highlight that SCADA control systems connected to the internet are subject to reconnaissance, fabrication, and Denial of Service attacks from the internet The SCADA network architecture based on the open standard communication protocols renders the system more vulnerable to cyber-attacks. Valentine et al. explain software vulnerabilities in the ladder logic for the PLC and shows that current programming practices do not protect against logic errors and malicious code level attacks on SCADA systems [xxxii]. Edward Chikuni and Maxwell Dondo [xxxiii] investigate the security challenges in electrical power systems. This paper analyzes the SCADA computing vulnerabilities which include hardware (RTU, IED, master, etc) vulnerabilities and software vulnerabilities. Dong Wei et al. [xvii] demonstrate potential cyber-attacks and their impacts on the power grid and classify these attacks into three categories, attacks on automation devices, attacks on protocols, and attacks on topologies. Attacks on

automation devices occur when attackers take advantage of the vulnerabilities of the devices such as PLCs, IEDs to launch attack against the SCADA system. Attacks on protocols occur when attackers use the vulnerabilities of the SCADA network protocols to attack the SCADA system. Attacks on topologies occur when attacks take advantage of the vulnerabilities of topology design of the SCADA system to attack. In the [xxxiv], the authors describe the difference of security issue between general IT network and industrial process network, and the application of security concepts and tools. In the IT community, the primary security issues usually include confidentiality and integrity, while in control systems, the primary security issues usually include data integrity and availability. Gardner et al. [xxxv] discusses an RTU failure vulnerability for power infrastructure. In [xxxvi], McDaniel et al. present security and privacy issues for the smart grid. McDaniel et a primarily discuss data integrity issues for software and smart meters in the smart grid Byres [xxxvii] lists attack goals, technical difficulty, severity of impact, probability of detection for the critical infrastructures. Byres concluded that the detection probability of compromising MTU writing data to RTU and compromising RTU writing data to MTU is low.

In order to analyze the vulnerabilities and cyber threats of SCADA systems, many researchers focus on building a SCADA testbed or simulation. A simulation tool for building the SCADA system which can test the effect of attacks on real devices and applications is proposed in [xxxviii]. The presented simulation tool supports intergradations of external devices into the simulation system. The attacks presented in this work only include DOS attack and spoofing. In [xxxix], Wang et al. provide a SCADA simulation environment that can model vulnerability exploitations. This

environment can provide Modbus, Profibus, network traffic. However, this simulation environment does not include any simulated plants or control systems. Fovino et al. [xii], present details of a physical power plant emulator and demonstrates attack scenarios that can drive the simulated system into critical state. In [xl], Queiroz et al. propose an architecture of a modular SCADA testbed based on MODBUS protocol and describe a distributed Denial of Service attack on that testbed. Mallouhi et al. [xli] propose a SCADA security evaluation testbed and preliminary attacks against it. The SCADA testbed can provide simulated environment to SCADA security researcher. an alternative to simulating SCADA control systems is to provide network data logs of captured during execution of attacks against SCADA control systems and capture while a system is behaving normally. Such datasets are less expensive and can therefore be made available to many researchers. A contribution of this dissertation is a set of datasets for SCADA security researchers.

Attack on SCADA Systems

In [xlii] [xliii], the authors investigate the vulnerabilities of a SCADA system which monitors and controls the Gignac irrigation canal system located in South France. These two papers present a linearized shallow water partial differential equation (PDE) system that can model water flow in the Gignac canal network which uses lateral offtakes for water withdrawal. The authors develop a stealth deception attack based on the knowledge of switching the PDE parameters and proportional (P) boundary control actions to steal water from the pool using the offtake. The authors tested the attack with experiments in a simulation environment and in the Gignac irrigation canal system. These attacks can enable attackers steal water from the canal.

Jie Yan et al [xliv] identifies vulnerabilities and develops cyber-attack scenarios against a wind farm SCADA system. These scenarios include sending malicious commands to the wind turbine, a Man-In-The-Middle attack to send false measurement data to the wind farm operator, and internal attacks. These attacks may cause economic loss and equipment damage problems in the power system.

Dillon Beresford [xlv] at the 2011 Blackhat conference introduced reconnaissance, fingerprint, replay, authentication bypass, and remote attacks against a Siemens Simatic S7 PLC. This paper analyzes the vulnerabilities of the S7 PLC and the PROFINET protocol and introduces the MetaSploit Auxiliary S7 PLC scanner module.

Terry Fleury, Himanshu Khurana and Von Welch [xlvi] propose a taxonomy of attacks against energy control systems. This paper presents an Attack Vulnerability Damage (AVD) model which includes classes of cyber-attacks, energy control system vulnerabilities, and the potential damages. The authors discuss several attacks such as probe, flood, bypass, terminate, execute, modify and deletion attacks.

Mallouhi et al. [xlvii] presents SCADA cyber-attacks against a MODBUS TCP protocol based testbed. This paper discusses two types of attack, MODBUS and TCP attacks. The MODBUS attacks include a Denial of Service attack which issues illegal commands from a compromised HMI and a Man-In-The-Middle attack. The TCP attacks include a TCP SYN flood attack and a TCP ACK flooding attack.

In [xlviii], Sridharet al. discuss data integrity attacks and a Denial of Service attack against a SCADA control system. The data integrity attacks include the Min attack and the Max attack [xlix]. The goal of this type of data integrity attack is to misguide the operator into making decisions based on malicious fake data. The DOS attack in this

research makes the sensing signal unavailable to the control room or makes the control signal unavailable to the physical devices.

Le et al. [[l]] present a false data injection attack that can lead to financial loss against the state estimation algorithm used in a deregulated electricity market. These attacks affect the system by compromising numbers of sensors and sending false measurements to Regional Transmission Organizations (RTO) in order to makea profit from the market. In [[li]], another group of false data injection attacks against the state estimation algorithm in electric power grids are introduced. These false data injection attacks can bypass current bad measurement detection or state estimation techniques. The authors analyze two attack scenarios. The first scenario is called Limited Access to Meters. In this case, the attacker can only access some specific meters. In this scenario, random false data injection attacks and targeted false data injection attacks are developed. Another scenario is called Limited Resources to Compromise Meters. In this scenario, attackers have limited resources to compromise meters. Experimental results show that the attacker can efficiently construct attack vectors in both scenarios, which can change the state estimation result and modify the predicted state.

In [lii], Dong et al. introduce an event buffer overflow attack against the DNP3 protocol. There are two types of event buffers in the DNP3 protocol, the sequence of events buffer and the most recent event buffer. The authors identify a vulnerability in the sequence of events buffer due to the fact that it is fed by all slaves from whom a data aggregator acquires data and uses a first-come-first-serve rule in buffer. Buffer flooding attacks launched on a DNP3 network can lead to a high load flow that can occupy most

of the bandwidth. In this situation, the sequence of events buffer will drop normal packets.

Intrusion Detection System

There are two main types of detection techniques known as the signature based detection and the anomaly based detection. Signature based IDS use an attack signature database to determine whether the signature has been triggered. Signature based IDS are effective for detecting known attacks and have small false positive rates. Anomaly based IDS use mathematical techniques such as machine learning or data mining to check for deviations from a predefined normal behavior model. Anomaly based detection may detect previously unknown attacks, also known as zero day attacks. Anomaly based IDS typically have lower detection accuracy than signature based IDS and a higher rate of false positives.

Signature Based IDS

Signature based IDS focus on matching signatures stored in databases with information in network packets. This approach is efficient when detecting known attacks. Within the area of SCADA security, control system security researchers have developed signature based IDS which monitor the MODBUS network transactions watching for signatures of known attacks and vulnerabilities.

Quickdraw [liii] is an application to enhance the Snort IDS [liv] to create security log events for a SCADA control system. The snort preprocessors allow the Snort detection engine to check the packets using industrial control system protocols such as Modbus, DNP3. However, the signature database depends on the security expert

knowledge and experiences. Furthermore, signature based IDS cannot detect new attacks, so they are often behind attackers. Many industrial control system devices are not updated with patches on a regular basis and therefore may not include the latest security patches. Therefore a signature based IDS is a valuable tool to detect older attacks which may not be possible against a patched system.

Anomaly Based IDS

Anomaly based IDS use statistical models or machine learning algorithms to classify network traffic as normal or abnormal (or into smaller sub-classes). Various model types or classifiers can be used to build an anomaly based IDS, including neural networks, linear methods, regression models, and Bayesian networks [lv].There are two types of inaccuracies in IDS: false positives and false negatives. False positives [lvi] [lvii] generate a false alarm when there is no intrusion, while false negatives miss an actual intrusion. The accuracy of anomaly intrusion detection systems relies on training dataset completeness, proper input feature development, and the choice of classifier.

Wang and Stolfo [lviii] present a payload based anomaly network intrusion detection system called PAYL. PAYL models the normal application payload of network traffic using an unsupervised method. A byte frequency distribution profile and the standard deviation of the payload are used to train the IDS classifier. In the detection stage, PAYL uses the Mahalanob is distance to compare the new record against the pre-defined profile. The Columbia University Computer Science (CUCS) department network (CUCS) datasets were captured from a university network and include attack traffic and normal traffic. The evaluation results show that PAYL can successfully detect the attacks in CUCS dataset.

Artificial neural networks (ANN) model the biological nervous system and are widely used in pattern classification, intrusion detection, and statistical analysis. ANN are one of the most commonly used classifiers for intrusion detection systems [lix, lx, lxi, lxii, lxiii]. In [54], the authors use a neural network combined with an analytic hierarchy process. The experimental results of this research show that the artificial neural network is suitable to solve multiple issues of intrusion detection systems such as regular updating, detection rate, false positive rate, false negative rate, suitability and flexibility. In [56], Fan et al. describe different ANN based classifiers such as Multilayer Perception (MLP) [lxiv], Generalized Feed-Forward (GFF), Modular Neural Network (MNN), Jordan/Elman network (JEN), Principal Component Analysis (PCA) Network [lxv], Self Organized Maps (SOM) [lxvi], and Recurrent Network (RN) [lxvii].

Decision trees are another useful technique for pattern classification. A decision tree consists of non-leaf nodes and leaf nodes. Each non-leaf node represents a test over an attribute and each leaf node corresponds to a class of decision result. There are many algorithms for the decision tree which includes ID3 [lxviii], C4.5 [lxix] and CART [lxx]. Most decision tree algorithms implement a top down strategy, i.e. from the root to the leaves. There are two steps in the decision tree classification: constructing the tree and applying the tree to a dataset. There are many works using decisions tree to detect attacks and malicious behaviors. Sheen et al. [lxxi] combine the decision tree algorithm and a feature selection algorithm by using KDD99 dataset. In [lxxii], Juan et al. apply the C4.5 algorithm to build a decision tree based intrusion detection system.

SCADA-Specific Intrusion Detection

The Computer Science Laboratory at SRI International developed a model based intrusion detection system [lxxiii] for MODBUS TCP protocol based SCADA networks. This research applied three model based detection algorithms to monitor MODBUS TCP traffic. The first algorithm is the protocol-level model, which is used to characterize MODBUS TCP requests and responses based on the expected characteristics of MODBUS specified in [lxxiv] [lxxv]. This algorithm focuses on detecting invalid MODBUS function codes, dependent fields such as data addresses, the number of data items, and the length of packets, since these features may indicate reconnaissance attacks or Denial of Service attempts. The protocol-level model applies customized Snort rules [lxxvi] to detect violations of the above specifications. The second algorithm is the expected communication patterns model which specifies expected MODBUS TCP protocol based network communication patterns. This algorithm aims to detect attacks that violate these expected patterns. The last algorithm is a learning-based approach for detecting changes in server or service availability. This algorithm checks server and service availability by using a Bayesian network to detect Denial of Service attacks. This research provides the systematic approach to build the model based intrusion detection system on MODBUS TCP based SCADA network. The experimental validation shows that the model based IDS are effective for detecting attacks in the SCADA network. However, model based IDS could cause many false alarms.

Linda, Vollmer and Manic [lxxvii] use the combination of a neural network learning algorithm, the Error Back-Propagation algorithm [lxxviii], and the Levenberg-Marquardt algorithm [lxxix] to build an intrusion detection system for critical

infrastructures. This research used software tools such as Nmap, Nessus [lxxx] and MetaSploit [lxxxi]to generate intrusion packet records. The network features used to train the neural network in this paper are derived from the packet header and include the target address, and the function code. One technique used in this research is Window Based Feature Extraction. This algorithm captures a number of neighboring packets in a single vector to calculate statistical features. Some significant window based attributes are the number of IP address in the window, the time length of the window, and the average interval between packets. These attributes are selected since they can describe a network traffic record accurately. The experimental result of this work provides evidence that neural network based intrusion detection is a promising approach for monitoring process control systems.

Yang et al [lxxxii] apply the Auto Associative Kernel Regression (AAKR) model and the Statistical Probability Ratio Test (SPRT) to design an intrusion detection system for a simulated SCADA system. AAKR is a nonparametric, empirical modeling method that uses historical observation to output classification decisions for new instances. The SPRT was introduced in [lxxxiii], which was used to decide whether a new instance is normal or abnormal. In order to simulate a SCADA system, the authors used several SUN servers and workstations to setup the system. They used the continuous system telemetry harness (CSTH) [lxxxiv][lxxxv], which was developed by Sun Microsystems to monitor the server activity and build an initial profile of the system's normal working status. The method used in this research is pattern matching. This technique is used to detect anomalies by analyzing deviations from normal behaviors in the network. The traffic profile is created to describe the normal behavior which consists of link utilization, CPU

usage, and login failure. The instance consists of these predetermined behaviors that represent SCADA network behavior. The AAKR model is used to predict the "correct" version of the new instance. The correct version is compared with past observations which present normal behaviors, and the result is input to SPRT model to decide whether it is abnormal. The experimental results show that this intrusion detection system can detect anomalous behaviors in the SCADA network efficiently. However, the real SCADA network traffic profile may be different from traffic available from this simulated system. Furthermore, the authors consider the Simple Network Management Protocol (SNMP) as the most important data source of the network traffic statistics, which may not applicable for a real SCADA network.

Oman and Phillips [lxxxvi] designed an intrusion detection and event monitoring system for a SCADA network to monitor the settings changes on SCADA devices and monitor malicious commands sent to a RTU. This method uses XML to describe system information details about the SCADA device such as IP address, telnet port, and legal commands for the device. Each command has an entry in the RTU's XML profile. A Perl program parses that profile and creates a Snort signature for that command. The second component of this IDS is device settings change monitoring. The device settings change monitoring component is implemented by maintaining a settings repository for the SCADA network. Each device has one or more baseline settings files in this repository. Device settings will be compared with predetermined and protected baseline settings. This IDS also provides revision control in order to archive settings for later review and to help guard against operator error. Furthermore, this work provides uptime monitoring by verifying whether a command succeeds. If the command is succeeds then the path is

healthy. This intrusion detection and event monitoring systems is useful to the SCADA system operators as well as for securing the SCADA system. However, this work current only detects and monitors malicious behaviors against an RTU. It doesn't provide any detection solution to protect the MTU.

Carcano et al [lxxxvii] present an approach for intrusion detection for a SCADA system based on multidimensional critical state analysis. The fundamental methodology is monitoring the evolution of the SCADA system's state. In order to track and analyze the evolution of a SCADA system, four elements are needed. The system description and critical state representation use the Industrial State Modeling Language (ISML). The state evolution monitor tracks the evolution of the system state by sniffing the network traffic between the master and the slave to gain the information about the system. This work measures the Manhattan Distance between the current state and the critical state in a boiling water reactor system. The experimental results include the false positive rate and the false negative rate, memory usage performance, and packet capturing performance. The results show this research is effective as a SCADA system IDS. However, the attacks in this work only include malicious behaviors which will lead the system to critical state. In real attacks against a process control system, attackers need to identify the devices in the network first and then launch other types of attacks. Some attacks will not force the system into a critical state. These attacks must still be detected. In order to detect attacks against a SCADA system, the IDS should detect all classes of attacks or multiple IDS should be used to detect all classes of attacks.

Zhang et al [lxxxviii] created the Smart Grid Distributed Intrusion Detection System (SGDIDS) for a three layer smart gird network by using a support vector machine (SVM)

classifiers and a artificial immune system (AIS) classifier. This research divides the smart grid into three layers, home area networks (HAN), neighborhood area networks (NAN) and the wide area network (WAN). The SGDIDS has an intrusion detection module to detect security threats in each layer. The distributed intrusion detection system consists of multiple intelligent modules; the information acquisition module (IAM) gathers the network traffic and energy consumption information. The Data Segmentation module (DSM) and the Preprocessing Module (PM) processes the data to make it suitable for the detection algorithm. The analyzing module (AM) detects malicious intrusions. SVM [lxxxix] was chosen as the classification algorithm due to the fact that it is highly accurate. An artificial immune system is also selected to make a comparison with SVM algorithm. To evaluate the performance of the intrusion detection system, The KDD99 dataset [xc] was used to evaluate the performance of the intrusion detection system. This research shows that the SVM algorithm based intrusion detection system can detect attacks targeting a smart grid network and the architecture of the SGDIDS is a promising approach for monitoring the smart grid network traffic. However, the KDD99 dataset is based on communication in computer networks, not on the smart grid. The protocols, the architecture of the network and the features of the records are significantly different. The KDD99 dataset cannot show the real behavior of a smart grid network.

Valdes and Cheung [xci] use a learning based communication pattern anomaly detection technique to detect anomalous network traffic patterns from a process control system. This approach is based on the assumption that the process control systems have fairly regular communication patterns between the master and the slave, as well as a static device address space. The IDS classifies network traffic into normal, anomalous,

and attacks using an adaptation of the pattern anomaly detection technique [xcii]. This approach checks patterns in a stream of packets by using the competitive learning algorithm. If a pattern matches existing patterns, according to a similarity function returning a value above a specified threshold, then the library pattern wins. The pattern is a vector of features such as source and destination IP address and destination port. The IDS maintains a database of recent and historical network flow profiles in order to evaluate the current pattern. The anomalous network flow in this work includes scanning the network using the Nmap and the vulnerability analysis using the Nessus and modifying data points on a MODBUS server. The detection results of the flow anomaly detection show that this IDS is able to detect anomalous flows effectively. However, this work includes limited attack traffic since most of the anomalous traffic is derived from scan activities. In addition, the authors do not provide the false positive rate for the IDS.

Linda, Manic and McQueen [xciii]present the Known Secure Sensor Measurements (KSSM) method to detect malicious false control system state by using physical measurements. This research applied a subset of secure physical measurements that are sent in sequence after unsecure measurements used for control. The comparison of the selected subset value and KSSM values reveals potential malicious false data. The KSSM system consists of a KSSM control module and the communication network. The KSSM control module consists of Signal Analyzer and the Sensor Selector. The Sensor Selector uses a number of sensors each time randomly and Signal analyzer monitors selected sensor data and detects malicious false data injection. The experimental results shows the selection algorithm of KSSM can demonstrate the current pattern of the network traffic and the show the speed of the KSSM data in different bandwidth setting profiles.

However, authors didn't give the detection rate performance of the KSSM. Furthermore, this paper only demonstrates that the KSSM can detect false measurement attacks in the process control system.

Rrushi and Kang [xciv] adapted the estimation algorithm using a statistical approach for anomaly based intrusion detection in SCADA systems. The algorithm evaluated the malicious credibility of the payload based on checking the effect the packet will have on a variable. The variables including input register variables, holding register variables, etc of control systems are stored in a matrix. The attack credibility is calculated by using applied logistic regression analysis and maximum likelihood estimation [xcv]. The probability is used to classify the element vectors in the matrix as normal or abnormal. An experimental evaluation was implemented using a MODBUS TCP protocol based test bed using MatPLC modules [xcvi]. MatPLC is a PLC simulation program for Linux operating system. The attacks in this evaluation include shellcode injection, stack buffer overflow, heap buffer overflow, pointer overwrites, array out of range, and inertial attack [xcvii]. The false positive rate is zero and the detection rate is 98% which is very good. However, this experiment uses a small simulated system. The authors do not demonstrate the algorithm's efficiency in a large scale process control system.

Chee et al. [xcviii] present a anomaly based intrusion detection algorithm that can be used to detect malicious behaviors in substation control system. In their former research, the proposed RAIM model [xcix] was introduced. The RAIM was used to monitor and detect different types of malicious behaviors such as bad password and large file transfer. In this paper, the authors cited four attack patterns of attack against power systems: intrusion reconnaissance, modify the file system, modify the target's settings,

and change the target state. A vector which includes four weight factors is used to model the four patterns mentioned above. A matrix stores a historical set of vectors for the same substation. The intrusion possibility is calculated based on the distance between maximum and average of values of a row vector. The experimental results of this research show the vulnerability possibility and other system information such as how a attacked substation can make more attack effort on the entire power system. However, this research does not include detection rate and false positive rate result.

Table 2 Comparison of the SCADA Intrusion Detection

NAME OF SYSTEM	PUBLISH YEAR	DETECTION PRINCIPLE	IMPLE-MENT	THREAT MODEL	PROTOCOL
SRI Modbus[lxxiii]	2007	Specification	yes	gain access, reconnaissance, attack host and server	Modbus
NNIDSCI[lxxvii]	2009	Anomaly	yes	use Nmap, Nessus, MetaSploit	N/A
AKKR-SPRT[lxxxii]	2006	Anomaly	yes	DOS attack simulated by SUN servers	SNMP
IDAEM[lxxxvi]	2008	Signature	yes	attack on RTU	N/A
Multidimensional-CSA[lxxxvii]	2011	Specification	yes	simulated attack on critical state of system	Modbus
SGDIDS[lxxxviii]	2011	Anomaly	yes	KDD 99	N/A
Pattern detection[xci]	2009	Anomaly	yes	reconnaissance	Modbus
KSSM[xciii]	2012	Anomaly	yes	false data injection	N/A
Statistical parameter estimation[xciv]	2009	Anomaly	yes	overflow exploits	Modbus
RAIM[xcviii]	2011	Anomaly	yes	change system setting, file system and status	IEEE C37.118

Table 1summarizes the recently published SCADA-Specific intrusion detection systems presented in this chapter. A generic drawback of these research works is their threat models only include subsets of the attack classes presented in this work; reconnaissance, measurement injection, command injection, and denial of service. Each work also used unique datasets created by the researchers specifically to evaluate the IDS presented in each paper. Exploit coverage is very limited for each of the datasets presented in Table 2. Some of them only have reconnaissance attack to the process control system, while some of them only include response injection attack. The malicious behaviors in these datasets don't cover some very critical attacks. For this reason, it is very hard to measure the performance of the presented IDS. Since each IDS is evaluated with a unique dataset it is impossible to compare the IDS. This literature search highlights two needs for SCADA security IDS research. First, there is a need for a comprehensive taxonomy of attacks against SCADA control systems. Second, datasets which include normal SCADA system network traffic and traffic captured during attacks is needed. Such datasets should be provided to researchers in the SCADA IDS field to provide a common means for comparison of different IDS approaches. This dissertation provides comprehensive attack taxonomy, describes datasets created to support IDS research needs, and provides an evaluation of two IDS approaches for SCADA security systems

CHAPTER III

EXPLOITS ON SCADA SYSTEMS

SCADA systems are widely used in different critical infrastructures, such as water distribution, gas transmission, and nuclear power plants. Outside attackers with knowledge of the specific control system or insiders can launch attacks against applications of the SCADA system. Attackers may perform malicious command injection attacks to take control of the remote devices such as RTU. This type of attack may push the system into a critical state, damage field equipment, or cause financial loss if the attack leads to incorrect process operation. Intruders can also launch malicious false data injection attacks against HMI software which can hide real conditions of remote field equipment, or mislead operators into taking wrong actions based on the false system information such as measurement, field equipment status.

Many protocols used in SCADA systems lack security features. Attackers can perform passive reconnaissance attacks such as traffic analysis and eavesdropping. During reconnaissance attacks, attackers can gather control system network information, map the network architecture, and identify device characteristics such as manufacturer, model number, supported network protocols, system addresses, and system memory maps.

DOS attacks aim to stop the proper functioning of the SCADA system. Attackers can flood meaningless packets to the network to interrupt communication between MTU

and RTUs to make devices unavailable, break the feedback control loop and stop the sensors from reporting the latest measurements. Many devices such as switches and programmable logic controllers (PLC) in SCADA networks have vulnerabilities due to the fact that these devices have limited resources. Many industrial switches are vulnerable to simple denial of service attacks. There are many attacks which target specific PLC. For example, attackers may send malicious payloads to shut down or restart the network module of a PLC or an attacker may send malicious payloads to shut down or restart the CPU module of a PLC.

This dissertation describes network vulnerabilities of three SCADA systems; a gas pipeline, a water storage tank, and a chemical processing station. The water storage tank control system models oil storage tanks found in the petrochemical industry. Petrochemical refineries use oil and oil byproducts to produce gasoline, kerosene, diesel, and many types of plastics. In a common configuration, oil arrives by sea and is pumped into oil storage tanks to provide a consistent supply of oil to the refinery operation. Oil storage tank control systems similar to the one modeled by the water tank control system are used to monitor oil inventory and distribute oil to refinery processes. The gas pipeline control system models a gas pipeline used to move natural gas or other petroleum products to market. The chemical processing control system models a chemical mixing operation. Chemicals from two tanks are mixed in a third tank following a recipe defined by an operator using human machine interface software.

Serial Communication Based Control Systems

The MSU SCADA security laboratory (henceforth called the lab) was used to develop a set of SCADA control system exploits. The lab contains five laboratory scale

SCADA control systems [c]. Each SCADA control system includes Control Microsystems, INC SCADA Pack Light PLCs as MTU and RTU. The MTU is connected via RS-232 serial port to a PC running the OE IFIX human machine interface software. The MTU and RTU are connected wirelessly using integrated Freeware 900MHz radios. The MTU and RTU can be configured to communicate using DNP3, MODBUS ASCII, or MODBUS RTU communication standards. The five laboratory scale control systems model a gas pipeline, a factory assembly line, a water tower, a water storage tank, and an industrial blower. All five control systems are mechanically functional models.

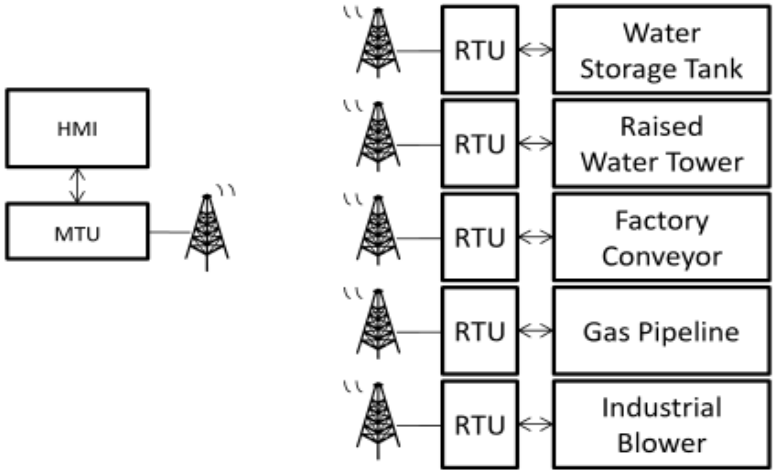

Figure 2 Serial communication based control systems

Figure 2 shows a high level schematic of the five serial port control systems. A single HMI is used to control all five control systems. The control systems may be operated individually or simultaneously in any grouping.

The HMI used with the serial port control systems is GE/Fanuc iFix. The HMI provides an interface for an operator to monitor and control the water storage tank system. The GE/Fanuc iFix HMI supports three communication protocol MODBUS ASCII, MODBUS RTU, and DNP3. All three communication protocols are primarily command response based protocol. A master node, in this case the HMI, sends commands to slave nodes, the individual RTU, which execute the command and then provide a response. Commands include requests for information such as reading values stored in system registers and required changes to system state (via changing system set point registers).

The HMI forwards MODBUS commands to the MTU which in turn forwards commands to the RTU. The MTU is configured as a repeater. The MTU includes two EIA-232 UARTs. The first UART is connected by serial port cable to the HMI host. The second serial port UART is connected to an industrial900 MHz radio. Commands from the HMI are received on the HMI port and forwarded to the radio port. The industrial900 MHz radio is also a repeater which wirelessly broadcasts commands and responses to other radios in the network(there is one radio for each RTU). Responses from RTU are handled in a similar manner except information flows in the opposite direction.

The MTU and RTU are identical Control Microsystems, Inc. SCADAPack LP PLC. Each PLC is controlled by firmware. Firmware may be written as Ladder logic, in ANSI C, or maybe a combination of both. As mentioned above, the MTU PLC is configured as a repeater; the MTU copies commands and responses received from the HMI port to the radio port or vice versa. Each RTU PLC contains custom ladder logic to control an individual physical process.

The RTU ladder logic for the five serial port control systems use a common configuration. The RTU includes input registers, also known as setpoint registers. The HMI software makes changes to setpoint register values to control the physical process. Common setpoint register types include, mode settings, actuator (valve, breaker, and switch) settings, and process parameter settings (maximum and minimum valves)for controlled process parameters, PID settings, etc. RTU also include output registers. Output registers contain measured values from the physical process and state information for actuators in the control system. Output registers may be connected to analog inputs to the RTU, digital inputs to the RTU, or be driven by ladder logic or C firmware.

Water Storage Tank Control System

The water storage tank control system models oil storage tanks found in the petrochemical industry. Petro chemical refineries use oil and oil byproducts to produce gasoline, kerosene, diesel, and many types of plastics. In a common configuration, oil arrives by sea and is pumped into oil storage tanks to provide a consistent supply of oil to the refinery operation. Oil storage tank control systems similar to the one modeled by the water tank control system are used to monitor oil inventory and distribute oil to refinery processes. Water was substituted for oil for safety reasons.

The water storage tank control system contains primary storage tank and secondary water storage, a pump to move water from the secondary tank to the primary tank, a gravity fed manual relieve valve which allows water to flow from the primary to secondary tank, and a sensor which provides the water level in the primary tank as a percentage of total capacity. The secondary tank is not a feature of an industrial oil storage tank. The secondary tank is used to provide a destination for water when it leaves

the primary tank and a source for water to fill the primary tank. The water tank control system is a closed loop.

The water tank RTU ladder logic includes 6 setpoint registers; HH, HI, LO, and LL water level setpoint register, a pump override setpoint register, and a mode setpoint register. The RTU ladder logic also includes 3 output registers which store process parameters; pump state, water level, and alarms tate.

An operator uses the HMI to monitor and remotely control the water storage tank. The operator can place the system in automatic or manual mode. In both modes, the HMI polls the RTU every two second. Each poll is performed by a MODBUS command being sent from the HMI to the RTU. The MODBUS command requests to read the alarm state, the pump state, and the water level. The same MODBUS command includes values for the 6 setpoints, HH, HI, LO, LL, pump override, and mode. Each time a command is received from the HMI, the RTU updates all setpoints with values from the command and responds to the read request. A single RTU response returns the alarm state, pump state, and water level. If the communication link between the MTU and RTU is broken the RTU will continue operating with the last configured setpoints.

The water storage tank includes manual and automatic control modes. In automatic mode, the RTU ladder logic program attempts to maintain water level between the Land H setpoints using an ON/OFF controller technique. When the RTU ladder logic program detects that the water level has reached the L level it turns on the water pump. When RTU ladder logic program senses that the water level has reached the H level it turns off the water pump. If the manual relief valve is open the water level in the tank will oscillate between the H and L setpoints continuously. If the relief valve is closed the

water level depends on the pump state at the time of closing. If the pump is on when the relief valve is closed, the water level will rise to the H setpoint and the pump will turn off. The water level will remain constant until the relief valve is re-opened. If the pump is off when the relief valve is closed, the water level will remain constant until the relief valve is re-opened. If, due to a system fault, the water level rises to the HH setpoint or falls to the LL setpoint an alarm is triggered. The alarm sounds at the water storage tank. In manual mode, the pump state is controlled manually by the HMI. An operator can manually activate or deactivate the pump and manually activate. In manual mode the HMI continues to poll the RTU to read process settings and conditions every second. If due operator error, the water level rises to the HH setpoint or falls to the LL setpoint the alarm is triggered.

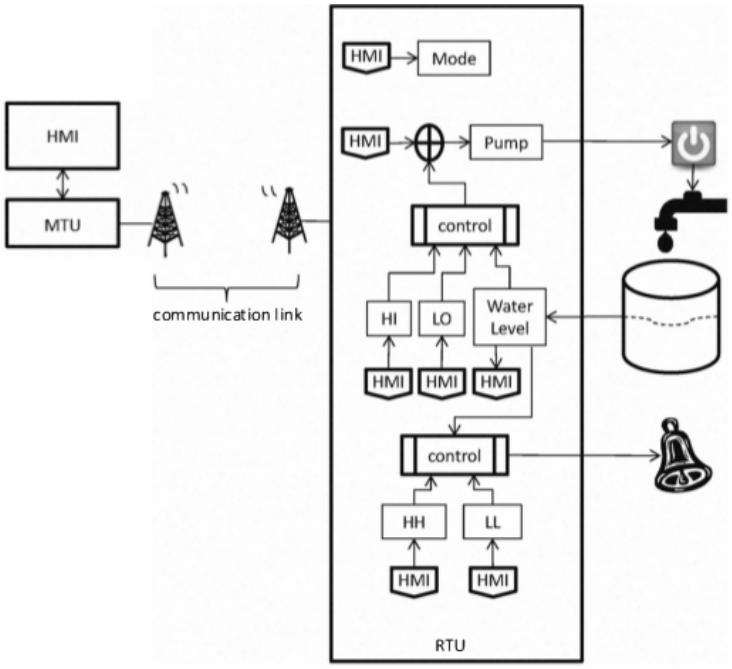

Figure 3 Water Tank Control System Schematic

Gas Pipeline Control System

The gas pipeline control system models a gas pipeline used to move natural gas or other petroleum products to market. Cyber penetration of control systems monitoring and controlling a gas pipeline control systems can lead to loss of the visibility and loss of control of the gas pipeline.

The gas pipeline control system contains a closed loop gas pipeline connected to an air pump which pumps air into the pipeline. A manual release valve and a solenoid release valve are available to release air pressure from the pipeline. A pressure sensor is attached to the pipeline which allows pressure visibility at the pipeline and remotely on an HMI screen.

Ladder logic is used to program the gas pipeline MTU and RTU. The MTU ladder logic forwards commands and responses to the RTU and HMI respectively. The RTU ladder logic includes ten setpoint registers; a system mode setpoint register, a control scheme setpoint register, a relief valve open/close setpoint register, a pump on/off setpoint register, a pressure setpoint register, a proportional integral derivative(PID) gain setpoint register, a PID reset setpoint register, a PID rate setpoint register, a PID dead band setpoint register, and a PID cycle time setpoint register. The system mode setpoint register allows an operator to place the system into manual, automatic, or off states. The control scheme setpoint allows an operator to select between pressure control using a pump on/off scheme or a relief valve open/close scheme. The relief valve open/close setpoint allows an operator to manually control the relief valve. The pump on/off setpoint allows anoperator to manually control the pump. The pressure setpoint provides a target

gas pressure in pounds per square inch(PSI). The PID gain, reset, rate, dead band, and cycle time setpoints configure the system using a PID feedback control mechanism.

The RTU ladder logic also includes three output registers which store process parameters; pressure measured in PSI, pump state, and relief valve state. The gas pipeline control scheme includes automatic and manual modes. In automatic mode, the RTU ladder logic program attempts to maintain gas pressure at the pressure setpoint using a PID control scheme. The control variable is either the state of the relief valve or the state of the pump depending upon the control scheme setpoint. In manual mode, the gas pipeline pump and relief valve are controlled manually through the HMI by an operator.

Ethernet Based Control System

Many modern control systems use Ethernet based communications infrastructures. The trend of using Ethernet based communications for control systems has led to increased cyber security awareness and research and development in the control system domain. The Ethernet network based testbed includes a chemical processing system in the MSU SCADA security laboratory.

In a chemical processing factory, in order to follow a chemical mixing recipe, one or more materials need to be sent to a reactor under certain conditions such as temperature and pressure. The chemical processing control system testbed in the lab models a chemical reactor in chemical plant. Water was substituted for chemical materials for safety reasons. The structure of the testbed is illustrated in Figure 4.

The testbed contains a reverse tank, reactor tank A and reactor tank B, a water level sensor in each tank to measure the percentage of the water in the tank, valves controlled by the PID controller and flow meters in each water pipe, and emergency

valves. Water in tank A and tank B simulate two types of chemical materials in a reactor. In a common configuration, chemical materials will be sent to the reactor in a certain percentage. For instance, material A will be 25% and material B will be 75% during the process. The water in tank A and tank B reaches the respective setpoints to simulate two materials reaching their percentage requirement.

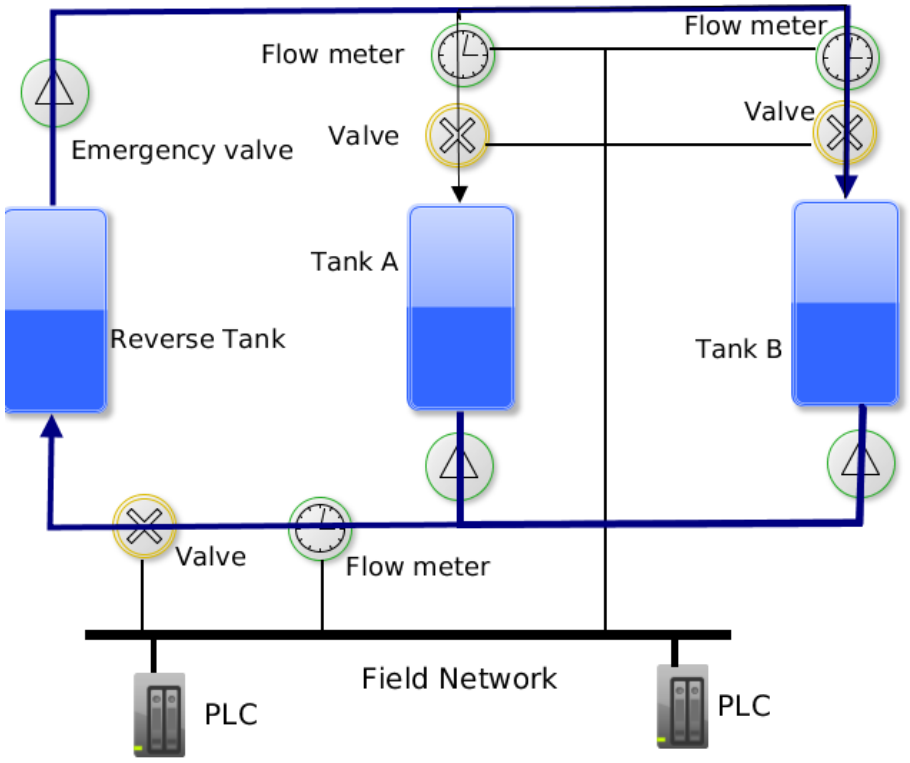

Figure 4 Structure of Chemical Processing System Testbed

The controller ladder logic in the chemical mixing testbed includes a water level setpoint register, water level register, pump state register, flow speed register and alarm state register. The PID control scheme is used to control of tank A and tank B. The PID

controller attempts to maintain water level at the water level setpoint. When the controller detects the water level is close to the setpoint, it causes an offtake valve to reduce the flow speed. The reverse tank's function is to provide water to tank A and tank B. The pump in the reverse tank pumps water to tank A and tank B. Another PID controller maintains the reverse tank water level. When water in tanks A and B reach their respective setpoints, the controller will adjust valves to ensure the water flow speed into tank A and B equals the water flow speed into the reverse tank.

Figure 5 shows the network topology of the chemical processing system testbed. An operator uses Human Machine Interface (HMI) software to monitor and remotely control the testbed. The protocol between HMI PC and Master is Common Industrial Protocol (CIP). The master exchanges information with PLCs though Siemens switch. Note: In the figure, SE-Switch stands for Siemens switch. In order to add device and protocol diversity to testbed two different brands of PLCs and protocols are implemented in this testbed.

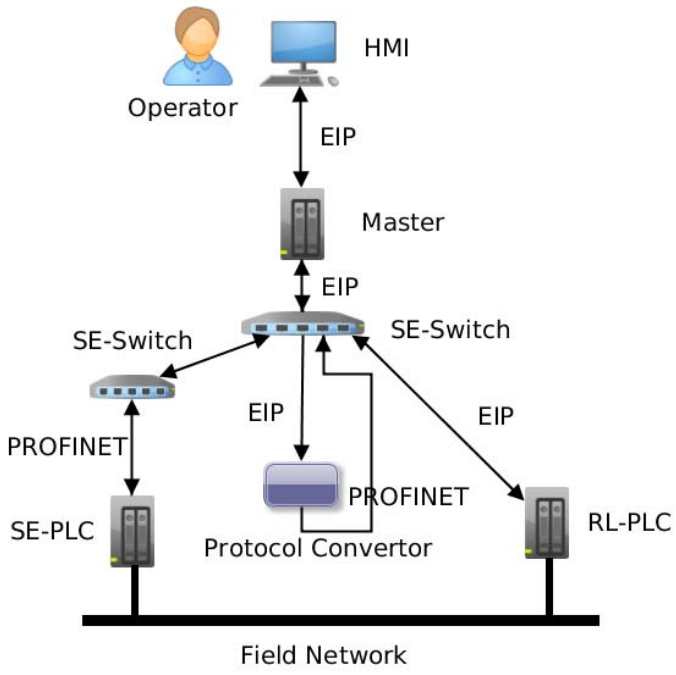

Figure 5 Network Topology of Chemical Processing System Testbed

The PLC in the left of Figure 5 is a Siemens Simatic S7 300 PLC which is widely used in process control systems. The Siemens Simatic S7 300 PLC uses the PROFINET protocol, which is based on industrial Ethernet. There are over three million PROFINET devices are used in the industrial area [1]. The switch and master support the EtherNet/IP protocol. In order to implement two different protocols in one system, a protocol convertor is added. The function of the protocol convertor is to allow the Siemens Simatic S7 300 PLC to communicate with the switch and the other devices in the system.

The PLC on the right of the Figure 5 is an Allen-Bradley Micrologix 1400 PLC (In the figure, RL-Switch stands for Allen-Bradley Micrologix 1400 PLC), which uses the EtherNet/IP protocol. The Industrial Ethernet Protocol (Ethernet/IP or EIP) is an open industrial network standard that applies the CIP protocol by encoding CIP messages

in Ethernet frames. EIP is built on the standard TCP/IP protocols. EIP was developed by Rockwell Automation and is now managed by the Open DeviceNet Vendors Association (ODVA).

Attacks on Serial Communication Based Control Systems

In this section a set of cyber-attacks against industrial control systems (ICS) are grouped into four attack classes; reconnaissance, response injection, command injection, and denial of service. The response injection class is further divided into naïve malicious response injection (NMRI) and complex malicious response injection (CMRI) attacks. The command injection class is further divided into malicious state command injection (MSCI), malicious parameter command injection (MPCI), and malicious function code command injection (MFCI) attacks. Table 3 lists 28 individual attacks described in this paper and lists each attacks sub-class. This section defines the four classes and their sub-classes and provides detailed descriptions of each attack listed in Table 3.

Table 3 List of Attacks against MODBUS Industrial Control Systems

Attack Index	Name	Classification
1	Address Scan	Reconnaissance
2	Function Code Scan	Reconnaissance
3	Device Identification	Reconnaissance
4	Points Scan	Reconnaissance
5	Memory Dump	Reconnaissance
6	Naïve Read Payload Injection	NMRI
7	Invalid Read Payload Size	NMRI
8	Naïve False Error Response	NMRI
9	Negative Sensor Measurement(s)	NMRI
10	Sensor Measurement Grossly Out of Bounds	NMRI
11	Sporadic Sensor Measurement Injection	NMRI
12	Random Sensor Measurement Injection	NMRI
13	Constant Sensor Measurement Injection	CMRI
14	Slope Sensor Measurement Injection	CMRI
15	High Slope Measurement Injection	CMRI
16	Low Slope Measurement Injection	CMRI
17	Replayed Measurement Injection	CMRI
18	Altered System Control Scheme	MSCI
19	Altered Actuator State	MSCI
20	Continually Altered Actuator State	MSCI
21	Altered Proportional Integral Derivative Parameter(s)	MPCI
22	Altered Control Set Point	MPCI
23	Force Listen Only Mode	MFCI
24	Restart Communication	MFCI
25	Clear Data Log	MFCI
26	Change ASCII Input Delimiter	MFCI
27	Invalid Cyclic Redundancy Code (CRC)	DOS
28	MODBUS Slave Traffic Jamming	DOS

Reconnaissance Attack

Reconnaissance attacks gather control system network information, map the network architecture, and identify the device characteristics such as manufacturer, model number, supported network protocols, system address, and system memory map. This section describes four reconnaissance attacks against MODBUS servers; the address scan, the function code scan, the device identification attack, and the points scan. The

address scan discovers ICS servers connected to a network. The function code scan identifies supported network operations which can be performed for an identified server. The device identification attack allows an attacker to learn a discovered device's vendor name, product code, major and minor revision, et cetera. The points scan allows the attacker to build a device memory map.

Attacks described in this section were implemented against MODBUS servers because MODBUS is an open standard and is a popular network protocol used for ICS devices. While these attacks are known to function against MODBUS servers the vulnerabilities the attacks exploit are general enough to also likely be found in other network protocols used by ICS.

Industrial control system users often develop standard hardware, software and control scheme parameter configurations which are duplicated throughout a control system. For example, an electric transmission system may use a standard panel which includes the same protective relays used at many substations throughout a single company's transmission system. A second example is pump stations for a gas pipeline. Pump stations are distributed along the gas pipeline to ensure product flow. Programmable logic controllers (PLC) and the program on those controllers will be similar throughout the system.

The combined results of the address scan function code scan, the device identification attack, and the points scan can be used to generate a signature for MODBUS servers common to a particular company, use case, or vendor. This signature can then be used to build maps of discovered systems by company, by use case, or by

vendor. Such signatures can also be used to build a database of vulnerabilities and exploits for each aforementioned category.

Attack 1 is the *Address Scan*. MODBUS servers use either an IP address for MODBUS/TCP systems or a single byte address for MODBUS RTU and ASCII systems. Attackers can perform an address scan to identify MODBUS server addresses which are in use. Each MODBUS server is assigned a unique address. MODBUS systems typically have a static configuration in which the number of servers does not change and the address assignment of the individual clients does not change. MODBUS/TCP servers can have any legal IP address. MODBUS/TCP servers listen on TCP port 502.Statement 1 defines the legal address range for MODBUS RTU and ASCII systems.

$$ADDR \in \{0,..,247\} \quad (1)$$

The MODBUS protocol requires addressed servers to return a response code after being addressed by a query. The response may be acknowledgement of a successful transaction or indicate an error message. No response will be received for MODBUS queries addressed to nonexistent servers. To identify MODBUS servers an attacker can send MODBUS queries to each legal MODBUS address and wait for any response. Note, for MODBUS RTU and ASCII systems the 0 address is for broadcast commands. No response is sent for broadcast commands and therefore this address would typically not be used for address scan attacks.

Attack 2 is the Function *Code Scan* attack. After MODBUS server addresses are identified an attacker may wish to scan servers to identify supported function codes. The MODBUS function code field is a single byte. MODBUS specifications define four types

of function codes; public function codes, user defined function codes, reserved function codes, and error function codes. Statement 2 lists the set of public function codes.

$$PFC \in \{1,2,3,4,5,6,7,8,11,12,15,16,17,20,21,22,23,24,43\},43\} \quad (2)$$

User defined function codes must be in the range defined by statement 3.

$$UFC \in \{65, \ldots, 72,100, \ldots 110\} \quad (3)$$

Reserved function codes are codes in the public code space which have been used by legacy devices and which are not supported as public codes. This set is most often empty. Reserved function codes are in the range defined by statement4.

$$RFC \in \{65535, 9, 10, 13, 14, 41, 42, 90, 91, 125, 126,127,127\} \quad (4)$$

When a MODBUS query generates an error at the MODBUS server an error function code is returned in the response. The error function code is the query function code + 0x80. An error function code exists for all legal function codes from statement 4 regardless of whether the underlying public, user defined, or reserved function code is supported by the MODBUS server. Therefore MODBUS function code scans should not scan the error function codes.

An attacker can perform a MODBUS function code scan by sending a query to all function codes in the sets defined by statement 4. MODBUS query payloads vary by function code. However, an attacker need not form a proper payload for each function code to determine if a function code is supported by a MODBUS server. Function code scans can be grouped into two categories by the function code scan attack. If the function

code is not supported an exception code 1 (invalid function code) response will be returned. All other responses, whether indicating an error or transaction success, indicate the function code is supported by the targeted server.

Attack 3 is the *Device* Identification attack. Attackers can also fingerprint remote devices to learn specific information such as the vendor name, product code, and revision number. This information can be used to search for known vulnerabilities in exploit databases such as Exploit Database (EDB) [ci] or to download exploit scripts targeting to the identified and fingerprinted system.

MODBUS servers may implement a function code to allow a client to read device identification information. For MODBUS RTU and ASCII servers function code 0x11 allows an attacker to retrieve the current run status and additional information which is device specific. Device specific contents may include sensitive information.

MODBUS servers implement second read device identification function code, 0x2B. There are three Read Device ID object types: basic, regular, and extended information. Basic information is mandatory for all MODBUS servers and includes the vendor name, the product code, and the major and minor revision. The regular information is optional and includes the vendor uniform resource locator (URL), the product name, the model name, and the user application name. The extended information is optional and includes user defined objects.

Attach 4 is the *Point Scan* attack. In industrial automation programming points are objects used to store programming variables or input values tied to sensors or output values tied to actuators. Points can be accessed from within a program running on a programmable logic controller (PLC) or programmable automation controller (PAC).

Points are also available for read and write access via network interfaces attached to PLC and PAC. When the PLC or PAC is networked it acts as a server. A point scan is used to identify implemented points within such a server. A point scan may also be used to read the contents of implemented points to gain system intelligence.

For MODBUS servers, points are grouped into data blocks called coils, discrete inputs, holding registers, and input registers. Coils and discrete inputs represent a single Boolean bit, however, when accessed the legal values are 0x00 and 0xFF. Holding and input registers are 16-bit words. Coils are read and write capable and discrete inputs are only read capable. Holding registers are read and write capable and input registers are only read capable. Each data block may have its own set of contiguous address space or the data blacks may share a common memory address space. This choice is vendor and device specific.

Points scans should are performed after a function code scan to ensure that function codes used by the points scan are valid. MODBUS points scans can be performed with the following function codes; read coils (0x01), read discrete inputs (0x02), read holding registers (0x3), read input registers (0x4), write single coil (0x05), write single register (0x06), write multiple coils (0xF), write multiple registers (0x10), and read/write multiple registers (0x17). A point scan for each of the above function codes will be similar. Each provides different information.

A point scan first attempts to read from each legal location of each type of data block and then attempts to write to each legal location of each type of data block. To maximize the specificity of point scan results the smallest quantity of data should be used for both the read and write scans. During the read scan two results are possible, assuming

the function code has been pre-validated by a function code scan. First, exception code 1 indicates the address is invalid. This means this address is not valid for that type of data block. Second, the read command is valid and a MODBUS packet with read results is returned. During a write scan three results are possible, assuming the function code has been pre-validated by a function code scan. First, exception code 1 indicates the address is invalid. This should not occur if a read scan is performed first and invalid read addresses are removed from the write scan. Second, exception code 3 indicates the address is legal but the location is read only. This can occur if the write coil function is attempting to write to a coil or if the write register function is attempting to write to an input register. By performing read scans for each data block type followed by write scans for each data block type an attacker can learn, for each legal address; whether the address is available for use, is the address read only or read write capable, which functions can be used to access the address.

In addition to creating a memory map for the MODBUS server, a read scan can be used to make a copy of the devices memory contents. This memory image can be useful in planning subsequent attacks.

Attack 5 is the *Memory Dump* attack. As mentioned above, points are used to store programming variables and input values tied to actuators or output values read from sensors. For many PLCs all points are readable via the network communications interface. However, only points intended to store input values from an external source or output values read by an external device should typically be read over the network. A memory dump attack reads points associated with internal programming variables which are not intended to be read over the network. A memory dump attack may be intentional.

In this case an attacker may use a memory dump attack to gather critical information about a system in preparation for a later attack. A memory dump attack may also be an unintended result of a points scan.

Malicious Response Injection Attack

Industrial control systems commonly use polling techniques to continuously monitor the state of a remote process. Polling takes the form of a query transmitted from the client to the server followed by a response packet transmitted from the server to the client. The state information is used to provide a human machine interface to monitor the process, to store process measurements in historians, and as part of feedback control loops which measure process parameters and take requisite control actions based upon process state.

Many industrial control system network protocols lack authentication features to validate the origin of packets. This enables attackers to capture, modify, and forward response packets which contain sensor reading values. Industrial control system protocols also often take the first response packet to a query and reject subsequent responses as erroneous. This enables attackers to craft response packets and use timing attacks to inject the responses into a network when they are expected by a client.

Response injection attacks take three forms. First, response injection attacks can originate from control of a programmable logic controller or remote terminal unit, network endpoints which are the servers which respond to queries from network clients. Second, response injection attacks can capture network packets and alter contents during transmission from server to client. Finally, response injections may be crafted and transmitted by a third party device in the network. In this case, the response there may be

multiple responses to a client query and the invalid response may assume prominence due to exploiting a race condition or due to secondary attack such as a denial of service attack which stops the true server from responding.

In this section multiple response injection attacks are discussed. The response injection attacks are grouped into two categories; 1) Naive Malicious Response Injection (NMRI) attacks and 2) Complex Malicious Response Injection (CMRI) attacks.

Naive Malicious Response Injection (NMRI): Naive Malicious Response Injection (NMRI) attacks lack sophistication. NMRI attacks leverage the ability to inject response packets into the network but lack information about the process being monitored and controlled. NMRI attacks may send invalid payloads. For example, an attacker may know have performed a set of reconnaissance attacks to learn system addresses, function codes, and memory map, but lack specific details on what the monitored process is or lack details on valid data contents for each point found on a server. In this case, the attacker may craft a response injection attack with a payload of all zeroes, all negative numbers, all very large numbers, or other likely invalid contents. Alternatively, NMRI attacks may be based on limited process information. For example, an attacker may know process details such as process limits or valid contents for each point found on a server but not have the capability to craft more sophisticated attacks. For example, an attacker may be able to cause an alarm. For this work, four NMRI attacks were developed.

Attack 6, *Naïve Read Payload Size*, is the first NMRI attack is based only on network protocol knowledge. MODBUS read coil, discrete input, holding register, and input register queries include a quantity field to specify the number of objects to be returned by the server. An NMRI attack can craft malicious responses which include the

correct quantity of returned objects which are all zeroes or all ones. Alternatively, the NMRI can return the correct number of requested objects with random contents. Random contents is particularly interesting for the read coils and discrete inputs cases since these returned values are specified to be limited to only 0x00 and 0xFF for each coil or discrete input.

Attack 7, Invalid Read Payload Size, is an NMRI attack based only in which the requested number of objects from the read coils, discrete input, holding register, or input register query is ignored. The response payload is either larger or smaller than the requested amount. The response payload may be formed by trimming or extending a valid payload, or by creating a payload with zeroes, ones, or random bytes.

Attack 8, Naïve False Error Response, is an NMRI attack in which falsified error responses are returned to the client after a read command. For MODBUS an error packet is formed by adding 0x80 to the function code followed by an exception code. Read command exception codes are limited to 0x01, 0x02, 0x03, and 0x04. This NMRI attack can send random exception codes which fall in the legal range or send random exception codes which are outside the legal range.

Attack 9, Negative Sensor Measurements, is an NMRI attack which consists of injecting negative process measurements. This attack was implemented for two laboratory scale process control systems. First, for a gas pipeline system which measures the gas pressure in a pipe, the attack sent negative pressure readings. Second, for a water tank control system which measures the water level in a tank, the attack sent negative water level readings. In both cases the attack requires prior knowledge of the address of the gas pressure or water level field. The attack is also requires prior knowledge that the

gas pressure or water level fields are floating point values. In normal operation, neither gas pipeline pressure nor the water tank water levels take values less than zero.

Attack 10, Sensor Measurements Grossly Out-Of-Bounds, is an NMRI attack which injects process measurements significantly outside the bounds of alarm set points. For example, both the aforementioned process control systems include alarm set points for gas pressure and water level respectively. These alarm level set points are named "High High" (HH), the high level alarm, and "Low Low" (LL), the low level alarm. Results of an attack which sends very low water level measurements are graphed in Figure 6. In this graph, most measurements are between the low (L) and high (H) system set points. The water tank control logic turns on and off a pump to keep the water level between the L and H set points. The values labeled NMRI are falsified measurements significantly below the LL set point. This is a naïve attack because the attacker may not know the HH and LL set points but may simply send very large or very small values in expectation of an alarm. It is also naïve because the falsified measurements are also sporadic, where an actual system could not change water levels at the rates indicated by the graph in Figure 6.

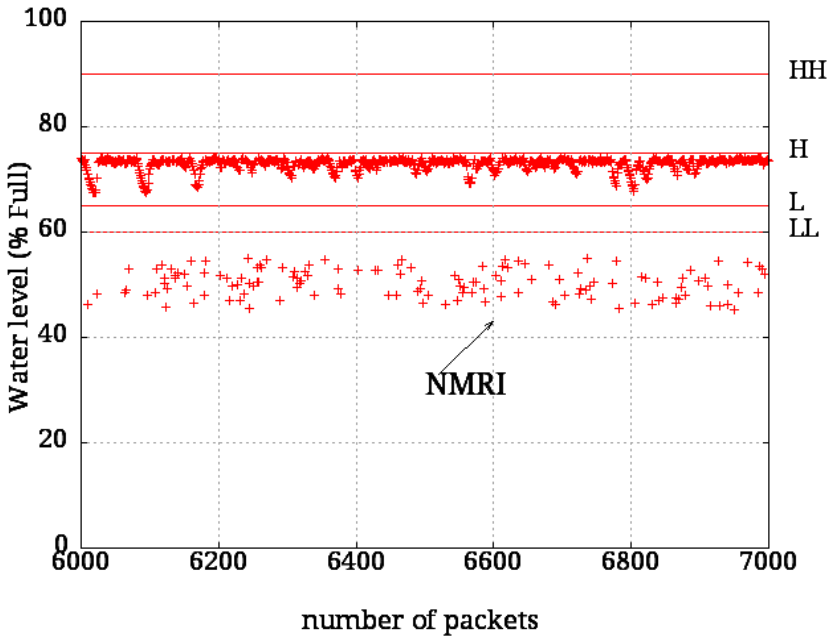

Figure 6 NMRI Attack: Sporadic Low Level Measurements

Attack 11, Sporadic Sensor Measurement Injection, is an NMRI attack which sends false process measurements outside the bounds of H and L control set points while not outside the alarm set point range formed by HH and LL. For the attack used for this work falsified water levels are sporadically sent to the MODBUS client. Both the water tank and gas pipeline systems regularly have measurements outside the H and L limits due to a time delay between measuring the gas pressure or water tank level and sending the command to turn off the pump adding water or gas to the physical process. This makes developing an automated intrusion detection rule based strictly on the H and L limits difficult. However, this NRMI attack differs from the ordinary out of bounds situation in that the response injection packets are sporadic in nature.

Attack 12, Random Sensor Measurement Injection, is an NMRI attack which sends random process measurements for the gas pipeline pressure and water tank water

level. Since these measurement values are random some falsified measurements are within process limits (LL/HH and L/H limits) and some are not.

Complex Malicious Response Injection (CMRI) attacks add a level of sophistication above that of the NMRI attacks. CMRI require understanding of the cyber physical system being attacked. CMRI attacks attempt to mask the real state of the physical process being controlled to negatively affect the feedback control loop managing the cyber physical system. In this section five CMRI attack scenarios are discussed.

Attack 13, Constant Sensor Measurement *Injection*, is a CMRI attack in which the attacker sends malicious packets containing the same measurement over a period of time in order to mask the real state of the system. This attack will appear to the operator as a constant level for a measured parameter. This attack reflects an incorrect state of the system and may lead an operator or automated control algorithm to take incorrect control actions. For example, in the water tank control system if a falsified water level is lower than the L set point, an operator may attempt to use a manual control mode to increase the water level when in actuality is already correct. This may lead to overflow in the water tank. This attack can have similar impact on the gas pipeline control system. A false low pressure response injected maliciously may lead an operator to attempt to raise the pressure and result in physical harm to the pipeline. Alternatively, a false high pressure response injected maliciously may lead an operator to attempt to lower the pressure by turning off a pump or opening a relief valve. This could lead to loss of flow in the pipeline and a resulting loss of gas distribution service.

Attack 14, Calculated Sensor Measurement Injection, is a CMRI attack in which calculated process measurements are injected. This attack simulates a process

measurement trend such as a water level or gas pressure increasing or decreasing. For example, an attack can inject falsified response packets which simulate a water level trend increasing from a normal level such as 20% to 100%. Such an attack would cause the operator to turn off the water pump while the actual water level is 20% full. This attack requires system knowledge and an accurate model of the system being attacked.

Attack 15, Replayed Measurement Injection, is a CMRI attack in which means the attacker replays captured process measurements to the client to give the operator the impression the system running normally.

Attacks 13-15, are designed to appear like normal process functionality. These attacks can be used to mask other process changes such changes to process state through malicious command injection attacks. Because these attacks project a state of normalcy they are very difficult to detect.

Attack 16, High Frequency Measurement Injection, is a CMRI attack in which the frequency of process measurement changes is increased beyond a normal rate. For example, the falsified responses may indicate a fast rising water level or fast decreasing gas pressure. This attack scenario may appear to match the system behavior common at a different time of a day and may cause an operator to misconfigure the system to handle the falsified demand. However, in the case of water or gas distribution, increased pump speed may lead to overflow since demand is actually lower. Figure 7 shows a graph of changing oil storage tank level measurements before and during a High Frequency Measurement Injection Attack. In the figure, the frequency of liquid level changes is normal at first (the left side of the graph) and then during the attack (right side of the

graph) the liquid level rises and falls more rapidly. Such a change may simulate a period of high demand.

Attack 17, Low Frequency Measurement Injection, is a CMRI attack in which the frequency of process measurement changes is decreased below a normal rate. For example, the falsified responses may indicate a slowly rising water level or slowly decreasing gas pressure. This attack is similar to the high frequency CRMI attack except the low frequency case simulates a period of low usage.

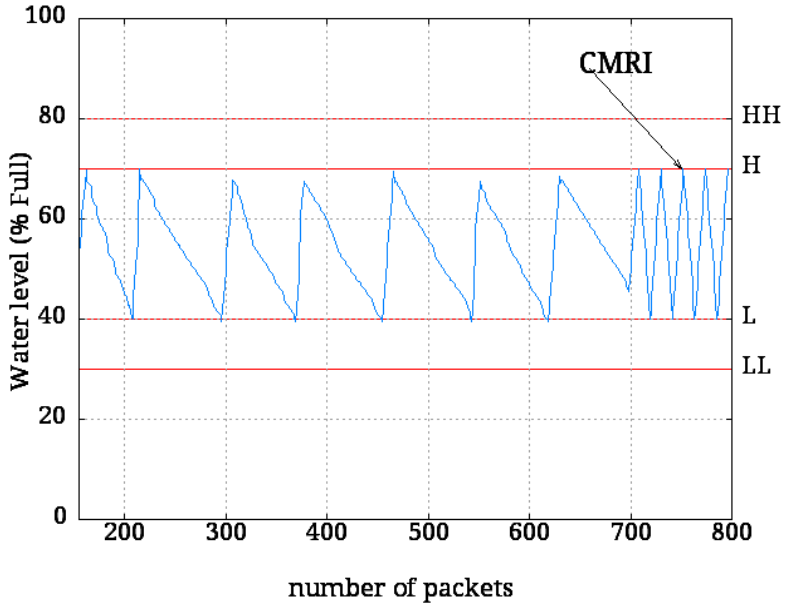

Figure 7 CMRI on Water Tank Storage Control System

Malicious Command Injection Attack

Command injection attacks inject false control and configuration commands into a control system. Human operators oversee control systems and occasionally intercede with supervisory control actions. Hackers may attempt to inject false supervisory control

actions into a control system network. Remote terminals and intelligent electronic devices are generally programmed to automatically monitor and control the physical process directly at a remote site. This programming takes the form of ladder logic, C code, and registers which hold key control parameters such as high and low limits gating process control actions. Hackers can use command injection attacks to overwrite ladder logic, C code, and remote terminal register settings.

The potential impacts of malicious command injections include interruption process control, interruption of device communications, unauthorized modification of device configurations, and unauthorized modification of process set points.

As mentioned in the response injection discussion above much industrial control system network protocols lack authentication features to validate the origin of packets. This enables attackers to capture and alter command packets. Additionally, attackers can craft original command packets and directly inject them into the control system network.

In this section multiple command injection attacks are discussed. The command injection attacks are grouped into three categories; 1) Malicious State Command Injection (MSCI) attacks 2) Malicious Parameter Command Injection (MPCI) and 3)Malicious Function Code Injection (MFCI).

Malicious State Command Injection (MSCI) MSCI attacks change the state of the process control system abnormally to drive the system from a safe state to a critical state by sending malicious commands to remote field devices. MSCI attacks may require a single injected command or multiple injected commands.

Typically actuators, such as switches or valves, connected to physical processes are connected to a digital or analog output connected to a remote terminal unit (RTU) or

intelligent electronic device (IED). Each output connects to the cyber system by modeling it as a digital point in a register. Changing the state of a bit or bits in such a register has an immediate impact on the physical actuator. For example, a pump may have an ON/OFF mechanism which is changed by writing a value to a bit a in a register on a remoter terminal unit (RTU). Such registers can be manipulated by network protocol write commands. For example, the MODBUS protocol includes write coil and write register commands. An attacker who understands a device's implementation specifics including a memory map can craft a command to alter actuator states. Two examples of MSCI attacks are provided.

Attack 18, Altered System Control Scheme, is a MSCI attack which changes the control state of a gas pipeline control system from automatic mode to manual mode and then turns on a pump which increases the pressure within the pipe. In this laboratory scale control system programmable logic controller (PLC) with MODBUS-RTU server is connected to a pump, a solenoid, and a pressure meter. In automatic mode the user sets a target pressure and the PLC switches the pump between ON/OFF mode, and switches the solenoid between OPEN/CLOSED modes to fire the solenoid to open and close the relief valve which in turn controls the pressure in the pipe. In manual mode the pump state and solenoid states are no longer controller by the PLC program and become directly controlled by register values store in the PLC. MODBUS commands can be used to change the values stored for system control mode, pump state, and solenoid state. To implement the attack first a write register (MODBUS function code 03) command to address 0xABCD is used to switch the control mode to manual. Next, a write register command to address 0xABCD is used to turn on the pump. For the gas pipeline control

system a pipe pressure above 60 PSI is considered a critical state. Pressure above this value will potentially damage system components. Placing the system in manual mode and turning on the pump causes the pressure to climb toward this critical value. An operator monitoring the system with a human machine interface should notice the climbing pressure and can take control to correct the issue. Additionally, the rising gas pressure may trigger a process alarm to gain an operators attention.

Attack 19, Altered Actuator State, is an MSCI attack scenario which changes system actuator states one time. For the gas pipeline system Altered Actuator State attacks include command injections which turn the pump on or off and command injections which open or close the relief valve. For the oil storage tank system an Altered Actuator State attack was implemented to turn the pump on or off.

Attack 20, Continually Altered Actuator State, is an MSCI attack scenario which changes system actuator states repeatedly to attempt to damage the actuator devices or other system components. Two examples are provided for this paper. First, command packets are continually transmitted to switch the state of the gas pipeline pump. This is achieved by sending MODBUS write register commands continually inverting the bit controlling the pump ON/OFF state. The second attack repeatedly transmits MODBUS write register commands to invert the state of the solenoid controlling the relief valve. In both cases repeatedly changing the state can lead to ware on the system and cause component failures. This attack also can lead physical process into unknown and potentially unsafe states. An operator monitoring the system with an HMI can observe this attack in progress by the continuous state changes. However, since the attack relies on a stream of commands changing the state of the pump or solenoid an operator will not

be able to simple change correct the system state to fix the problem since a new attack command will immediately follow and again alter the system state. As such preventing such an attack will require stopping the attack at its source.

Malicious Parameter Command Injection (MPCI) Industrial control systems often include tight control loops between the physical process, sensors and actuators, and a programmable logic controller (PLC) directly connected to the sensors and actuators. The PLC will include a program which monitors process sensors, computes a control action, and then executes control actions via manipulating process actuators. Typically, this control loop is setup as a distributed control system which can function without a network connection to supervisory and monitoring systems. Set points are values written into PLC registers which direct the PLC program. Examples of set points include high and low set points on the gas pipeline. In automatic mode the PLC opens the relief valve or turns of the pump, depending on another set point value, when the pressure reaches the high set point. Malicious Parameter Command Injection (MPCI) attacks alter PLC set points.

Attack 21, Altered Proportional Integral Derivative (PID) Parameter(s) changes. PID controllers are tuned by setting multiple gain parameters to properly control the system. Changing any of the PID parameters can cause the controller to perform incorrect control actions. Typically, a process engineer tunes the PID controller during system installation and then these parameters are left constant. However, the PID control parameters are typically implemented as registers on the PLC and can therefore be altered by a network command. The gas pipeline system uses a PID control scheme to manage the pressure in the pipeline. For the gas pipeline system all PID parameters are stored in

contiguous memory locations. A MODBUS write multiple registers command was used to overwrite all PID parameters with invalid values. For the attack used with this work all PID parameters were overwritten with zero values. This change has an immediate noticeable impact on the gas pipeline causing the pump to repeatedly turn on and off at a rate of approximately 3 times per second.

Attack 22, Altered Control Set Point, is an MPCI attack which changes device set points. Set points are typically used to provide variable control over a system. For example the oil storage tank system uses an ON/OFF control scheme to keep the amount of liquid in a tank between a low set point and a high set point. A level sensor continuously monitors liquid level as a percentage of tank full and turns a pump on an off to add liquid to the tank. A MODBUS write register command was used to change both the high and low set points. This attack also alter alarm values stored in PLC registers to disable alarms by changing set points liquid level alarms to values in line with the altered high and low set points.

Malicious Function Code Injection (MFCI) Application layer protocols sometimes include commands which have unintended consequences when used by attackers.

MODBUS function code 8, named "Diagnostics" includes three sub-function codes, commands, which can be used to disrupt the client server communication link. The original intent of the diagnostics function code was to provide a means to diagnose and address communication issues. The diagnostics command is only required for serial port MODBUS systems.

Attack 23, Force Listen Only Mode, causes a MODBUS server to no longer transmit on the network. The diagnostic function code, MODBUS function code 8, includes a sub function code to force a MODBUS server into listen only mode. Many industrial control systems use polling techniques in which the master node, such as human machine interface (HMI) software, polls the MODBUS servers periodically for data. The HMI displays data to human operators who may then take supervisory control actions based upon the current state of the system. There also exist wide area control schemes which poll MODBUS servers for data to support automated control actions. A MODBUS server which is placed in listen only mode by an attacker will not respond to queries and in the situations described will result in a loss of system visibility and control.

Attack 24, Restart Communication, sends a command which causes the MODBUS server to restart which leads to a temporary loss of communication. The diagnostic function code includes a sub function code to restart the remote device and cause it to execute its power up diagnostic tests. This loss of communication leads to a temporary inability to observe and control the process. During the restart period local control from the program running on the field device is also lost. Multiple successive restart communication attacks can lead to a near complete loss of communication with and control over the process.

Attack 25, Clear Communications *Event Log*, clears the MODBUS server's communications event log. The restart communication command also includes an option to clear a remote device's communication event log. This attack may use to erase evidence of a prior attack.

Attack 26, Change ASCII Input Delimiter, changes the delimiter used for MODBUS ASCII devices. MODBUS ASCII devices use ASCII characters as delimiters to identify the start and end of packets. The change ASCII input delimiter command can be used to change the ASCII character which denotes the end of a packet. A Change ASCII Input Delimiter Attack changes this delimiter without the knowledge of the MODBUS client which normally communicates with the MODBUS server. This attack requires the ability to penetrate the communication link between the client and server. Penetrating of such a serial link is possible when vulnerable wireless radios or other intermediary communication systems are used to connect the client and server.

Denial of Service Attack

Denial of Service (DOS) attacks against industrial control system attempt to stop the proper functioning of some portion of the cyber physical system to effectively disable the entire system. As such DOS attacks may target the cyber system or the physical system. DOS attacks against the cyber system target communication links or attempt to disable programs running on system endpoints which control the system, log data, and govern communications. DOS attacks against the physical system vary from the manual opening or closing of valves and switches to destruction of portions of the physical process which prevent operation. This work concentrates on DOS attacks against the communication system. Two DOS attacks were developed for this work.

MODBUS systems may be MODBUS TCP/IP, MODBUS RTU, or MODBUS ASCII. MODBUS TCP/IP is a routable protocol which allows other devices to initiate DOS attacks targeted to a victims IP address. MODBUS RTU and ASCII use RS-232 or RS-485 physical layers. These serial port protocols are considered non-routable.

However, MODBUS RTU and ASCII devices are vulnerable to certain DOS attacks. RS-232 is a point to point protocol used for MODBUS connections over short distances, typically less than 20 meters. However, control systems often connect a remoter terminal unit or master terminal unit to a wireless radio using RS-232 then use the radio to transmit across longer distances. Such radio links can be penetrated by attackers and can therefore be the source of a DOS attack.RS-485 serial links allow multipoint network topologies. In these cases a device on the network can become infected with malware and then the infected device can initiate a DOS attack against other devices on the network.

Traffic jamming is a class of DOS attacks in which high volumes of traffic are sent to a network endpoint. Attackers attempt to overwhelm the endpoint by either sending transmissions faster than they can be processed or by sending packets crafted to cause software errors which generate exceptions which crash the network stack, the running program, or the operating system of the targeted device.

Attack 27, Invalid Cyclic Redundancy Code (CRC), injects a large number of MODBUS packets with incorrect CRC. Packets with invalid CRC are rejected by both MODBUS servers and clients. The victim device must check the CRC of each packet. A flood of packets with invalid CRC scan overwhelm a device and cause it to crash or the flood may stop communication with other legitimate devices via loss of ability to transmit and/or receive packets.

Attack 28, MODBUS Master Traffic Jamming, is a traffic jamming attack in which a non-addressed slave transmits out of turn. MODBUS RTU and ASCII systems often are configured with a single master connected to multiple slaves. When there are multiple slaves only the addressed slave should respond to master queries. The *MODBUS*

Master Traffic Jamming Attack is a traffic jamming attack in which a non-addressed slave transmits out of turn. For RS-485 systems, in both the 2-wire and 4-wire cases, the slave transmit wire is shared by all slaves attached to the bus. In this case, a slave transmitting out of turn will cause a legitimate slave's transmission to be garbled and lost and result in a timeout and retransmission by the master. A *MODBUS Master Traffic Jamming Attack* against a RS-232 system with wireless radio between the master and slave node is described in [cii]. In this attack a wireless penetrator transmits continuously. The proprietary wireless radio includes a carrier sense back off arbitration scheme which causes legitimate slaves to continuously wait for a clear line to transmit. In laboratory experiments, attackers were able to force a legitimate slave to stay idle ad infinitum.

Figure 8 plots water level measurements observed by a HMI connected to the MODBUS master. The MI continuously queries the slave to read the water level. For this experiment the water storage tank was set to keep the water level in the tank between 40% and 70% full by cycling a water pump which fills the tank. The tank was configured to continuously drain water during the experiment. During normal operation the HMI sees the water level rises to the high set point when the pump is on and drops to the slow set point when the pump is off. Figure 8 shows the impact of the MODBUS Master Traffic Jamming Attack from the perspective of the HMI. When the attack starts the HMI no longer receives responses to its water level queries and therefore the water level in the plot no longer changes. This loss of process visibility can cause an operator or automated algorithm to misoperate the system. During this attack the master is also no able to transmit commands. As such the operator may notice that something is wrong but is not able to send commands to the remote system during the attack.

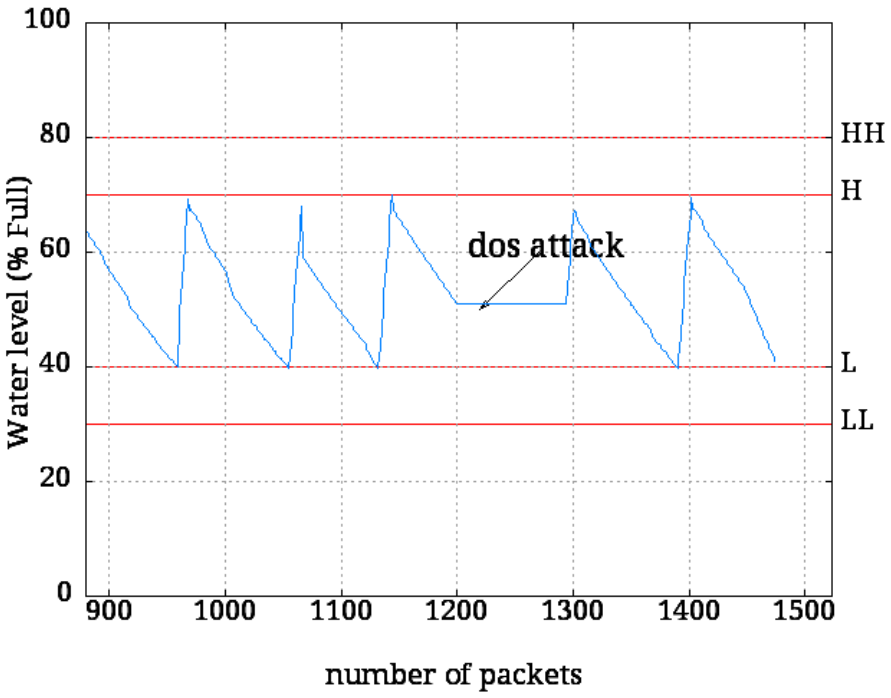

Figure 8 DOS Attack on Water Storage System

Attacks on Chemical Processing Control System

The EIP protocol was released in 2001, and it was designed for exchanging real time application messages in a process control system. EIP encodes the CIP message in TCP frames. Typical devices which use the EIP protocol are PLCs, HMIs, and robot sand IO adapters. The EIP protocol follows the Open Systems Interconnection (OSI) model. A layer-7 protocol in the OSI model is illustrated in Figure 9. In the application layer, the CIP protocol provides the application object library. In the presentation layer, it supports messaging services. The session layer provides message routing and connection services. EIP supports two types of message connection: explicit message connection and implicit connection. Explicit Message connection is a point to point transaction between two nodes using the TCP/IP protocol to access devices in the system. An implicit connection

is a one to many connections to provide broadcast or multicast services using the UDP/IP protocol over the Ethernet network.

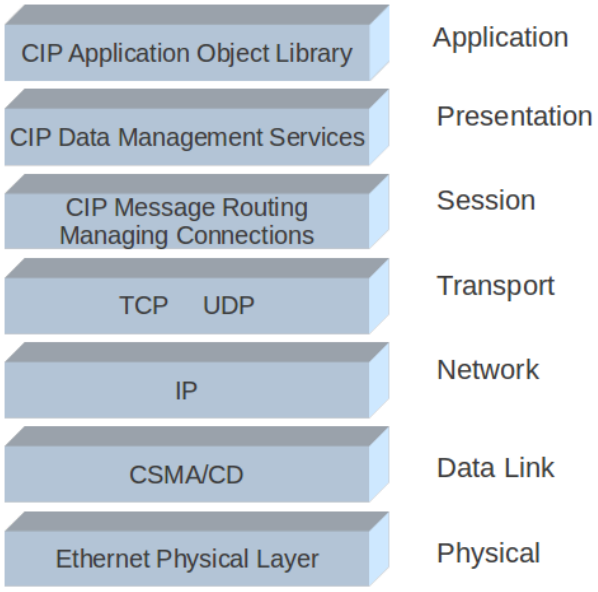

Figure 9 The Ethernet/IP Protocol OSI Stack Representation

The normal EIP session generally follows a common set of steps. A client opens a TCP connection to the port 44818 (EtherNet/IP port) on the server. The client then sends a non-message command NOP to the server to test whether the server is reachable. The NOP command is used to check the TCP connection status in the EIP protocol. Both the server and the client can generate this command, and the receiver doesn't respond to the NOP command. Next, the ListIdentity command is optionally issued by the client using a UDP broadcast to the network. The server responds to the ListIdentity command with the device information such as vendor ID, serial number, device type etc. Next the client sends the RegisterSession command to open a new session. The server responds with an

EIP packet including a session handle. This completes the EIP session connection establishment stage. Figure 10 shows the EIP protocol session establishment. The dashed line means a message is optional. After the session is established, the client sends the SendRRData command to request the server data. The server responds to the client's request with system information system such as sensor measurement readings or register contents. EIP continues in this session transmission stage until a session is terminated. In the EIP session termination stage, the client sends the UnregisterSession command to the server to terminate the current session, and the session expires. Figure 10 shows the EIP protocol session transmission and termination. The dashed line in Figure 10indicatesthat the service is not necessary for every session.

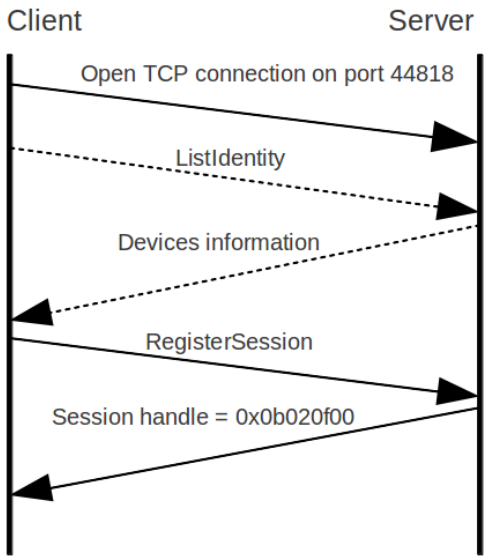

Figure 10 EIP Session Connection Stage

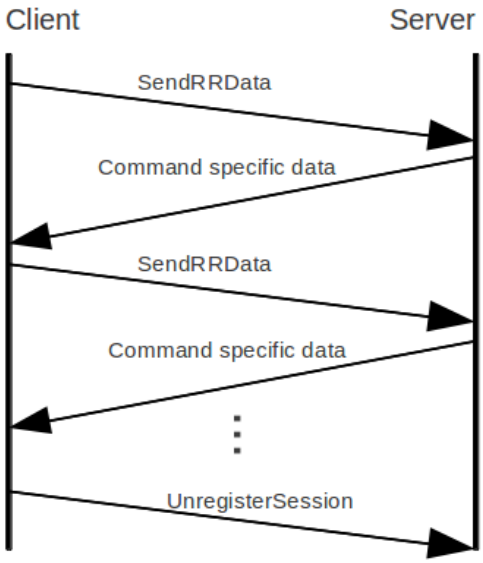

Figure 11 EIP Data Transmission and Session Termination

Three major sources of vulnerabilities in the EIP protocol are identified. The vulnerabilities will be addressed in terms of attacks in order to analyze the potential threats of the vulnerabilities.

Reconnaissance Attack

Due to the fact that the EIP protocol lacks an authorization mechanism, attackers can craft Listidentity packets to broadcast in the control system network to retrieve EIP protocol based device information. The Listidentity command was discussed in the previous section. After broadcasting a ListIdentity packet, devices will respond with information as follows: device IP address, EIP port number, vender ID, device type which specifies the device belongs to a communication adapter or PLC, product code, revision, status, serial number, product name length, product name and state.

EIP packets are sent in plain text and therefore an attacker can see this information during the reconnaissance attack stage. This vulnerability can be used to analyze process control system network topology, identify the gateway address or switch address and subsequently launch other attacks such as Denial of Service attacks against these nodes. The attacker also can use device information to identify already known hardware and software vulnerabilities from vulnerability databases.

An exploit was created to test this vulnerability in the SCADA lab. A hand crafted ListIdentity command packet was created. The crafted packet format is showed in in Table 4. An example of the ListIdentity packet is showed under the Table 4.The crafted EIP packet was broadcasted using the UDP protocol. The UDP port number used was 44818.

Table 4 EIP Packet Format

command	length	Session handle	status	Sender content	options

```
Command: (0x0063) List Identity
Length: 0
Session handle: 0x00000000
Status: success
Sender content: 0x0000000000000000
Options: 0x00000000
```

After receiving the packet, the devices will report their identity information to attacker. In the chemical processing control system testbed, there are three devices (the

master, the RL-PLC and the protocol convertor) based on EIP protocol, each of them responded with one ListIdentity response packet including the device details listed above. The protocol convertor identity information is shown in Table 5 as an example (vender ID is stricken from this report). The other two devices identity information formats are similar to the information in Table 5.

Table 5 Device Identity Information

Item name	Item value
Vendor ID	--- --- --- --- ---
Device type	Communication adapter
Product code	120
Revision	17.02
Status	0x0060
Serial number	0x1a27317f
Product name length	23
Product name	1769-L35E Ethernet Port
state	0x03

As described in the last section, this reconnaissance attack may allow attackers to gain device information to formulate future attacks against a SCADA system. During the reconnaissance attack stage, an attacker will obtain device information and search vulnerability databases to find weaknesses of a device or system. For example, some types of PLCs have vulnerabilities on their CPU modules and Ethernet modules, the attacker can send malformed packets to reset the CPU module or the Ethernet module.

Denial of Service Attack

Two types of DOS attacks against the EIP protocol were developed. The first attack is a packet flooding attack. The second is a EIP Session Handle Hijacking Attack.

Packet Flooding: The concept of the packet flooding type of Denial of Service attack is to prevent th e legitimate remote device's system infor mation from reaching the master. The PLC or the pro tocol convertor reports the sensor measurement reading to the master.

In order to launch the DOS attack, the attacker flood smalformed EIP packets to stop information exchange. The malformed packet format is similar the protocol shown in Table 4 except the contents of the packet are as follows:

```
Command: (0x0063) List Identity
Length: 0
Session handle: 0x00000000
Status: success
Sender content: 0x0000000000000000
Options: 0x00000000
```

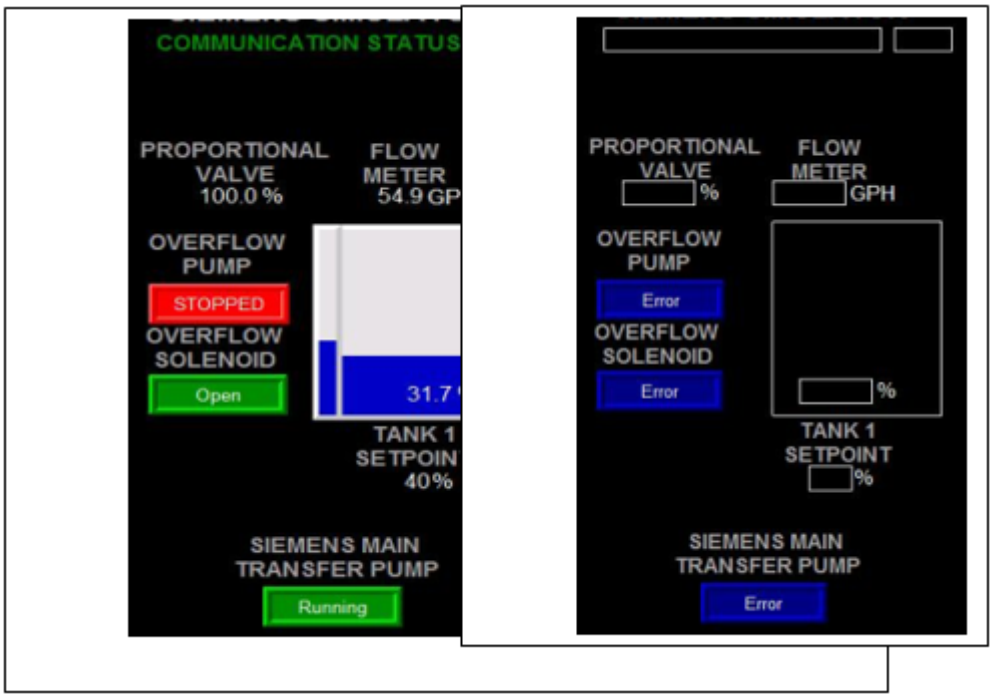

Figure 12 Chemical Processing System HMI

This attack floods malformed EIP command packets to the SCADA network in order to consume the remote device's resources such as CPU memory and process or bandwidth and network bandwidth. In the chemical processing control system network, the PLC and the communication protocol convertor use UDP multicast to transmit sensor measurement reading information. Due to the fact that each EIP multicast packet includes an EIP sequence number, the next sequence number is the current value plus one. The receiver will check the sequence number before receiving the packet. The malformed packet includes an incorrect EIP sequence number. The flood of malformed packets interrupts the normal network traffic flow and blocks real multicast packets leading the master to lose communication with each PLC. The HMI software loses the current system information. Figure 12 shows the part of HMI software before and during the DoS attack.

In the left side of Figure 12the HMI software shows a normal situation. The normal HMI screen shows remote device communication status, valve and pump status, and the water measurement value, and flow speed. During the DoS attack, critical system information is blocked by the flood of malformed packets. The HMI software loses the ability to monitor the state of the physical system. The right side of Figure 12 shows the HMI with blank graphical areas and error messages where system status information is normally shown.

EIP Session Handle Hijacking Attack: There are two requirements for of the *EIP Session Handle Hijacking Attack*. First, the attack requires the ability to spoof a legitimate HMI PC or master IP addresses on the SCADA network. Second, the attack requires the ability to calculate the next TCP sequence number. The first requirement is needed so that the attacker can inject malicious traffic and the second is required for the so that packet will appear legitimate.

The high level idea of this attack is to terminate the EIP session connection between the HMI PC and the master. The key to this attack is that the terminate session packet has to be sent after the EIP session is established. To achieve this, the attacker uses a reconnaissance attack to identify device's function in the network and sends the terminate session packet with correct TCP SYN number.

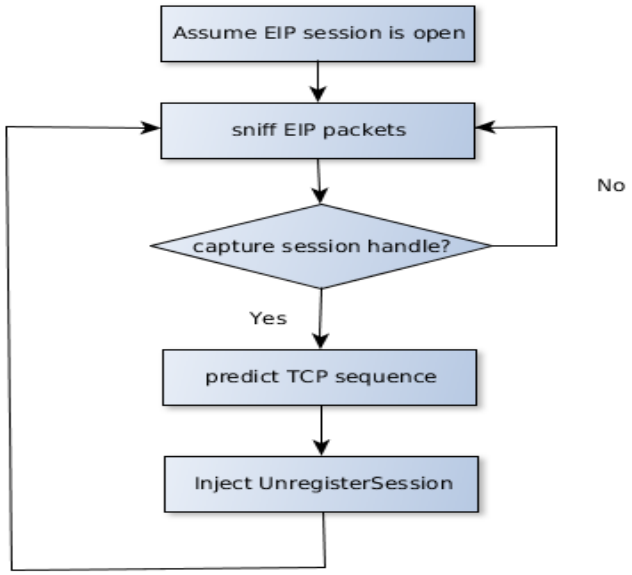

Figure 13 EIP Session Handle Hijacking Attack

Figure 13 illustrates the attack sequence. Here we assume the EIP session is already established. The victim HMI PC first initiates the EIP SendRRData command to request the system state information, the malware discovers the session handle by continuously sniffing the commands from the HMI PC. Once the session handle, the malware starts to predict the next TCP SYN sequence number. Next the malware spoofs the HMI PC IP address and sends a malformed Unregister Session packet to the master. The format of the message is similar to the protocol described in Table 3 and the content of this message is as follows:

```
Command: (0x0066)unresgister session
Length: 0
Session handle: (captured session handle)
Status: success
Sender content: 0x0000000000000000
Options: 0x00000000
```

The receiver closes the current TCP/IP connection when it receives the malicious command packet. The attacker needs to spoof the HMI PC and response this request to close the TCP/IP connection. After closing a new EIP session, the HMI PC may initiate a new EIP session; the malware can terminate it again continuously.

This chapter discussed cyber-attacks against both serial based and Ethernet based SCADA networks. The work in this chapter includes introducing the classification of attacks on serial based SCADA network and the classification of attacks on Ethernet based SCADA networks. Also this chapter discusses vulnerabilities of SCADA network protocols and the effects of attacks on the SCADA systems.

CHAPTER IV

SCADA DATASETS

The KDD Cup 1999 dataset [xc] provides a labeled dataset for intrusion detection researchers in the information technology (IT) security area. The KDD Cup 1999 dataset includes about 5 million connection records which are labeled as normal connection records or as one of four categories of attacks: probing attacks, denial of service (DOS) attacks, user to root (U2R) attacks, and remote to local (R2L) attacks. Each record is about 100 bytes in length. Each connection record includes 41 attributes such as time duration, protocol type, service name, etc. Probing attacks are information gathering attacks which map the network to gather information, such as which services are available, which ports are open or what type of operating system runs on the target host. DOS attacks can lead hosts to run out of computing capability or memory resources so that they refuse to service legitimate requests. U2R attacks aim to improve an attacker's authority to root. R2L attacks aim to gain local access from remote machines. The KDD Cup 1999 dataset is used as a benchmark to validate different intrusion detection classification algorithm’s performance. The training dataset includes 494,021 instances, in which Probe attacks comprise 4,107 (0.83%) instances, DOS attacks comprise 391,458 (79.24%) instances, the R2L attacks comprise 1,126 (0.23%)instances, U2R attack comprise 52 (0.01%) instances and normal scenarios comprise 97,277 (19.69%)instances.

The KDD Cup 1999 dataset is a widely used public dataset for evaluating anomaly based intrusion detection systems. However, many researchers in the SCADA security area use their own datasets to test intrusion detection systems. There is no standard dataset which includes normal network traffic and attacks against a SCADA network which can provide researchers a benchmark to compare SCADA IDS performance. This chapter documents a dataset created to allow researchers a common platform to evaluate the performance of data mining and machine learning algorithms designed for intrusion detection systems for SCADA systems. The dataset includes different classes of attacks which simulate different attack scenarios on process control system networks. A common dataset provides IDS researchers with the following benefits. First, not all researchers have access to SCADA equipment to generate their own dataset. A common dataset will therefore allow more researchers to work in this domain. Second, a common dataset allows researchers to independently validate other researcher's results. Third, a common dataset allows the performance of different algorithms to be compared which should lead to more efficient more accurate intrusion detection system designs over time.

The datasets created for this work are stored in the Attribute Relationship File Format (ARFF) for use with the Waikato Environment for Knowledge Analysis (WEKA) [[ciii]]. WEKA is a comprehensive framework of state-of-the-art machine learning algorithms. WEKA allows researchers to compare or verify new machine learning algorithms. WEKA is not only a collection of machine learning techniques, but also can incorporate new algorithms. WEKA provides various built-in classification algorithms,

such as Bayes Network classifier, Radial Basis Function (RBF), Decision Tree, Neural Network, etc.

There are a number of data mining, machine learning and statistical projects that use or wrap WEKA. There are many tools that use WEKA for text analysis such as TagHelper [civ] which can be used to analyze conversational data such as email or chat logs. The BioWeka [cv] project is used for knowledge discovery and data analysis for biologists. The BioWeka project developed a WEKA extension to use various formats of bioinformatics datasets in WEKA. WEKA is also widely used in distributed data mining projects. Weka4WS [cvi] is a framework that was developed for a distributed data mining algorithm on a Grid system. Weka-Parallel [cvii] is a platform based on WEKA which provides cross-validation for classification algorithms.

The organization of the dataset produced for this dissertation is similar to the KDD Cup 1999 dataset's organization. Each instance in the dataset represents one captured network transaction pair. Each entry represents a merged MODBUS query, a client (MTU) to server (RTU) transaction and a response, a server (RTU) to client (MTU) transaction. An instance in the dataset describes network traffic information and the current state of the process control system via content features. Each instance is a row in the dataset and each instance consists of multiple columns. Each column represents the value of a feature or attribute. There are two groups of features in the dataset; network traffic features and content features. Each instance also includes a label which is the class of attack or "normal" if the instance is not part of an attack. All features are discussed in detail in this chapter.

Table 6 SCADA Dataset Indices and Descriptions

Dataset Index	Description
I	Gas pipeline system full dataset
II	Water storage tank system full dataset
III	Gas pipeline system reduced size (10%) dataset
IV	Water storage tank system reduced size (10%) dataset

The SCADA network dataset includes four datasets from two serial based process control systems. Dataset I includes transactions from a gas pipeline system and dataset II includes transactions from a water storage tank system. Dataset I contains79556 instances and dataset II contains 193667 instances. The datasets are generated from network flow records captured with a serial port data logger in a laboratory environment. A second set of reduced size datasets were also created. Dataset III is a gas pipeline system dataset with 10% of the instances of Dataset I. Dataset IV is a water storage tank system dataset with 10% of the instances of Dataset II.

The two smaller datasets are provided to reduce memory requirements and processing time when validating classification algorithms. The smaller datasets provide researchers an option to verify the classification algorithms quickly and easily. The smaller datasets were created by sampling the larger datasets. Instances in the smaller datasets were randomly selected from the larger datasets with the constraint that the proportion of each class of attack is the same in the larger and smaller datasets.

All four datasets are available online at http://www.ece.msstate.edu/~morris/icsdatasets to make them available to other researchers.

Communication data logs in SCADA networks not only describe network traffic patterns such as remote device address, function code, etc., information in the payload of the packets also provide a description of the control flow or state of the control system. There are two categories of features in the dataset; network traffic features and content features. Network traffic features describe the communication pattern within the control system. Compared with enterprise IT networks, process control system network topologies are static and the services provided by network attached devices (MTU and RTU) are regular. Some attacks against control systems may change the communication pattern in the network, so network traffic features are used to describe the normal traffic pattern and detect malicious activity. Network traffic features include the device address, function code, length of packet, packet error checking information, and the time interval between packets. Second, content features describe the current state of the control system. This information is contained in the payload of the packet. Content features characterize the behaviors of different nodes in the control system. These features provide the potential to detect attacks that cause nodes such as the MTU or RTU to behave abnormally. Content features include sensor measurements, supervisory control inputs, and distributed control decisions. Sensor measurements vary by system. The attacks described in this paper were implemented and exercised against laboratory scale gas pipeline and water storage tank control systems. Sensor measurements used were gas pressure measured in pounds per square inch (PSI) for the gas pipeline and water level measured as a percent of tank fullness for the water storage tank. Supervisory control inputs include set points, PID parameters, system mode, and pump on/off input. Distributed control decisions include pump state, relief valve state, and alarm states.

Ten network traffic features were developed. The first and second network traffic features are the command device address and response device address. Each MODBUS server has a unique address. For MODBUS serial communication the device address is the device ID of the target device or server, aka, the RTU. The command device address in the command packet should match the response device address in the response packet. This feature is primarily intended to aid in the classification of reconnaissance attacks. MODBUS servers are assigned fixed device addresses. MODBUS serial systems are configured such that all slaves see all master transactions. As such the slave must check the device address before acting upon a packet. The set of device addresses seen by a slave is limited and fixed according to the configuration of the system. Command device addresses or response device addresses which are not in the common set will appear as anomalous.

The third, fourth, fifth and sixth network traffic features are command memory, response memory, command memory count and response memory count. For MODBUS servers, points are grouped into data blocks called coils, discrete inputs, holding registers, and input registers. Coils and discrete inputs represent a single Boolean bit, however, when accessed the legal values are 0x00 and 0xFF. Holding and input registers are 16-bit words. Coils are read and write capable and discrete inputs are only read capable. Holding registers are read and write capable and input registers are only read capable. Each data block may have its own set of contiguous address space or the data blacks may share a common memory address space. This choice is vendor and device specific. The command memory and response memory are coil or register read/write start positions in command packet and response packet. The command memory count feature and response

memory count feature are the number of points to be read or written. These memory features can be used to describe the memory operation behaviors in the network traffic.

The seventh and eighth network traffic features are the command packet length and response packet length. The MODBUS Protocol Data Unit (PDU) is limited to 253 bytes. With the added device ID and CRC fields a MODBUS RTU packet may be 256 bytes. However, MODBUS communications between a master and slave typically are very repetitive. Typically, a small set of read and write commands are used. For the gas pipeline and water storage tank systems used for this work the master repeatedly performs a block write to a fixed memory address followed by a block read from fixed memory address. The read and write commands have fixed lengths for each system and the read and write responses have fixed lengths for each system. However, many of the attacks described in the first section of this paper have different packet lengths. As such the packet length feature provides a means to detect many attacks.

The ninth network traffic feature is the time interval. The time interval is a measurement of the time between the MODBUS query and the subsequent MODBUS response. The MODBUS protocol is a request-response protocol. During normal operation the time interval will vary only slightly. Malicious command injection attack and malicious response injection attacks and DOS attacks often will result in significantly different time interval measurements due to the nature of these attacks.

The tenth network traffic feature is the command/response cyclic redundancy code (CRC) error rate. This feature is a measurement of the rate of CRC errors seen in command and response packets. The command/response CRC error rate characterizes the command and response packet CRC error checking error rate. Because the SCADA

network traffic pattern is regular, the normal command and response CRC error rate is expected to stay in a certain range. For a normal system this rate will be very low. This rate will increase when a system is subjected to a DOS attack such as the Invalid CRC attack.

The content features are different between gas pipeline control system dataset and water storage system dataset. The gas pipeline control system uses a proportional integral derivative (PID) control scheme and the water storage system tank uses an ON/OFF control scheme. A common set of content features are listed in Table 7. Water storage tank system specific features are listed in Table 8. Gas pipeline system specific features are listed in Table 9. Extended descriptions of the features are provided under each table.

Table 7 Common Features for All Datasets

Feature Name	Feature Type	Description
command_address	Network Traffic	Device ID in command packet
response_address	Network Traffic	Device ID in response packet
command_memory	Network Traffic	Memory start position in command packet
response_memory	Network Traffic	Memory start position in response packet
command_memory_count	Network Traffic	How many memory bytes need to be R/W in command
response_memory_count	Network Traffic	How many memory bytes need to be R/W in response
command_length	Network Traffic	Total length of command packet
response_length	Network Traffic	Total length of response packet
crc_rate	Network Traffic	CRC error rate
time	Network Traffic	Time interval between two packets
comm_read_fun	Content	Value of command read function code
comm_write_fun	Content	Value of command write function code
response_read_fun	Content	Value of response read function code
response_write_fun	Content	Value of response write function code
sub_function	Content	Value of sub function code in the command/response
measurement	Content	Water level or pipeline pressure
control_mode	Content	Automatic mode, manual mode, or system off setting
pump state	Content	Pump state
manual pump setting	Content	Manual mode pump setting
label	Classification	Manual classification of this instance

The first five common content features are command read function code, command write function code, response read function code, response write function code and the sub function code. The function code feature allows the IDS to classify instances of uncommon function codes as anomalous.

The sixth common content feature is the process measurement. This feature provides the current measurement value of the water tank water level as a percentage of tank fullness or gas pipeline pressure in pounds per square inch (PSI). Naive malicious response injection attacksand complex malicious response injection attack will influence the process measurement during the attack. This feature aids in detection of these response injection attacks.

The seventh common content feature is system control mode. This feature is captured from a command packet. The system control mode places the system in auto, manual or shut down (off)modes. The value 0 represents off, the value, 1 represents on in manual mode, and the value 2 represents on in automatic mode. In automatic mode a ladder logic program in the programmable logic controller (PLC) connected to the system controls the system. In manual mode control is taken from manual override settings. One type of malicious state command injection attacks tries to modify the system working mode or shut down the system. This feature aims to specify the system working mode during one certain operation stage.

The eighth common content feature is the pump state. Both the gas pipeline and water storage tank systems use a pump to add air or water to the system to meet the target set point. This feature directly represents the pump state. If this feature is 1 then the pump is on and if this feature is 0 then the pump is off. When a system is in automatic

mode the pump state is controlled by the ladder logic program in the PLC connected to the system. In manual mode the pump state is controlled by the manual pump setting which is described below. One type of malicious complex response injection tries to modify this value in order to mask the actual pump working state. This feature can be used to describe the pump state switch sequence during the normal operation and during attack.

The ninth common content feature is the manual pump setting. This feature is captured from a command packet. The manual pump setting indicates the intended pump state if the system control mode is manual. This setting is ignored when the control mode is set to automatic. Malicious state command injection attacks may change the pump working mode continually or intermittently.

Table 8 Unique Features for Water Storage System Datasets

Feature Name	Feature Type	Description
HH	Content	Value of HH setpoint
H	Content	Value of H setpoint
L	Content	Value of L setpoint
LL	Content	Value of LL setpoint

There are four content features specific to the water storage tank system: HH, H, L and LL. These features are related to the water level set point in water storage system. In automatic mode, the RTU ladder logic program running in the PLC connected to the water storage tank attempts to maintain water level between the Land H setpoints using an ON/OFF controller scheme. When the RTU ladder logic program detects that the water level has reached the L level it turns on the water pump. When RTU ladder logic program senses that the water level has reached the H level it turns off the water pump.

The water storage tank includes a manual drainage valve which if open allows water to drain out of the tank. If manual drainage valve is open the water level in the tank will oscillate between the H and L setpoints continuously as the pump cycles on and off to add water to the tank as the water drains. If the manual drainage valve is closed the water level depends on the pump state at the time of closing. If the pump is on when the manual drainage valve is closed, the water level will rise to the H setpoint and the pump will turn off. The water level will remain constant until the manual drainage valve is opened. If the pump is off when the manual drainage valve is closed, the water level will remain constant until the manual drainage valve is opened and the water level subsequently drops to the L level. If, due to a system fault, the water level rises to the HH setpoint or falls to the LL setpoint an alarm is triggered on the human machine interface used to monitor the water storage tank. In manual mode, the pump state is controlled manually by the HMI. An operator can manually activate or deactivate the pump. The manual pump setting feature represents the manual control choice. In manual mode the HMI continues to poll the RTU to read all set points and measurements every second. If due to operator error, the water level rises to the HH setpoint or falls to the LL setpoint the alarm is triggered. The water level set point features describe the ordinary system operation settings. Various malicious command injection attacks aim to modify these set points.

Table 9 Unique Features for Gas Pipeline System Datasets

Feature Name	Feature Type	Description
set point	Content	Target gas pressure in the pipe in pounds per square inch
control_scheme	Content	Control scheme of the water storage system
solenoid state	Content	Solenoid to open gas relief valve state
gain	Content	Gain parameter value of PID controller
reset	Content	Reset parameter value of PID controller
deadband	Content	Deadband parameter value of PID controller
rate	Content	Rate parameter value of PID controller
cycletime	Content	Cycletime parameter value of PID controller

The first content feature unique to the gas pipeline is the set point which is the target gas pressure in PSI. In automatic mode, the RTU ladder logic program running in the PLC connected to the gas pipeline attempts to maintain the air pressure in the gas pipeline using a PID control scheme. The mechanism used by the PID controller to control pressure depends upon the control scheme feature. The control scheme feature selects whether the pump or a solenoid controlled relief valve is used to control the pressure in the pipe. If the control scheme is 0 then the pump is used to control pressure and if the control scheme is 1 then the solenoid controlled relief valve is used to control pressure. With the pump control scheme the pump is turned on and off by the PID controller. With the pump control scheme changes in air pressure are relatively slow since the pressure only drops due to system air leakage which is minimal. With the solenoid controlled relief valve air is allowed to escape the system based on the solenoid feature. With the solenoid controlled relief valve air pressure changes more rapidly since the relief valve allows air to escape at a higher rate than just leakage alone. The solenoid state feature is the state of the solenoid. The state of the solenoid is controlled by the PID controller in automatic mode and matches the solenoid setting in manual mode.

There are five content features related to the PID controller. This group of features control PID controller's behavior, which includes gain, reset, rate, dead band, cycle time. These PID parameters should be kept fixed during the system operation. One type of malicious parameter command injection may try to modify these parameters in order to interrupt the normal control operation discussed in the above section. This feature is potential to describe the difference between normal operation and this type of malicious parameter command injection attack.

The last feature for each instance in all datasets is the label. This feature provides a manual classification of each instance. Table 10 lists the eight possible label values.

Table 10 Instance Classification Values

Label Name	Label Numeric Value	Description
Normal	0	Instance not part of an attack.
NMRI	1	Naive Malicious Response Injection (NMRI) attack
CMRI	2	Complex Malicious Response Injection (CMRI) attack
MSCI	3	Malicious State Command Injection (MSCI) attack
MPCI	4	Malicious Parameter Command Injection (MPCI) attacks
MFCI	5	Malicious Function Command Injection (MFCI) attacks
DOS	6	Denial of Service (DOS) attacks
Reconnaissance	7	Reconnaissance attack

Normal traffic is labeled with the value 0. Normal traffic includes all network packets which are not part of an attack. Naive Malicious Response Injection (NMRI) attacks are labeled with 1. Naive Malicious Response Injection (NMRI) attacks lack sophistication. NMRI attacks leverage the ability to inject response packets into the network but lack information about the process being monitored and controlled. NMRI

attacks may send invalid payloads. A single NMRI attack consists of one dataset instance since this attack injects one malicious payload per attack.

Complex Malicious Response Injection (CMRI) attacks are labeled with 2. CMRI attacks add a level of sophistication above that of the NMRI attacks. CMRI require understanding of the cyber physical system being attacked. CMRI attacks attempt to mask the real state of the physical process being controlled to negatively affect the feedback control loop managing the cyber physical system. A single CMRI attack consists of multiple dataset instances since this attack injects a large number of packets per attack.

Malicious State Command Injection (MSCI) attacks are labeled with 3. MSCI attacks change the state of the process control system abnormally to drive the system from a safe state to a critical state by sending malicious commands to remote field devices. A single MSCI attack consists of one or multiple dataset instances.

Malicious Parameter Command Injection (MPCI) attacks are labeled with 4. Industrial control systems often include tight control loops between the physical process, sensors and actuators, and a programmable logic controller (PLC) directly connected to the sensors and actuators. MPCI may craft invalid controller parameter set points. A single MPCI attack consists of multiple dataset instances since this attack changes the controller parameter constantly.

Malicious Function Command Injection (MFCI) attacks are labeled with 5. MFCI attacks can be used to disrupt the client server communication link. A single MFCI attack consists of multiple dataset instances.

Denial of Service (DOS) attacks are labeled with 6. DOS attacks against industrial control systems attempt to stop the proper functioning of some portion of the cyber physical system to effectively disable the entire system. A single DOS attack consists of multiple dataset instances.

Reconnaissance attacks are labeled with 7. Reconnaissance attacks gather control system network information, map the network architecture, and identify the device characteristics such as manufacturer, model number, supported network protocols, system address, and system memory map. A single MSCI attack is consists of one or multiple dataset instances depending upon the reconnaissance attack scan speed and scan range.

The dataset instances are labeled as normal or by attack class, rather than by the individual attack names from Chapter III. This labeling scheme was chosen to match the KDD Cup 1999 dataset style which labeled by attack class than by a more specific label noting an exact exploit. Attacks in each attack class have similar exploit methods and similar impacts on the process control system.

This chapter describes four datasets which can be used to evaluate performance of data mining and machine learning algorithms for intrusion detection systems in SCADA systems. This network dataset includes different classes of attacks that simulate different attack scenarios on process control system networks. The datasets described in this chapter were used to evaluate the performance of an anomaly based intrusion detection system described in Chapter V.

The datasets described in this chapter are relevant to other industrial control systems (non-MODBUS, non-gas pipeline, or water tank system). The features in the dataset are divided into two groups: network traffic flow and content. Other industrial

control system protocols divide packets into network traffic related fields and content fields. Other protocols include similar though not identical network traffic information such as addresses, function codes, payload, and check sums. Many other industrial control system protocols also follow query response traffic patterns similar to MODBUS. The content features in the provided data set include remote commands and measured system state. Other systems will include similar items which are monitored and updated at similar rates. As such, the dataset described in this chapter can provide a benchmark to measure accuracy of data mining or machine-learning algorithm intrusion detection systems.

Reconnaissance, injection, and denial of service attacks are not unique to SCADA networks. Similar vulnerabilities exist against enterprise networks and small office home office networks. As such, the datasets described in this chapter are also relevant to non-industrial control system intrusion detection system research.

CHAPTER V

SCADA NETWORK INTRUSION DETECTION

Critical infrastructures are exposed to serious vulnerabilities that can be exploited by attackers. Research on cyber threat and vulnerability analysis shows the security challenges and the secure approaches in SCADA networks. There have been several real-world documented incidents and cyber-attacks affecting SCADA system, which clearly illustrate critical infrastructure vulnerabilities. These reported incidents demonstrates that cyber-attacks on SCADA systems might produce a variety of financial damage and can be harmful events to humans and the environment.

There are many security mechanisms that can secure process control system such as access control, authorization etc. These solutions are required in the defense in depth for process control systems. However, none of them are completely secure. Attacks continue to defeat security mechanisms to penetrate systems and execute attacks against confidentiality, integrity and availability. Because current security mechanisms can never be 100% secure, intrusion detection systems are needed to monitor attacks in progress.

It is hypothesized that building a SCADA system intrusion detection dataset to evaluate SCADA IDS performance is possible through the SCADA system threat model analysis and different exploiting methods for both serial based and Ethernet based SCADA network. Most SCADA systems are designed for using over a long period, the topology of SCADA networks is static and the network communication traffic pattern is

regular. Therefore intrusion detection systems can use traffic patterns to detect abnormal communication traffic. Furthermore, process control systems are based on the feedback control loop, the system measurements are inputs to the controller such as PID controller. Based on the system current measurements, system setting parameters and controller parameters, the next measurements can be predicted.

SCADA networks are real time systems. Typically the availability is important and computer capability and memory resources in SCADA devices are limited. In order to monitor the network traffic and detect malicious behaviors in process control systems while changing the SCADA infrastructure's protocol or architecture at minimum cost, an Intrusion Detection System (IDS) is required in the SCADA network.

There are two main types of detection techniques known as signature based detection and anomaly based detection. The signature based IDSs apply the attack signature database to determine whether the signature has been triggered. It is effective when detecting known attacks and with low false positives. Anomaly based detection uses mathematical techniques such as machine learning or data mining to check the deviations from the predefined normal behavior model. Anomaly based detection may detect new attacks.

The signature based intrusion detection approach monitors and matches the traffic record to known misuse patterns or rules to detect rules violations. Snort [xxviii] is an open source signature based network intrusion detection system. This research applies Snort IDS with signature rules combined with system state prediction rules to detect malicious behaviors in SCADA networks.

Anomaly IDS uses statistical models or machine learning algorithms to classify network traffic as normal or abnormal (or into smaller sub-classes). Various model types or classifiers can be used to build the anomaly IDS. This research applies Bayesian networks, neural networks and decision trees to classify network transactions as normal or abnormal. A decision tree algorithm is selected due to the fact that it is one of most useful techniques in deterministic classification. Bayesian network is a fundamental statistical approach to the problem of pattern classification. Neural network is also a widely classification algorithm in intrusion detection system. So these three classification algorithms are selected in this research.

This research builds a SCADA system intrusion detection system through the SCADA system threat model analysis and different exploiting methods both serial based and Ethernet based SCADA network. Most SCADA systems are designed for using a long period, the topology of SCADA network is static and the network communication traffic pattern is regular. So intrusion detection system can use traffic pattern to detect abnormal communication traffic.

Anomaly Based Intrusion Detection Result

This section describes the detection results of anomaly IDS in serial communication based process control system. We used a decision tree to classify network transactions as normal or abnormal. A decision tree consists of non-leaf nodes and leaf nodes. Each non-leaf node represents a test over an attribute and each leaf node corresponds to a class of decision result. There are many algorithms for decision trees which includes ID3 [cviii], C4.5 [cix] and CART [cx]. Most of the decision trees algorithms implement a top down strategy; i.e. from the root to the leaves. There are two steps in the

decision tree classification: constructing the tree and applying the tree to the dataset. In this research, C4.5 algorithm was selected because it is most representative and it is evolved from ID3 algorithm. The performance of the detection was measured by the detection rate and the false positive rate. Detection Rate (DR) represents the number of malicious packets that can be detected divided by the overall number of intrusions. False Positive rate (FP) is the number of normal packets that are detected as anomaly divided by the overall number of normal packets. The features discussed in last section are used as input to IDS.

The work not only use the first and second group of features discussed in last chapter, but also use prediction feature category.

Predictive features are calculated from the present system state using a physical model of the control. The basic idea of these features is to construct a feedback control model to describe the predicted system behaviors. Predictive features for process control systems include predicted sensor measurements. Based on the control system model, the input is the last sensor reading values received by MTU, and the control system's setting and the output is the predicted system measurements. The Next predictive feature is the predicted trend of system state which characterizes whether the predicted system state will go to another state or keep in the same one. The trend of system state includes measurement changing speed, derivative of process measurement. For example, it describes whether the measurement is increasing or decreasing or staying in a valid range. Predictive features include the following features:

Speed: Describes the increasing or decreasing rate of the gas pressure in the pipeline. Malicious complex response injection such as slow or fast response injection

may influence the measurement speed which is showed in HMI. This feature is potential to describe the difference between normal measurement changing in the response injection attack.

Derivative of process measurement: Characterizes the changing rate of the gas pressure speed. For example, in the gas pipeline control system the gas pressure changing rate is dependant on the current pressure value, the pump state, the valve state etc. It cannot increase the leakage speed when the pump is off and the valve is closed. This feature will be used to specify the system measurements speed behavior and help to detect malicious response injection attack.

Predicted measurement: This feature is an output of control system model, and the input of the model is system settings and the last invalid measurement. It characterizes the next value of the gas pressure. It is used to describe the distance between predicted measurement and the actual measurement in the next control output which may have contribution to detect malicious response injection attack.

Predicted speed: This feature is similar to the speed feature but it is computed with predicted measurement, and can be used to describe the distance between predicted speed and the actual measurement changing speed and may improve detection rate and decrease false alarm.

The Table 11 shows the anomaly based intrusion detection results for gas pipeline system and water storage system.

Table 11 Anomaly Based IDS Result

Dataset	Gas Pipeline System		Water Storage System	
	Detection Rate	False Positive	Detection Rate	False Positive
Reconnaissance	92.9%	0.6%	93.1%	0.7%
NMRI	95.0%	0.8%	97.6%	1.8%
CMRI	97.0%	0.4%	96.7%	2.2%
MSCI	94.5%	0.7%	95.1%	2.1%
MPCI	94.5%	0.4%	97.0%	0.9%
MFCI	98.0%	0.2%	98.0%	0.3%
DOS	100%	0.0%	100.0%	0.0%
Normal	99.4%	1.3%	99.5%	0.9%

The hardware is used in this experiment is described as follow. The CPU is Intel Core 2 Duo 2.26GHZ. The memory is 4G byte. The operating system is used in this experiment is Ubuntu 12. The total number of instance in gas pipeline system dataset is 79556. The time to build the model is 36.83 seconds. The Kappa statistic is 0.9844. The total number of instance in water storage system dataset is 193667. The time to build the model is 77.07 seconds. The Kappa statistic is 0.9885. The performance of the detection was measured by the detection rate and false positive rate. Detection rate represents the ratio between the number of malicious packets that can be detected and overall number of intrusions. False positive rate is the ratio between the number of normal packets that are detected as anomaly and the overall number of normal packets.

The anomaly based IDS detection rate for reconnaissance attacks was 92.9% for the gas pipeline control system and 93.1% for the water storage tank control system. A review of the misclassified attack case shows that the instances in reconnaissance attack class are well classified by the anomaly based IDS. The features in the training dataset related with command/response address, command/response write and read function code,

sub function code, command/response write and read memory are used for anomaly based IDS to describe reconnaissance attack profile. The IDS failed to detect the malicious packets that contain valid features information which listed above. During the reconnaissance attack, the network traffic contains valid device address, function code and memory address. The data mining algorithms will classify these instances as normal class due to the reason that in the training stage, the number of normal instances with valid features is much greater than the reconnaissance attack instance with valid features. So the IDS will misclassify these malicious packets as normal packets and not trigger alarms. The false positive rate for the reconnaissance attacks was 0.6% for the gas pipeline control system and 0.7% for the water storage tank control system. This means the anomaly based IDS misclassify the normal traffic or other types of attack as reconnaissance attacks. This is due to the total number of reconnaissance attack instances' low presence in the training dataset.

The anomaly based IDS detection rate for NMRI attacks was 95% for the gas pipeline control system and 97.6% for the water storage tank control system. A review of the misclassified attack case shows that the instances in NMRI attack class are also well classified by the anomaly based IDS. The features in the training dataset related with process control system measurement, measurement speed measurement derivative system working mode and pump working status are used for anomaly based IDS to describe NMRI attack profile. The IDS failed to detect the malicious packets that contain gas pressure measurement or water level measurement in the valid normal measurement range. During the NMRI attack, the network traffic may contain valid process control system measurement, measurement speed measurement derivative system working mode

and pump working status. These features are very relevant with features in the normal network traffic class. So the IDS will misclassify these malicious packets as normal packets and not trigger alarms. The false positive rate for the NMRI attacks was 0.8% for the gas pipeline control system and 1.8% for the water storage tank control system. This means the anomaly based IDS misclassify the normal traffic or CMRI attack class as NMRI attack class. This is due to the reason that some CMRI attack process is very short so that the presence of injected packets is low and the traffic pattern is similar with NMRI attack class.

The anomaly based IDS detection rate for CMRI attacks was 97% for the gas pipeline control system and 96.7% for the water storage tank control system. A review of the misclassified attack case shows that the anomaly based IDS failed to detect some CMRI packets. The features in the training dataset related with process control system measurement, measurement speed measurement derivative system working mode and pump working status are used for anomaly based IDS to describe CMRI attack profile. During the CMRI attack, the network traffic may contain valid process control system measurement, measurement speed measurement derivative system working mode and pump working status. The measurement behavior of Calculated Sensor Measurement Injection attack is closed to the normal traffic. These features are very relevant with features in the normal network traffic class. So the IDS will misclassify these malicious packets as normal packets and not trigger alarms. The false positive rate for the CMRI attacks was 0.4% for the gas pipeline control system and 2.2% for the water storage tank control system. This means the anomaly based IDS misclassify the normal traffic or NMRI attack class as CMRI attack class. This is due to the reason that the measurement

behavior pattern of Calculated Sensor Measurement Injection attack in the CMRI attack process is similar with it in the normal traffic class.

The anomaly based IDS detection rate for MSCI attack was 94.5% for the gas pipeline control system and 95.1% for the water storage tank control system. A review of the misclassified attack case shows that the instances in MSCI attack class are also well classified by the anomaly based IDS. The features in the training dataset related with process control system setpoints, system working mode and pump working mode and pump working status are used for anomaly based IDS to MSCI attack profile. The IDS failed to detect the malicious packets that contain gas pressure setpoint or water level setpoints in the valid range. During the MSCI attack, the network traffic may contain valid combination of system setpoints, system working mode, pump working mode and valid current system state. These features are very relevant with features in the normal network traffic class. So the IDS will misclassify these malicious packets as normal packets and not trigger alarms. The false positive rate for the MSCI attacks was 0.7% for the gas pipeline control system and 2.1% for the water storage tank control system. This means the anomaly based IDS misclassify the normal traffic as MSCI attack class. This is due to the reason that the some combinations of system setpoints, system working mode, pump working mode and valid current system state pattern in the MSCI attack process is similar with it in the normal traffic class.

The anomaly based IDS detection rate for MPCI attacks was 94.5% for the gas pipeline control system and 97.0% for the water storage tank control system. A review of the misclassified attack case shows that the instances in MPCI attack class are well classified by the anomaly based IDS. The features in the training dataset related with

process control system setpoints and the controller parameters are used for anomaly based IDS to MSCI attack profile. The IDS failed to detect the malicious packets that contain gas pressure setpoint or water level setpoints and the controller parameters in the valid range. During the MPCI attack, the network traffic may contain valid system setpoints, controller parameters and valid current system state. These features are very relevant with features in the normal network traffic class. So the IDS will misclassify these malicious packets as normal packets and not trigger alarms. The false positive rate for the MSCI attack was 0.4% for the gas pipeline control system and 0.9% for the water storage tank control system. This means the anomaly based IDS misclassify the normal traffic and MSCI as MPCI attack class. This is due to the reason that the some combinations of system setpoints, controller parameters and valid current system state pattern in the MPCI attack process is similar with it in the normal traffic class and MPCI attack class.

The anomaly based IDS detection rate for MFCI attacks was 98.0% for the gas pipeline control system and 98.0% for the water storage tank control system. A review of the misclassified attack case shows that the instances in MFCI attack class are well classified by the anomaly based IDS. The features in the training dataset related with function code and sub function are used for anomaly based IDS to MFCI attack profile. The IDS failed to detect the malicious packets that contain function code and sub function in the valid range. During the MFCI attack, the network traffic may contain valid function code and subfunction code. These features are very relevant with features in the normal network traffic class. So the IDS will misclassify these malicious packets as normal packets and not trigger alarms. The false positive rate for the MFCI attack was

0.2% for the gas pipeline control system and 0.3% for the water storage tank control system. This means the anomaly based IDS misclassify the normal traffic as MFCI attack class. This is due to the reason that the normal traffic also contains very low presence network packets that include same function code and subfunction code as in the MFCI attack.

The anomaly based IDS detection rate for DOS attacks was 100% for the gas pipeline control system and 100% for the water storage tank control system. A review of the misclassified attack case shows that the performance of the IDS is very effective. The false positive rate for the DOS attack was 0% for the gas pipeline control system and 0% for the water storage tank control system. For the DOS attack detection, because MODBUS's maximum payload length limit is 256 bytes, the data logger will store packet with maximum data length. Furthermore, the DOS floods meanness characters in traffic, the MODBUS packet's device address, function code and other information in the packet doesn't contain any useful information. There are many features in the dataset to describe this abnormal behavior and classification algorithms can detect this type of attack with high detection rate and low false positive.

Signature Based Intrusion Detection Result

This section provides a set of rules for a signature based intrusion detection system. The rules described in the section are designed to detect the attacks described in chapter 3. The rules are divided into two types; stand-alone and state-based rules. The set of rules developed for this work are listed in Table11, stand-alone rules, and Table 12, state based rules. Stand-alone rules parse a single MODBUS packet looking for a match to a specific signature. If the signature is present in the parsed packet then the packet is

classified as a match and an alert is issued. The stand-alone rules are implemented using the Snort intrusion detection tool. The second type of rule is called state based. State based rules require knowledge from previous MODBUS packets or from another source, such as a process sensor. This extra knowledge may be related to the protocol state or the state of the industrial control system being monitored. State based rules are processed using a Snort pre-processor, hence forth referred to as the state based layer. Snort passes the MODBUS payload in its entirety to the state based layer. The state based layer is written in the C programming language. A set of C-language structures were developed to store the state of the MODBUS protocol and a historical model of the state of the industrial control system. The model of the state of the industrial control system is system specific and requires expert knowledge to develop. The protocol state structure stores the last received MODBUS packet. The historical state for the industrial control system holds the command state and the process state. The command state is updated each time a command is sent to a MODBUS server. For the gas pipeline the command state includes items such as the on/off state of the pump, the open/closed state of the relief valve, the system mode (manual or automatic), copies of set points, and other process specific control information. The process state includes measurements related to the process. For the gas pipeline the process state includes the last pressure reading and other system measurements. Figure 14 shows the intrusion detection system architecture with separate stand-alone and state based layers.

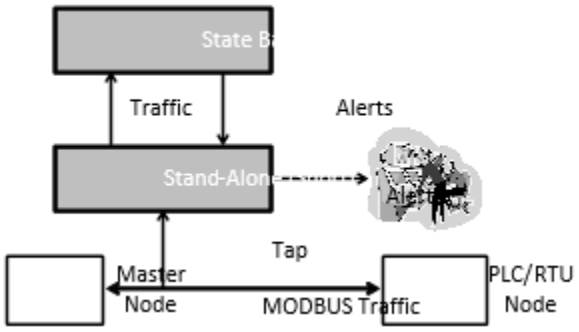

Figure 14 Intrusion Detection System Architecture

Table 11lists IDS stand-alone IDS signatures and Table 12lists IDS signatures which are state based and require a preprocessor.

Stand-alone rules #1-7 are used to detect reconnaissance attacks. MODBUS address scans differ based upon the upper network layers. MODBUS/TCP address scans search for IP addresses with MODBUS servers. MODBUS RTU and ASCII use a 1 byte address field. MODBUS RTU and ASCII address scans search for address values which provoke a response. MODBUS systems typically have a static set of member nodes each with a fixed address. A white list of system address, whether IP addresses or 1 byte addresses for MODBUS RTU and ASCII systems, can be developed. Stand-alone rules #1 and #2 from Table 12 are used to detect packets which are addressed to a system not in the RTU/ASCII or TCP address white lists respectively. A system specific white list can be developed which includes all legal function codes a MODBUS server supports and is allowed to receive. The function code white list can include all public MODBUS function codes and any user defined function codes. However, many systems use only a

subset of the public MODBUS function codes. As such the function code white list should be limited to function codes used by the system being protected.

Stand-alone rule #3 alerts if a packet is received which includes a function code not in the function code white list. Device ID scans are detected with stand-alone rule #4 which alerts if the function code is 0x11 or 0x2B. Since the allowed function code white list will not include function codes 0x11 and 0x2B stand-alone rule #3 will also alert for Device ID attacks. Care should be taken when implementing the device ID scan rule since it effectively disallows systems and their operators from reading device IDs. If reading the device ID is required the device ID scan rule should be disabled. Point scans are detected using 3 rules.

First, stand-alone rules #5 and #6 use address white lists to detect read and write transactions address disallowed memory regions. The two address white lists are a writeable regions white list and a readable regions white list. These white lists include a list of contiguous address regions which allow memory writes and reads respectively. Rules #5 and #6 rules confirm the start address is within the white listed address region. Rules #5 and #6 also compute the end address and confirm the end address is within the white listed address region. The end address is computed by adding the read or write length to the start address. The length value is available for all write and read function codes. Rules #5 and #6 are implemented as separate stand-alone rules for each write and read function code. They are described here as two rules to save space.

Stand-alone rule #7 is also used to detect point scans. Stand-alone rule #7 alerts if an invalid address exception code is detected for a read or write function code. Properly functioning MODBUS clients should not attempt to access an address which is not

configured for reading on the target MODBUS server. As such any instance of the invalid address exception code is evidence of a points scan. Stand-alone rule #7 is not adequate to detect all points scans as it is possible certain memory regions are available on a MODBUS server by default while not actually in use for the specific control system being protected. The white lists associated with rules #5 and #6 should only include address regions needed for the specific control system being protected. Stand-alone rule #6 is also used to detect memory dump attacks. The read address white list should only include read addresses reserved for access through the network and should not include addresses used for internal program variables.

Table 12 MODBUS Stand-alone Intrusion Detection Rules

Number	Name	Description	Attack Detected	
1	RTU/ASCII Invalid Address	packet address ∉address white list	1	Done
2	TCP Invalid Address	IP address ∉IP address white list	1	Done
3	Function Code Scan	packet function code ∉ function code white list	2	Done
4	Device ID Scan	function code ∈ {0x11, 0x2B}	3	Done
5	Write Points Scan	(start address && end address)∈ write address region white list	4	Done
6	Read Points Scan	(start address && end address)∈ read address region white list	4, 5	Done
7	Invalid Address	Response & function code ∈{0x81, 82, 83, 84, 85, 86, 8F, 90, 95, 96, A2, A3, A4} & exception code = 02	4	Done
8	Invalid PID Parameter	gain != 115 | rate != 0 | reset != 0.2 | deadband != 0.5 | cycle time != 1.0	21	Done
9	Gas Pipeline Invalid Set Point	H > 30 | L < 5 | HH > 40 | LL < 0	22	Done
10	Oil Storage Tank Invalid Set Point	H > 80 | L < 20 | HH > 90 | LL < 10	22	Done
11	Force Listen Only Mode	function code = 0x08 & sub-function = 0x0004	23	Done
12	Restart Communication	function code = 0x08 & sub-function = 0x0001	24	Done
13	Clear Communications Event Log	function code = 0x08 & sub-function = 0x0001& data = 0xFF00	25	Done
14	Change ASCII Input Delimiter	function code = 0x08 & sub-function = 0x0003	26	Done
15	Illegal Packet Length	Packet length > 255	28	Done
16	Invalid Response Payload Content	Response & function code is 0x80 & each byte in payload is 0x00 or 0xFF	6	Done
17	Invalid Response Payload Size	Response & function code is 0x80 & payload size != request payload size in the command	7	done
18	Invalid Error Response	Function code is 0x80 & exception code !∈{0x1, 2, 3, 4 }	8	done

Table 13 MODBUS State Based Intrusion Detection Rules

Number	Name	Description	Attack Detected	
1	Gas Pipeline Pressure Negative Response	Response & function code = 03& point address = 04 & value <0	9	Done
2	Oil Storage Tank Fill Level Negative Response	Response & function code = 03 & point address = 07 & value <0	9	Done
3	Gas Pipeline Measurement Out-of-bounds	Response & function code = 03& point address = b7 & (value < LL OR value > HH)	10	Done
4	Oil Storage Tank Measurement Out-of-bounds	Response & function code = 03 & point address = e9 & (value < LL OR value > HH)	10	Done
5	Gas Pipeline Measurement Max Rate of Change	Response & function code = 03 & point address = b7 & slop e > SL_{max}& slope != 0	11, 12, 15, 16	Done
6	Oil Storage Tank Measurement Max Rate of Change	Response & function code = 03& point address = e9 & (M_1-M_0/T_1-T_0) > SL_{max} & slope != 0	11, 12, 15, 16	Done
7	Gas Pipeline Measurement Min Rate of Change	Response & function code = 03 & point address = b7 & slope < SL_{min} & slope != 0	14, 17	Done
8	Oil Storage Tank Measurement Min Rate of Change	Response & function code = 03& point address = e9& slope < SL_{min} & slope != 0	14, 17	Done
9	Gas Pipeline Constant Measurement	Response & function code = 03& point address = b7& NC > T & SystemMode = AUTO	13	Done
10	Gas Pipeline High Pressure Critical State	pressure > HH & pump = ON & relief valve = CLOSED	18, 19, 20, 21, 22	Done
11	Gas Pipeline Low Pressure Critical State	pressure < LL & (pump = OFF | relief valve = OPEN)	18, 19, 20, 21, 22	Done
12	Oil Storage Tank High Liquid Level Critical State	liquid level > HH & pump = ON	18, 19, 20, 21, 22	Done
13	Oil Storage Tank Low Liquid Level Critical State	liquid level < LL & pump = OFF & system = ON	18, 19, 20, 21, 22	Done
14	Invalid CRC Count	invalid CRC count > 5 in time window 5	27	Done
15	MODBUS Flood	MODBUS Packet Count > 5 in time window 5	28	Done

State based rules #1 and #2 alert when sensor measurements are negative. These rules are system specific. State based rule #1 alerts for a negative pressure measurement for a gas pipeline system. State based rule #2 alerts for a negative water level measurement for an oil storage system. In each case the rule must be programmed with the function code used to read the measurement, the exact address of the point which stores the measurement the measurement, and the width of the measurement in bytes. There may be more than one function code used to poll the MODBUS server for the measurement. In this case multiple instances of the rule can be created to cover each case. Also, often MODBUS clients will read many points in a single read. In this case state based rules #1 and #2 should be updated to check the correct set of bytes within the larger read payload. These rules are state based because MODBUS read responses do not include the address which was read from. As such, the state based layer stores the read command details, including the read start address and quantity, and uses this information within the rule.

State based rules #3 and #4 alert when process measurements are grossly out of bounds. These rules are system specific. State based rule #3 alerts for an out-of-bounds pressure measurement for a gas pipeline system. State based rule #4 alerts for an out-of-bounds water level measurement for an oil storage system. Both rules are programmed with extreme limits for their respective process. These rules are configured as stand-alone rules for this work implying that the extreme limits for the process are static or are changed infrequently. For processes in which the extreme limits are variable the rules could be converted to state based rules. In this case, a C structure would be used to store the current extreme limits. The extreme limits could be updated by out of channel

communication such as a secure socket opened between the IDS and the MODBUS client which changes the extreme limits. The extreme limits may be set relative to the alarm thresholds for the process measurement. In this case updates to the extreme limits could be learned by sniffing the network traffic for commands which write to the alarm value points. Monitoring network traffic to learn the alarm set points is less trustworthy than learning of changes through an out of channel trusted link. Similar to state based rules #1 and #2 these rules are state based because MODBUS read responses do not include the address which was read from.

State based rules #5 - #8 alert if the rate of change of a sensor measurement exceeds or falls below specific maximum and minimums respectively. State based rules #5 and #6 alert for high rates of change. A high rate of change may be a symptom of a high slope, sporadic, or random measurement injection attacks. As successive measurements are observed in network traffic the most recent measurement value and the timestamp of the latest measurement are stored in the state based layer. As new measurements are observed the rate of change of process measurements is calculated and compared with a predefined maximum rate of change value. The rate of change is calculated using equation 1, where M0 is the current measurement, M1 the previous measurement, T1 is the time stamp of the current measurement, and T0 is the timestamp of the previous measurement. Control systems tend to poll sensor measurements periodically meaning T1-T0 will be approximately constant. For state based rules #5 and #6 the term SLmax is the maximum rate of change allowed. In some systems the separate maximum rates of change will be defined for the rising and falling measurement case. In such cases two rules will be needed for each system. The maximum rate of change is

system specific and must be defined in consultation with a system expert. This rule will not detect all sporadic and random sensor measurement injections since some injected measurements may be close enough to the previous value to not trigger the alert. However, during an extended attack many measurements will trigger an al ert.

$$slope = \left|\frac{M_1 - M_0}{T_1 - T_0}\right| \quad (5)$$

State based rules #7 and #8 alert for low rates of change. These rules detect low slope measurement injection attacks. Similar to the maximum rate of change, the minimum rate of change is system specific and system experts should be consulted when setting this limit. In most systems no change in a measurement is acceptable. As such rules #7 and #8 do not alert if the calculated slope is 0.

State based rule #9 alerts when a threshold of T consecutive packets is observed with the same process measurement. In the description of state based rule #9 from Table 7, the variable NC is the count of consecutive packets without a measurement change. Two laboratory scale process control systems were used to validate this work; a gas pipeline and an oil storage tank. In both cases the process measurement continuously changes when the system is placed in automatic mode (referred to as AUTO in Table 7). The gas pipeline opens and closes a relief valve to control pressure in a pipeline. The oil storage tank turns a pump on and off as the oil level drops and falls. The state based layer calculates the count of consecutive packets without a process measurement change. The state based layer also stores the current process system mode (AUTO or MANUAL). State based rule #9 only detects constant level injection attacks in system state in which

the process measurement is known to change. For systems where the measurement may legally be constant these rules are not applicable.

A constant level injection attack which starts when the measurement is allowed to be constant may be detected by the rate of change rules (state based rules #6 and #7) if the injected measurement varies significantly from the measurement observed immediately before the attack initiates. Subsequent packets during the attack will not trigger alerts from the rate of change rules.

State based rules #6-#9 calculate the rate of change of a process measurement. Equation 1 shows the method for calculating the rate of change. Equation 1 subtracts the current measurement from the immediately prior measurement when calculating the rate of change. For some systems it may be necessary to adjust this calculation to calculate the rate of change over a smaller or larger time window depending upon the measured variable and the frequency of measurements.

State based rules #10 - #13 monitor the physical process state and alert when the process is in a critical state. Here a critical state is defined as a state of alarm in which the control settings will drive the system further away from a normal system state and more into the alarm state. For laboratory systems used for this work from critical states were defined. First, state based rule #10 alarms if the gas pipeline pressure is above the high level alarm set point (HH) and the pump is on and the relief valve is closed. Second, state based rule #11 alarms if the gas pipeline pressure is below the low level alarm set point (LL) and the pump is off or the relief valve is open. Third, state based rule #12 alarms if the oil storage tank liquid level is above the alarm set point (HH) and the pump is on. Fourth, state based rule #13 alarms if the oil storage tank liquid level is below the alarm

set point (HH) and the pump is off and the system is on. Each of these states should never occur for these physical processes. These IDS rules may alert due to an actual process fault which leads the system to a critical state or may occur due to a cyber-attack driving the process to a critical state via command injection attack. These state based rules will alert for some but not all cases of the altered system control scheme, altered actuator state, continually altered actuator state, altered proportional integral derivative parameter(s), altered control set point command injection attacks. The rules will only alert if the command injection drives the process to a critical state.

Stand-alone rule #8 alerts when a command to set PID parameters to an invalid value is detected. The gas pipeline system uses a PID control to open and close the relief valve to keep the pressure within the high (H) and low (L) set points. PID parameters are set when a system is installed and PID parameters are rarely changed after initial setup. Small changes in PID parameters can lead to drastic changes in system behavior. As such rule #8 alerts for any change in any of the five PID parameters; gain, rate, reset, dead band, and cycle time.

Stand-alone rules #9 and #10 provide limits process for set points. Both the gas pipeline and oil storage systems used for this work include H, L, HH, and LL set points. The H and L set points control the maximum and minimum pressure or liquid level for the gas pipeline or oil storage tank respectively. These values are used by the program running in the PLC connected to the physical processes to determine when to turn on/off the pump for both systems and open close the relief value for the gas pipeline. The HH and LL are alarm set points. For both systems process measurements outside the HH and

LL bounds trigger an alarm. The limits used for stand-alone rules #9 and #10 are set with system knowledge and represent reasonable values for the 4 set points.

The malicious function code injection attacks are detected with stand-alone rules #11-#14.Each rule detects a specific function code and sub-function code combination. Stand-alone rule #13 also matches a portion of the packet payload before alerting.

State based rules #14 detects the invalid cyclic redundancy code (CRC) flood attack. MODBUS-RTU mode uses a 16-bit CRC. MODBUS-ASCII uses an 8-bit longitudinal redundancy code (LRC).MODBUS-TCP frames do not include a CRC. Rather MODBUS-TCP frames depend upon the TCP CRC for error checking. The functions to generate the MODBUS-RTU and MODBUS ASCII CRC and LRC respectively are available in the MODBUS over Serial Line Specification and Implementation Guide [cxi]. Each packet is monitored in the state based layer. The CRC/LRC is calculated within the state based layer and compared to the CRC/LRC with the packet. A count of failed CRC/LRC over a time window is kept. If the number of failed CRC/LRC exceeds a programmable threshold an alert is issued. For this work the time window was 1seconds and the number of failed CRC required to trigger the rules was 2.

Attack #28, MODBUS Slave Traffic Jamming, is detected with two rules. First, one method of implementing the MODBUS Slave Traffic Jamming attack is to transmit a continuous stream of bytes from a slave radio. In this case the contents are irrelevant. To detect this random stream of bytes, stand-alone rule #15 looks for packets of greater than 255 byte length. These are not legal MODBUS packets and should not appear on the line.

In the case that the flood contents are made up of MODBUS packets a count of MODBUS packets in a given time window is used to detect the flood.

Table 14 Signature based IDS Detection Results

Dataset	Gas Pipeline System		Water Storage System	
	Detection Rate	False Positive	Detection Rate	False Positive
Reconnaissance	98.7%	0.0%	98.7%	0.0%
NMRI	95.4%	0.8%	94.2%	0.8%
CMRI	92.5%	0.5%	93.7%	0.6%
MSCI	89.8%	0.7%	90.1%	0.7%
MPCI	93.1%	0.4%	93.0%	0.6%
MFCI	100%	0.0%	100%	0.0%
DOS	100%	0.0%	100%	0.0%
Normal	99.5%	0.3%	99.5%	0.3%

Table 15 Signature based IDS System Usage

Dataset	Gas Pipeline System		Water Storage System	
	CPU Usage	Memory Usage	CPU Usage	Memory Usage
Reconnaissance	0.3%	1.5%	0.3%	1.5%
NMRI	0.3%	1.5%	0.3%	1.5%
CMRI	0.3%	1.5%	0.3%	1.5%
MSCI	0.3%	1.5%	0.3%	1.5%
MPCI	0.3%	1.5%	0.3%	1.5%
MFCI	0.3%	1.5%	0.3%	1.5%
DOS	1.3%	1.5%	1.3%	1.5%
Normal	0.3%	1.5%	0.3%	1.5%

The signature based IDS detection rate for reconnaissance attacks was 98.7% for the gas pipeline control system and 98.7% for the water storage tank control system. A review of the misclassified attack case shows that the signature based IDS failed to detect the malicious packets that contain valid device ID, function code and read/write memory

address. During reconnaissance attack, attackers scan ranges of device address, function code and memory. These ranges contain valid device address, function code and memory address which have been defined in Snort rules. So the Snort will miss these malicious packets and not trigger alarms. The false positive rate for the reconnaissance attacks was 0% for the gas pipeline control system and 0% for the water storage tank control system. This means the signature based IDS does not misclassify the normal traffic or other types of attack as reconnaissance attacks. The Snort rules for reconnaissance attacks mainly focus on the network content such as device ID, function code and memory address. The normal traffic packets and other attack packets will not satisfy these rules.

The signature based IDS detection rate for NMRI attacks was 95.4% for the gas pipeline control system and 94.2% for the water storage tank control system. A review of the misclassified attack case shows that the signature based IDS failed to detect malicious packets that contain gas pressure measurement or water level measurement out of the bound that defined by the Snort rules. During the NMRI attack, when the malicious packet contains a false measurement that is very close to the next valid or true measurement, the malicious packet will not violate the snort rules. The false positive rate for the NMRI attacks was 0.8% for the gas pipeline control system and 0.8% for the water storage tank control system. A review of false positive case shows that the IDS misclassify some normal traffic packets and CMRI packets as NMRI packets. Sometimes the valid measurements may fall in the invalid range of setpoints. These situations will trigger false alarm. For example, the normal water level measurement may be in the range of [HH, H] or [L, LL], which will lead to false positive due to the fact that these ranges are out of the valid bound defined in the Snort rules.

The signature based IDS detection rate for CMRI attacks was 92.5% for the gas pipeline control system and 93.7% for the water storage tank control system. A review of the misclassified attack case shows that the signature based IDS failed to detect malicious packets that contain multi gas pressure measurements or water level measurements that violate the valid system measurements states defined by the Snort rules. State based rules #6-#9 calculate the rate of change of a process measurement. During CMRI attacks, when attackers send simulated normal wave of measurements in the period, the IDS will miss these malicious packets. The false positive rate for the CMRI attacks was 0.5% for the gas pipeline control system and 0.7% for the water storage tank control system. A review of false positive cases shows that the IDS misclassifies some normal traffic packets and NMRI packets as CMRI packets. Sometimes the valid normal measurement wave may trigger false alarms. For example, in the normal water level measurement wave, some of packets may delay or loss, which will lead to the current speed of measurement changing and will increase false positive rate.

The signature based IDS detection rate for MSCI attacks was 89.8% for the gas pipeline control system and 90.1% for the water storage tank control system. A review of the misclassified attack case shows that the signature based IDS failed to detect some packets that contain malicious system state commands. The Snort rules defined ranges of the system parameters. When these crafted parameters are in the Snort pre-defined range, these malicious packets will pass. For example, when the crafted system measurement setting point is in the valid range, it will not trigger alarms. The false positive rate for the MSCI attacks was 0.7% for the gas pipeline control system and 0.7% for the water storage tank control system. A review of false positive case shows that the IDS

misclassifies some normal traffic packets as MSCI packets. Sometimes the valid normal command may trigger false alarm. For example, in the normal water level measurement wave, some of packets may be delayed or lost, which will lead to the current speed of measurement significantly, changed and will lead to false positive.

The signature based IDS detection rate for MPCI attacks was 93.1% for the gas pipeline control system and 93.0% for the water storage tank control system. A review of the misclassified attack case shows that the signature based IDS failed to detect some MPCI packets contain settpoint parameters or the controller parameters values that do not violate the Snort rules. During the MPCI attack, when the malicious packet contains false setpoint parameters values are closed or equal to true setpoint parameters or the controller parameters, the malicious packet will not violate the snort rules. The false positive rate for the MPCI attacks was 0.4% for the gas pipeline control system and 0.6% for the water storage tank control system. A review of false positive cases shows that the IDS misclassify some normal traffic packets as MPCI packets. Sometimes the valid normal command may trigger false alarm. For example, sometimes the operator may set the setpoints in the different range, these setpoints maybe in the Snort rules defined malicious setpoint bound and will lead to false positive.

The signature based IDS detection rate for MFCI attacks was 100% for the gas pipeline control system and 100% for the water storage tank control system. A review of the detection result shows that the signature based IDS can detect this type attack at high detection rate. The false positive rate for the MFCI attacks was 0% for the gas pipeline control system and 0% for the water storage tank control system. The IDS only need to monitor the function code and sub function code in the packet. If the invalid combination

of function and sub function code occurs in the traffic, the IDS will trigger alert. In this research, there are four types of attack defined in the MFCI attack, force listening mode, restart communication, clean communication event log and change ASCII input delimiter.

The signature based IDS detection rate for DOS attacks was 100% for the gas pipeline control system and 100% for the water storage tank control system. A review of the detection result shows that the IDS is very effective when detecting DOS attack. The false positive rate for the DOS attacks was 0% for the gas pipeline control system and 0% for the water storage tank control system. For the DOS attack detection, because MODBUS's maximum payload length limit is 256 bytes, the data logger will store packet with maximum data length. Furthermore, the DOS floods meanness characters in traffic, the MODBUS packet's device address, function code and other information in the packet doesn't contain any useful information. There are many features in dataset to describe this abnormal behavior and classification algorithms can detect this type of attack with high detection rate and low false positive.

This chapter discussed the implementation of a signature based intrusion detection system and an anomaly based intrusion detection system. The signature based intrusion detection system implementation introduces Snort rules used to detect attacks described in Chapter 3. The Snort rules were evaluated in a laboratory setting to demonstrate effectiveness by measuring accuracy. The rules were highly effective at detecting the aforementioned attacks with low false positive rates. Signature based rules are most effective for detecting known attacks. Anomaly based intrusion detection systems can be used to detect unknown attacks. The anomaly based intrusion detection system implementation used a decision tree algorithm to classify network traffic into the classes

described in Chapter 4; normal, Naive Malicious Response Injection (NMRI), Complex Malicious Response Injection (CMRI), Malicious State Command Injection (MSCI), Malicious Parameter Command Injection (MPCI), Malicious Function Command Injection (MFCI), Denial of Service (DOS), and Reconnaissance. Detection results demonstrate that the features in the datasets and a set of predictive features described in this chapter can provide sufficient information to provide a high detection rate with low false positive rate. Because the signature based IDS excel at detecting known attacks while anomaly based IDS excel at detecting unknown attacks using both styles of IDS is recommended for SCADA systems.

CHAPTER VI

CONCLUSION

SCADA control system networks are vulnerable to attacks from external and internal sources. Intrusion detection systems are needed to detect attacks on SCADA networks and targeting SCADA system devices. In order to support intrusion detection system in this area this dissertation provided four contributions.

First, four categories of cyber-attacks (reconnaissance, malicious command injection, malicious response injection, and denial of service) were developed against two laboratory scale serial communication based SCADA systems; gas pipeline system and water storage tank system. In total twenty eight attacks against the serial SCADA systems were implemented and validated. Reconnaissance and denial of service cyber-attacks were also developed against chemical processing SCADA system which uses Ethernet based communication. Three attacks were implemented and validated for the chemical processing SCADA system. Second, this dissertation created a four network dataset which include reconnaissance, malicious command injection, malicious response injection, and denial of service classes of attacks and normal network traffic from SCADA control system networks. The datasets can be used to benchmark SCADA intrusion detection systems. Third, this dissertation implemented a signature based intrusion detection system. The signature based intrusion detection system was evaluated in a laboratory setting to demonstrate effectiveness by measuring accuracy. The signature

based intrusion detection system rules were highly effective at detecting the aforementioned attacks with low false positive rates. In total thirty signature based intrusion detection system rules were created and validated. Fourth, an anomaly based intrusion detection system for the SCADA network was developed. The anomaly based intrusion detection system implementation used a decision tree algorithm to classify network traffic into the normal class and the other attack classes described in aforementioned datasets. The anomaly based IDS also had a high detection rate and low false positive rate. A combined IDS which includes signature based IDS and anomaly based IDS is recommended for SCADA networks. A combined IDS will benefit from higher detection rates of signature based IDS while also benefitting from anomaly based IDS ability to detect previous unknown attacks, aka. zero day vulnerabilities.

This research developed a minimal set of attacks (reconnaissance attack and denial of service attack) against Ethernet based SCADA networks. Many more attacks are possible such as malicious response injection attack and malicious command injection attacks. The IDS research for this dissertation was based soley on the serial based SCADA systems. Datasets for Ethernet based systems should be developed. Additionally, signature based IDS rules should be written for Ethernet based SCADA systems. Finally, anomaly based IDS should be developed for Ethernet based SCADA systems.

REFERENCES

[1] Pacific northwest national laboratory and U.S. department of energy, The role of authenticated communications for electric power distribution," in Proc. Natl. Workshop—Beyond SCADA: Networked Embedded Control for Cyber Physical Systems, Nov. 2006, p. 1.

[2] S. Rinaldi, J. Peerenboom, and T. Kelly. "Identifying, Understanding, and Analyzing Critical Infrastructure Interdependencies," IEEE Control Systems Magazine, IEEE, December 2001, pp. 11-25.

[3] Santorelli, S. Who is looking for your SCADA infrastructure? March 2009. Published online. Sample June 30,2010.

[4] N. Falliere, L. O. Murchu, and E. Chien, W32.Stuxnet Dossier, Symantec Tech. Rep. 1.4, 2011.

[5] J. Slay and M. Miller, "Lessons learned from the maroochy water breach," in Critical Infrastructure Protection, ser. IFIP International Federation for Information Processing, E. Goetz and S. Shenoi, Eds. New York: Springer, 2007, vol. 253, pp. 73–82.

[6] K. Poulsen, "Slammer worm crashed Ohio nuke plant network," 2009 [Online]. Available: http://www.securityfocus.com/news/6767

[7] http://www.nerc.com/page.php?cid=2120

[8] Hadbah, A.; Kalam, A.; Al-Khalidi, H.; , "The subsequent security problems attributable to increasing interconnectivity of SCADA systems," Power Engineering Conference, 2008. AUPEC '08. Australasian Universities , vol., no., pp.1-4, 14-17 Dec. 2008

[9] http://www.modbus.org/

[10] DNP3 Specification Volume 7: IP Networking, DNP User's Group, December 2004.

[11] Ethernet/IP. Ethernet Industrial Protocol (EtherNet/IP). www.ethernet-ip.org, 2007

[12] Fovino, I.N.; Masera, M.; Guidi, L.; Carpi, G.; , "An experimental platform for assessing SCADA vulnerabilities and countermeasures in power plants," Human System Interactions (HSI), 2010 3rd Conference on , vol., no., pp.679-686, 13-15 May 2010

[13] Igor Nai Fovino and Andrea Carcano and Marcelo Masera and Alberto Trombetta, “Design and Implementation of a Secure Modbus Protocol”, Critical Infrastructure Protection III IFIP Advances in Information and Communication Technology, 2009, Volume 311/2009, 83-96

[14] Bagaria, S.; Prabhakar, S.B.; Saquib, Z.; , "Flexi-DNP3: Flexible distributed network protocol version 3 (DNP3) for SCADA security," Recent Trends in Information Systems (ReTIS), 2011 International Conference on , vol., no., pp.293-296, 21-23 Dec. 2011

[15] Munir Majdalawieh, Francesco Parisi-Presicce and Duminda Wijesekera, DNPSec: Disrtibuted Network Protocol Version 3 (DNP3) Security Framework. In Advances in computer, Information, and system Sciences, and Engineering: Proceedings of IETA 2005, pages 227-234. Springer, 2006

[16] Motta Pires, P.S.; Oliveira, L.A.H.G.; , "Security Aspects of SCADA and Corporate Network Interconnection: An Overview," Dependability of Computer Systems, 2006. DepCos-RELCOMEX '06. International Conference on , vol., no., pp.127-134, 25-27 May 2006

[17] Dong Wei; Yan Lu; Jafari, M.; Skare, P.M.; Rohde, K.; , "Protecting Smart Grid Automation Systems Against Cyberattacks," Smart Grid, IEEE Transactions on , vol.2, no.4, pp.782-795, Dec. 2011

[18] B. Schneier, Applied Cryptography, 2nd ed. New York: Wiley, 1996.

[19] W. Mao, Modern Cryptography: Theory and Practice. Upper Saddle River, NJ: Prentice-Hall, 2003.

[20] L. Pietre-Cambacedes and P. Sitbon, “Cryptographic key management for SCADA systems – issues and perspectives,” International Conference on Information Security and Assurance, 2008

[21] R. Dawson, C. Boyd, E. Dawson and JMG. Nieto, “SKMA, A Key Management Architecture for SCADA Systems”, Proc. of the AISW-NetSec workshop, ACM, 2006.

[22] C. Beaver, D. Gallup, W. Neumann & Torgerson M., “Key management for SCADA”, Technical Report SAND2001-3252, Sandia National Laboratories, 2002.

[23] D. Holstein, J. Tengdin, J. Wack, R. Butler, T. Draelos, P. Blomgren, Cyber Security for Utility Operations, NETL Project M63SNL34, Final Report, 2005.

[24] A. Wright, J. Kinast and J. McCarty, Low-latency cryptographic protection for SCADA communications, Proceeding of the Second International Conference on Applied Security and Network Security, pp. 263-277, 2004

[25] Dong-Joo Kang; Hak-Man Kim; , "Development of test-bed and security devices for SCADA communication in electric power system," Telecommunications

Energy Conference, 2009. INTELEC 2009. 31st International , vol., no., pp.1-5, 18-22 Oct. 2009

[26] Keith Stouffer, Joe Falco, Karen Kent, Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security – Recommendations of the National Institute of Standards and Technology, Special Publication 800-82, Initial Public Draft, September 2006

[27] Bonnie Zhu, Anthony Joseph and Shankar Sastry, Taxonomy of Cyber Attacks on SCADA Systems, 2008.

[28] http://www.snort.org/

[29] M. Brundle and M. Naedele "Security for process control systems: An overview", IEEE Security Privacy, vol. 6, no. 6, pp.24 -29 2008

[30] Hong, Sugwon; Lee, Myongho; , "Challenges and Direction toward Secure Communication in the SCADA System," Communication Networks and Services Research Conference (CNSR), 2010 Eighth Annual , vol., no., pp.381-386, 11-14 May 2010

[31] Dong-Joo Kang; Jong-Joo Lee; Seog-Joo Kim; Jong-Hyuk Park; , "Analysis on cyber threats to SCADA systems," Transmission & Distribution Conference & Exposition: Asia and Pacific, 2009 , vol., no., pp.1-4, 26-30 Oct. 2009

[32] Valentine, S.; Farkas, C.;, "Software security: Application-level vulnerabilities in SCADA systems," Information Reuse and Integration (IRI), 2011 IEEE International Conference on , vol., no., pp.498-499, 3-5 Aug. 2011

[33] Chikuni, E.; Dondo, M.; , "Investigating the security of electrical power systems SCADA," AFRICON 2007 , vol., no., pp.1-7, 26-28 Sept. 2007

[34] Dzung, D.; Naedele, M.; Von Hoff, T.P.; Crevatin, M.; , "Security for Industrial Communication Systems," Proceedings of the IEEE , vol.93, no.6, pp.1152-1177, June 2005

[35] Gardner, R.M.; Consortium, G.; , "A Survey of ICT Vulnerabilities of Power Systems and Relevant Defense Methodologies," Power Engineering Society General Meeting, 2007. IEEE , vol., no., pp.1-8, 24-28 June 2007

[36] McDaniel, P.; McLaughlin, S.; , "Security and Privacy Challenges in the Smart Grid," Security & Privacy, IEEE , vol.7, no.3, pp.75-77, May-June 2009

[37] E. J. Byres, M. Franz and D. Miller, The Use of Attack Trees in Assessing Vulnerabilities in SCADA Systems. International Infrastructure Survivability Workshop (IISw'04) Lisbon December 4, 2004.

[38] Queiroz, C.; Mahmood, A.; Tari, Z.; , "SCADASim—A Framework for Building SCADA Simulations," Smart Grid, IEEE Transactions on , vol.2, no.4, pp.589-597, Dec. 2011

[39] Wang Chunlei; Fang Lan; Dai Yiqi; , "A Simulation Environment for SCADA Security Analysis and Assessment," Measuring Technology and Mechatronics Automation (ICMTMA), 2010 International Conference on , vol.1, no., pp.342-347, 13-14 March 2010

[40] Queiroz, C.; Mahmood, A.; Jiankun Hu; Tari, Z.; Xinghuo Yu; , "Building a SCADA Security Testbed," Network and System Security, 2009. NSS '09. Third International Conference on , vol., no., pp.357-364, 19-21 Oct. 2009

[41] Mallouhi, M.; Al-Nashif, Y.; Cox, D.; Chadaga, T.; Hariri, S.; , "A testbed for analyzing security of SCADA control systems (TASSCS)," Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES , vol., no., pp.1-7, 17-19 Jan. 2011

[42] Amin, S.; Litrico, X.; Sastry, S.; Bayen, A. M.; , "Cyber Security of Water SCADA Systems-Part I: Analysis and Experimentation of Stealthy Deception Attacks," Control Systems Technology, IEEE Transactions on , vol.PP, no.99, pp.1, 0

[43] Amin, S.; Litrico, X.; Sastry, S. S.; , .; Bayen, A. M.; , "Cyber Security of Water SCADA Systems-Part II: Attack Detection Using Enhanced Hydrodynamic Models," Control Systems Technology, IEEE Transactions on , vol.PP, no.99, pp.1, 0

[44] Jie Yan; Chen-Ching Liu; Govindarasu, M.; , "Cyber intrusion of wind farm SCADA system and its impact analysis," Power Systems Conference and Exposition (PSCE), 2011 IEEE/PES , vol., no., pp.1-6, 20-23 March 2011

[45] Dillon Beresford, “Exploiting Siemens Simatic S7 PLCs”, Black Hat USA, July 8, 2011

[46] Fleury, Terry; Khurana, Himanshu; Welch, Von,” Towards A Taxonomy Of Attacks Against Energy Control Systems”, Critical Infrastructure Protection II, The International Federation for Information Processing, Volume 290. ISBN 978-0-387-88522-3. Springer US, 2009, p. 71

[47] Mallouhi, M.; Al-Nashif, Y.; Cox, D.; Chadaga, T.; Hariri, S.; , "A testbed for analyzing security of SCADA control systems (TASSCS)," Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES , vol., no., pp.1-7, 17-19 Jan. 2011

[48] Sridhar, S.; Manimaran, G.; , "Data integrity attacks and their impacts on SCADA control system," Power and Energy Society General Meeting, 2010 IEEE , vol., no., pp.1-6, 25-29 July 2010

[49] Yu-Hu. Huang, Alvaro A. Cardenas, et al, ”Understanding the Physical and Economic Consequences of Attacks on Control Systems, Elsevier, International Journal of Critical Infrastructure Protection 2009

[50] Le Xie; Yilin Mo; Sinopoli, B.; , "False Data Injection Attacks in Electricity Markets," Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on , vol., no., pp.226-231, 4-6 Oct. 2010

[51] Y. Liu, M. K. Reiter, and P. Ning, “False data injection attacks against state estimation in electric power grids,” Proceedings of the 16th ACM Conference on Computer and Communications Security, 2009.

[52] Dong Jin; Nicol, D.M.; Guanhua Yan; , "An event buffer flooding attack in DNP3 controlled SCADA systems," Simulation Conference (WSC), Proceedings of the 2011 Winter , vol., no., pp.2614-2626, 11-14 Dec. 2011

[53] Peterson, D., "Quickdraw: Generating Security Log Events for Legacy SCADA and Control System Devices," Conference For Homeland Security, 2009. CATCH '09. Cybersecurity Applications & Technology , vol., no., pp.227-229, 3-4 March 2009

[54] Caswell, B. Bealeand, J., Foster, J. and Faircloth, J. “Snort2.0 Intrusion Detection,” Syngress, Feb. 2003.

[55] Wang, Y., Statistical Techniques for Network Security, Modern Statistically-Based Intrusion Detection and Protection. IGI Global. October 2008.

[56] ZhenWei Yu, Jeffrey J. P. TSAI and Thomas Weigert An adaptive Automatically Tuning Intrusion Detection System. ACM Transactions on Autonomous and Adaptive System, Vol.3, No.3, Article 10, Publication date: August 2008.

[57] Wu Yang, Wei Wan, Lin Guo, Le-Jun Zhang.An Efficient Intrusion Detection Model Based on Fast Inductive learning. 2007 International Conference on Machine Learning and Cybernetics. pp3249-3254, 2007.

[58] K. Wang and S. Stolfo. Anomalous payload-based network intrusion detection. In Proceedings of the International Symposium on Recent Advances in Intrusion Detection (RAID), 2004.

[59] Ahmad, I.; Abdullah, A.B.; Alghamdi, A.S.; , "Evaluating neural network intrusion detection approaches using Analytic Hierarchy Process," Information Technology (ITSim), 2010 International Symposium in , vol.2, no., pp.885-890, 15-17 June 2010

[60] Fan Li, "Hybrid Neural Network Intrusion Detection System Using Genetic Algorithm," Multimedia Technology (ICMT), 2010 International Conference on , vol., no., pp.1-4, 29-31 Oct. 2010

[61] Abdel-Azim, M.; Abdel-Fatah, A.I.; Awad, M.; , "Performance analysis of artificial neural network intrusion detection systems," Electrical and Electronics Engineering, 2009. ELECO 2009. International Conference on , vol., no., pp.II-385-II-389, 5-8 Nov. 2009

[62] Jingwen Tian; Meijuan Gao; Fan Zhang; , "Network Intrusion Detection Method Based on Radial Basic Function Neural Network," E-Business and Information System Security, 2009. EBISS '09. International Conference on , vol., no., pp.1-4, 23-24 May 2009

[63] Lin Li-zhong; Liu Zhi-guo; Duan Xian-hui; , "Network intrusion detection by a hybrid method of rough set and RBF neural network," Education Technology and Computer (ICETC), 2010 2nd International Conference on , vol.3, no., pp.V3-317-V3-320, 22-24 June 2010

[64] Y. Yu, Y. Wei, et al., "anomaly intrusion detection approach using hybrid MLP/CNN neural network," intelligent systems design & applications. ISDA 6th int. conference, Vol.2, issue.16-18, pp: 1095-1102, October, 2006.

[65] W. Wang and R. Battiti, "identifying intrusions in computer networks with principle component analysis," ARES2006 IEEE computer society, pp. 270-279. 2006

[66] L. Vokorokos, A. Baláž, M. Chovanec, "intrusion detection system using self organizing map," Acta Electrotechnica et Informatica No. 1, Vol. 6, pp: 16, 2006

[67] J. Skaruz, "recurrent neural networks on duty of anomaly detection in databases," proceedings of 4th international symposium on neural networks: advances in neural networks part III, pp: 85-94, 2007.

[68] K. Jearanaitanakij, "Classifying Continuous Data Set by ID3 Algorithm", 2005 Fifth International Conference on Information, Communications and Signal Processing, pp. 1048-1051, December 2005.

[69] S. Ruggieri, "Efficient C4.5" IEEE Transactions on Knowledge and Data Engineering (vol. 14, no. 2), pp. 438-444, April 2002.

[70] S. R. Safavian, and D. Landgrebe, "A survey of decision tree classifier methodology", IEEE Transactions on Systems, Man and Cybernetics (vol. 21, no. 3), pp. 660-674, June 1991.

[71] Sheen, S.; Rajesh, R.; , "Network intrusion detection using feature selection and Decision tree classifier," TENCON 2008 - 2008 IEEE Region 10 Conference , vol., no., pp.1-4, 19-21 Nov. 2008

[72] Juan Wang; Qiren Yang; Dasen Ren; , "An Intrusion Detection Algorithm Based on Decision Tree Technology," Information Processing, 2009. APCIP 2009. Asia-Pacific Conference on , vol.2, no., pp.333-335, 18-19 July 2009

[73] Steven Cheung, Bruno Dutertre, Martin Fong, Ulf Lindqvist, Keith Skinner, Alfonso Valdes, "Using Model-based Intrusion Detection for SCADA Networks", SCADA Security Scientific Symposium, 2007

[74] Modbus IDA. Modbus application protocol specification v1.1a, June 4, 2004.

[75] Modbus IDA. Modbus messaging on TCP/IP implementation guide v1.0a, June 4, 2004.

[76] Martin Roesch, Snort - Lightweight Intrusion Detection for Networks, Proceedings of LISA '99: 13th Systems Administration Conference, USENIX

[77] Linda, O.; Vollmer, T.; Manic, M.; , "Neural Network based Intrusion Detection System for critical infrastructures," Neural Networks, 2009. IJCNN 2009. International Joint Conference on , vol., no., pp.1827-1834, 14-19 June 2009

[78] P. 1. Werbos, The Roots ofBackpropagation, New York: Johns Wiley & Sons, 1994.

[79] M. Hagan, M. Menhaj, "Training feedforward networks with the Marquardt algorithm," IEEE Transaction on Neural Networks, vol. 5, no. 6,pp.989-993, 1994

[80] Nessus: http://www.nessus.org.org/nessus/

[81] The Metasploit Project: http://www.metasploit.com/home

[82] Dayu Yang, Alexander Usynin, and J. Wesley Hines, Anomaly-Based Intrusion Detection for SCADA Systems, International Atomic Energy Agency (IAEA), Technical Meeting on Cyber Security, Idaho, 2006

[83] A. Wald, Sequential Analysis. New York, NY: John Wiley & Sons, 1947

[84] Kenny Gross, Keith Whisnant, Aleksey Urmanov, Kalyan Valdyanathan, Sajjit Thampy, Continuous System Telemetry Harness, Tech. Rep., [Online] Available: http://research.sun.com/sunlabsday/docs.2004/talks/1.03 Gross.pdf, 2005.

[85] Keith Whisnant, Kenny Gross, Natasha Lingurovska, Proactive Fault Monitoring in Enterprise Servers, in Proceedings of the 2005 International Conference on Computer Design, pp. 3-10, June 2005.

[86] Paul Oman, Matthew Phillips, Intrusion Detection and Event Monitoring in SCADA Networks, book chapter of Critical Infrastructure Protection, Pages 161-173, Springer Boston, 2007

[87] A. Carcano, A. Coletta, M. Guglielmi, M. Masera, I.N. Fovino, and A. Trombetta, "A Multidimensional Critical State Analysis for Detecting Intrusions in SCADA Systems," IEEE Transactions on Industrial Informatics, Vol.7, No.2, May 2011, pp.179-186.

[88] Yichi Zhang; Lingfeng Wang; Weiqing Sun; Green, R.C.; Alam, M.; , "Distributed Intrusion Detection System in a Multi-Layer Network Architecture of Smart Grids," Smart Grid, IEEE Transactions on , vol.2, no.4, pp.796-808, Dec. 2011

[89] S. R. Gunn, “Support vector machines for classification and regression,” Univ. Southampton, Faculty of Eng., Sci., Math.; School Electron. Comput. Sci., Tech. Rep., 1998.

[90] The KDD99 dataset [Online]. Available: http://kdd.ccs.uci.edu/databases/kddcup99/task.html

[91] Valdes, A.; Cheung, S.; , "Communication pattern anomaly detection in process control systems," Technologies for Homeland Security, 2009. HST '09. IEEE Conference on , vol., no., pp.22-29, 11-12 May 2009

[92] A. Valdes, "Detecting Novel Scans Through Pattern Anomaly Detection", Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX III), Volume 1, pp. 140-151.

[93] O. Linda, M. Manic, M. McQueen, "Improving Control System Cyber-State Awareness using Known Secure Sensor Measurements ," in Proc. 7th International Conference on Critical Information Infrastructure Security, CRITIS 2012, Lillehammer, Norway, Sep. 17-18, 2012

[94] Julian Rrushi and Kyoung-Don Kang, "Detecting Anomalies in Process Control Networks", Critical Infrastructure Protection III, IFIP Advances in Information and Communication Technology, Volume 311. ISBN 978-3-642-04797-8. Springer Berlin Heidelberg, 2009, p. 151

[95] D.Hosmer and S.Lemeshow, "Applied Logistic Regression", Wiley, Hoboken, New Jersey, 2000.

[96] de Sousa, Mário, Jirı Baum, Andrey Romanenko, and Pólo II Pinhal de Marrocos. "MatPLC: Towards Real-Time Performance." In Proceedings of the 2003 Real-time Linux Workshop. 2003.

[97] J. Larsen, SCADA security, presented at Blackhat DC, 2008

[98] Chee-Wooi Ten; Junho Hong; Chen-Ching Liu; , "Anomaly Detection for Cybersecurity of the Substations," Smart Grid, IEEE Transactions on , vol.2, no.4, pp.865-873, Dec. 2011

[99] C.-W. Ten, G. Manimaran, and C.-C. Liu, "Cybersecurity for critical infrastructures: Attack and defense modeling," IEEE Trans. Syst., Man, Cybern. A, Syst., Humans, vol. 40, no. 4, pp. 853–865, Jul. 2010.

[100] Morris T., Srivastava A. Reaves 8., Pavurapu K., Abdelwahed S., Vaughn R., McGrew W., Dandass Y. Engineering Future Cyber Physical Energy Systems: Challenges, Research Needs, and Roadmap. 2009 IEEE North American Power Symposium. October 4-6, 2009,Starkville, MS

[101] Offensive Security. The Exploit Database. http://www.exploit-db.com/ (Accessed July 22, 2013)

[102] Reaves, B.; Morris, T., "Discovery, infiltration, and denial of service in a process control system wireless network," eCrime Researchers Summit, 2009. eCRIME '09. , vol., no., pp.1,9, Sept. 20 2009-Oct. 21 2009

[103] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

[104] Wang, Y.-C., Joshi, M., Rose, C. P., Fischer, F., Weinberger, A., & Stegmann, K. (2007). Context Based Classification for Automatic Collaborative Learning Process Analysis. Artificial Intelligence in Education 2007.

[105] J. E. Gewehr, M. Szugat, and R. Zimmer. BioWeka - extending the weka framework for bioinformatics. Bioinformatics , 23(5):651–653, 200

[106] Domenico Talia, Paolo Trunfio, Oreste Verta, "The Weka4WS framework for distributed data mining in service-oriented Grids". Concurrency and Computation: Practice and Experience, vol. 20, n. 16, pp. 1933--1951, Wiley InterScience, November 2008

[107] S. Celis and D. R. Musicant. Weka-parallel: machine learning in parallel. Technical report, Carleton College, CS TR, 2002

[108] Shekhar R. Gaddam, Vir V. Phoha, Kiran S. Balagani, "K-Means+ID3: A Novel Method for Supervised Anomaly Detection by Cascading K-Means Clustering and ID3 Decision Tree Learnimg Methods", IEEE Transactions on Knowledge and Data Engineering, Vol. 19, No. 3, March 2007

[109] Ruggieri S., Efficient C4.5, IEEE Transactions On Knowledge And Data Engineering, Volume: 14, Issue: 2, March-April 2002, pp.438-444

[110] Waheed T, Bonnell RB, Prasher SO, et al. Measuring performance in precision agriculture: CART - A decision tree approach. Agricultural Water Management, 2006(84):173-185

[111] www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf