Chapter 5

INDUSTRIAL CONTROL SYSTEM TRAFFIC DATA SETS FOR INTRUSION DETECTION RESEARCH

Thomas Morris and Wei Gao

Abstract Supervisory control and data acquisition (SCADA) systems monitor and control physical processes associated with the critical infrastructure. Weaknesses in the application layer protocols, however, leave SCADA networks vulnerable to attack. In response, cyber security researchers have developed myriad intrusion detection systems. Researchers primarily rely on unique threat models and the corresponding network traffic data sets to train and validate their intrusion detection systems. This leads to a situation in which researchers cannot independently verify the results, cannot compare the effectiveness of different intrusion detection systems, and cannot adequately validate the ability of intrusion detection systems to detect various classes of attacks. Indeed, a common data set is needed that can be used by researchers to compare intrusion detection approaches and implementations. This paper describes four data sets, which include network traffic, process control and process measurement features from a set of 28 attacks against two laboratory-scale industrial control systems that use the MODBUS application layer protocol. The data sets, which are freely available, enable effective comparisons of intrusion detection solutions for SCADA systems.

Keywords: Industrial control systems, SCADA, intrusion detection, MODBUS

1. Introduction

Supervisory control and data acquisition (SCADA) systems are computer-based process control systems that control and monitor remote physical processes. SCADA systems are strategically important because they are widely used in the critical infrastructure. Several incidents and cyber attacks affecting SCADA systems have been documented; these clearly illustrate the vulnerability of critical infrastructure assets. The reported incidents demonstrate that cyber attacks against SCADA systems can have severe financial impact

J. Butts and S. Shenoi (Eds.): Critical Infrastructure Protection VIII, IFIP AICT 441, pp. 65–78, 2014.

Table 1. Intrusion detection systems by threat model and network protocol.

System	**Threat Model**	**Protocol**
SRI Modbus [2]	Access, reconnaissance and attack	MODBUS
NNIDSCI [8]	Traffic from Nmap, Nessus, Metasploit	–
AKKR-SPRT [16]	DoS attacks simulated by Sun servers	SNMP
IDAEM [10]	RTU attacks	–
Multidimensional CSA [1]	Simulated attacks on critical states	MODBUS
SGDIDS [17]	KDD 99 Cup Data Set	–
Pattern Detection [15]	Reconnaissance	MODBUS
KSSM [7]	False data injection	–
Statistical Estimation [12]	Overflow exploits	MODBUS
RAIM [14]	File system and status modification	C37.118

and can result in damage that is harmful to humans and the environment. In 2000, a disgruntled engineer compromised a sewage control system in Maroochy Shire, Australia, causing approximately 264,000 gallons of raw sewage to leak into a nearby river [13]. In 2003, the Slammer worm caused a safety monitoring system at the Davis-Besse nuclear plant in Oak Harbor, Ohio to go offline for approximately five hours [11]. The insidious Stuxnet worm [3], which was discovered in 2010, targeted nuclear centrifuge system controllers, modifying system behavior by distorting monitored process information and altering control actions.

Cyber security researchers have developed numerous intrusion detection systems to detect attacks against SCADA systems. Much of the research uses training and validation data sets created by the same researchers who developed the intrusion detection systems. Indeed, no standardized data set containing normal SCADA network traffic and attack traffic is currently available to researchers. In order to evaluate the performance of data mining and machine learning algorithms for SCADA intrusion detection systems, a network data set used for benchmarking intrusion detection system performance is sorely needed. This paper describes four data sets, which include network traffic, process control and process measurement features from a set of 28 attacks against two laboratory-scale industrial control systems that use the MODBUS application layer protocol. The data sets, which are freely available, enable effective comparisons of intrusion detection solutions for SCADA systems.

2. Related Work

Several SCADA security researchers have developed intrusion detection systems that monitor network traffic and detect attacks against SCADA systems. Table 1 lists example intrusion detection systems, the threat models they use and the network protocols they analyze. Note that each intrusion detection system uses a unique threat model. Some threat models are based on attacks executed against SCADA laboratory testbeds while others are based on manipulated data sets drawn from other domains. The network protocols also differ; MODBUS is the most common protocol (used in three systems) while the IEEE C37.118 protocol is used in just one system. The remaining systems use threat models with attacks implemented at different network layers.

A noticeable drawback of the research identified in Table 1 is that the threat models only include subsets of attack classes. Not surprisingly, exploit coverage is limited for each of the data sets. Only a few of the threat models consider reconnaissance attacks while some models only include response injection attacks. Indeed, the malicious behavior captured in the data sets is neither consistent nor comprehensive in terms of normal operations and attacks. For this reason, it is difficult to judge the effectiveness of an intrusion detection system against sophisticated attacks. This also leads to a situation in which researchers cannot independently verify intrusion detection results and cannot compare the performance of intrusion detection systems.

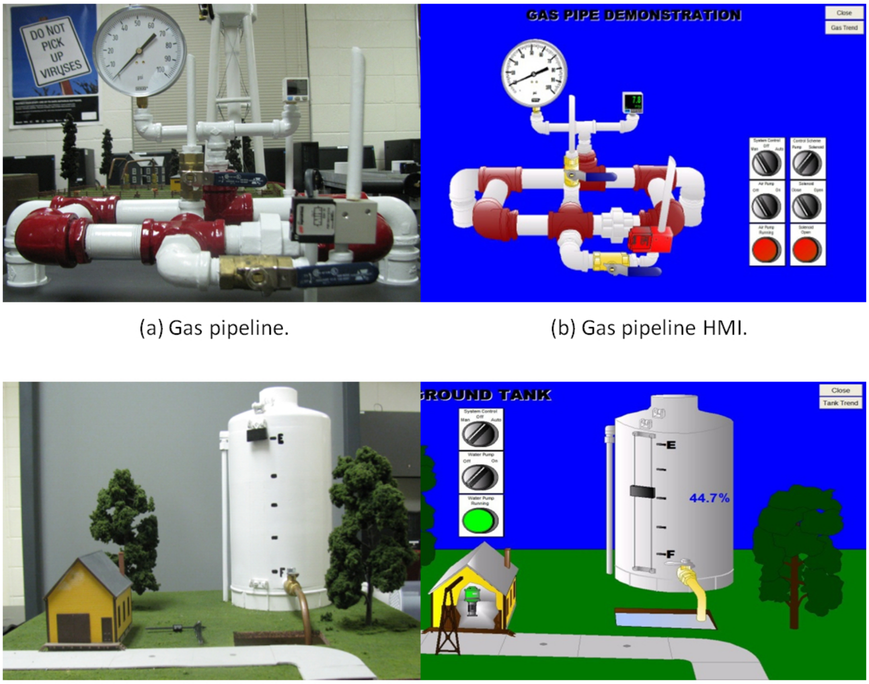
3. Test Bed Description

The data sets described in this paper were captured using a network data logger, which monitored and stored MODBUS traffic from a RS-232 connection. Two laboratory-scale SCADA systems were used: a gas pipeline and water storage tank.

Figure 1 shows the gas pipeline and water storage tank systems along with the associated human machine interfaces (HMIs). The gas pipeline system includes a small airtight pipeline connected to a compressor, a pressure meter and a solenoid-controlled relief valve. The pipeline system attempts to maintain the air pressure in the pipeline using a proportional integral derivative (PID) control scheme.

The water storage tank system includes a tank that holds approximately two liters of water, a manually-operated relief valve to deplete water from the tank, a pump to add water to the tank from an external water source and a meter to measure the water level as percentage of tank capacity. The water storage tank uses an on/off control scheme to maintain the water level between the high (H) and low (L) setpoints. The water storage tank activates an alarm when the water level is above the high alarm setpoint (HH) or below the low alarm setpoint (LL). Detailed descriptions of the functionality of the two systems and their respective components are provided in a separate paper [9].

A bump-in-the-wire approach was used to capture data logs and to inject attacks. The device was implemented via a C program running on a VMware virtual machine. The virtual machine included two RS-232 serial ports connected to a USB-to-serial converter. The C program monitored each serial port for traffic. Detected traffic was timestamped and recorded in a log file. To facilitate attacks, the C program incorporated hooks to inject, delay, drop and alter network traffic.



(a) Gas pipeline.

(b) Gas pipeline HMI.

(c) Water storage tank.

(d) Water storage tank HMI.

Figure 1. Gas pipeline and water storage tank systems.

4. Description of Attacks

The data sets presented in this paper include network traffic, process control and process measurement features from normal operations and attacks against the two SCADA systems. The attacks are grouped into four classes: (i) reconnaissance; (ii) response injection; (iii) command injection; and (iv) denial-of-service (DoS).

4.1 Reconnaissance Attacks

Reconnaissance attacks gather SCADA system information, map the network architecture and identify device characteristics (e.g., manufacturer, model number, supported network protocols, device address and device memory map). The reconnaissance class of attacks in the data set includes four attacks against MODBUS servers: address scan, function code scan, device identification attack and points scan. The address scan discovers SCADA servers connected to a network by polling for responses from different MODBUS addresses. The function code scan identifies supported MODBUS function codes that can be used by an identified server. The device identification attack allows an attacker to obtain device vendor information, product code and major and minor firmware revisions. The points scan allows the attacker to build a memory map of MODBUS coils, discrete inputs, holding registers and input registers.

4.2 Response Injection Attacks

SCADA 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. State information is provided to a human machine interface for monitoring the process, storing process measurements in a data historian and providing feedback to control loops that measure process parameters and take the appropriate control actions based on the process state. Response injection attacks alter responses from the server to client, providing false system state information.

Response injection attacks are divided into naive malicious response injection (NMRI) attacks and complex malicious response injection (CMRI) attacks. NMRI attacks leverage the ability to inject or alter response packets in a network; however, they lack the ability to obtain information about the underlying process being monitored and controlled. Eight NRMI attacks were used in creating the data sets described in this paper. The naive read payload size attack returns a malicious response with the correct payload size but sets the payload to all zeros, ones or random bits. The invalid read payload size attack returns a malicious response with a length that does not conform to the requested length. The invalid exception code attack returns false error responses to the client after a read command. The negative sensor measurements attack injects negative process measurements; this is problematic because many systems use floating point numbers to represent values that can only be positive. The sensor measurements grossly out-of-bounds attack injects process measurements that are significantly outside the bounds of alarm setpoints. The sporadic sensor measurement injection attack sends false process measurements outside the bounds of the H and L control setpoints while staying within the alarm setpoint range specified by HH and LL. The random sensor measurement injection attack sends random process measurements of gas pipeline pressure or water tank water level.

CMRI attacks attempt to mask the actual state of the physical process and negatively affect feedback control loops. They are more sophisticated than NMRI attacks because they require an in-depth understanding of the targeted system. As such, CMRI attacks are designed to appear like normal process functionality. These attacks can be used to mask alterations to process state perpetrated by malicious command injection attacks. CMRI attacks are more difficult to detect because they project a state of normalcy.

Five CMRI attacks were used to create the data sets. The constant sensor measurement injection attack repeatedly sends malicious packets containing the same measurement to mask the real state of the system. The calculated sensor measurement injection attack sends pre-calculated process measurements. The high frequency measurement injection attack increases the rate of change of a process measurement beyond its normal range. The low frequency measurement injection attack decreases the rate of change of a process measurement below its normal range. A replayed measurement injection attack resends process measurements that were previously sent from the server to a client.

4.3 Command Injection Attacks

Command injection attacks inject false control and configuration commands to alter system behavior. The potential impacts of malicious command injections include loss of process control, interruption of device communications, unauthorized modification of device configurations and unauthorized modification of process setpoints. Command injection attacks are divided into malicious state command injection (MSCI) attacks, malicious parameter command injection (MPCI) attacks and malicious function code command injection (MFCI) attacks. Comprehensive descriptions of these attacks are provided in [4].

MSCI attacks change the state of the process control system to drive the system from a safe state to a critical state by sending malicious commands to remote field devices. MSCI attacks may involve a single injected command or multiple injected commands. Three MSCI attacks were used to create the data sets. The altered system control scheme attack changes the control mode from automatic to manual and then turns on the compressor or pump to increase the pressure in the pipeline or raise the water level in the water storage tank, respectively. The altered actuator state attack changes the state of an actuator in a system. In the case of the gas pipeline system, this attack includes command injections that turn the compressor on or off, and those that open or close the relief valve; in the case of the water storage tank system, the altered actuator state attack turns the pump on or off. The continuous altered actuator state attack repeatedly changes the actuator states in a system. For example, command packets could be continually transmitted to switch the state of the compressor and pump in the pipeline and storage tank systems, respectively. Additionally, a continuous altered actuator state attack may be used to repeatedly transmit MODBUS write register commands to invert the state of the solenoid that controls the relief valve in the gas pipeline system.

MPCI attacks alter programmable logic controller (PLC) field device setpoints. The data sets include two MPCI attacks. The altered control setpoint attack changes the H and L setpoints for the water storage tank while disabling the liquid level alarms. A proportional integral derivative (PID) controller is commonly used in SCADA systems to maintain a desired setpoint by calculating and adjusting for system error; the altered proportional integral derivative parameter attack changes the PID parameters used in the gas pipeline system.

MFCI attacks use built-in protocol functions in a manner different from what was intended. The data sets include four MFCI attacks. The force listen only mode attack causes a MODBUS server to stop transmitting on the network. The restart communications attack sends a command that causes the MODBUS server to restart, leading to a temporary loss of communications. The clear communications event log attack erases the communications event log of the

MODBUS server. Finally, the change ASCII input delimiter attack changes the delimiter used for MODBUS ASCII devices.

4.4 Denial-of-Service Attacks

Denial-of-service attacks target communications links and system programs in an attempt to exhaust resources. The data sets include two denial-of-service attacks. The invalid cyclic redundancy code (CRC) attack injects a large number of MODBUS packets with incorrect CRC values into a network. The MODBUS master traffic jamming attack uses a non-addressed slave address to continually transmit random data to random destination addresses.

5. SCADA Traffic and Payload Data Sets

The KDD Cup 1999 Data Set [6] was developed to train and validate intrusion detection systems associated with traditional information technology systems. The use of this common data set by numerous researchers facilitated the independent validation of research results and the comparison of many intrusion detection system approaches. In the area of SCADA security, however, researchers develop their own data sets to test intrusion detection systems because there is a lack of availability and access to SCADA network traffic. Indeed, no standard data set is available that includes normal and attack traffic for a SCADA network that can serve as a benchmark to evaluate and compare SCADA intrusion detection system performance. This section describes a data set that is intended to provide researchers with a common platform to evaluate the performance of data mining and machine learning algorithms designed for SCADA intrusion detection systems. The data set includes different classes of attacks that cover a variety of SCADA system attack scenarios.

The common data set described in this paper has three primary benefits. First, not all researchers have access to SCADA equipment to generate their own data sets; a common data set would enable more researchers to work in the area of SCADA security. Second, a common data set would allow researchers to independently validate the results of other researchers. Third, a common data set would enable the comparison of the performance of different algorithms, leading to better intrusion detection systems.

5.1 Data Set Organization

The data sets created as a result of this research effort are stored in the Attribute Relationship File Format (ARFF) for use with the WEKA software [5]. WEKA is a comprehensive framework that enables researchers to compare and verify machine learning algorithms.

The organization of the MODBUS data set is similar to that of the KDD Cup 1999 Data Set [6]. Each instance in the data set represents one captured network transaction pair (e.g., merged MODBUS query and response). An instance includes network traffic information and the current state of the process

Table 2. Data sets.

Data Set	Index
Data Set I	Gas pipeline system complete data set
Data Set II	Water storage tank system complete data set
Data Set III	Gas pipeline system reduced (10%) data set
Data Set IV	Water storage tank system reduced (10%) data set

control system based on payload content. Note that each instance contains a label identifying it as normal MODBUS traffic or as attack traffic with the designated attack class.

Four data sets were created as part of this research. Table 2 provides the descriptions of the four data sets. Data Set I contains transactions from the gas pipeline system. Data Set II contains transactions from the water storage tank system. The two data sets were generated from network flow records captured with a serial port data logger.

Two reduced size data sets were also created. Data Set III is a gas pipeline system data set, which was created by randomly selecting 10% of the instances in Data Set I. Likewise, Data Set IV is a water storage tank system data set, which was created by randomly selecting 10% of the instances in Data Set II. The two reduced data sets minimize memory requirements and processing time when validating classification algorithms. They are intended for applications for which quick feedback is desired.

Two categories of features are present in the data sets: network traffic features and payload content features. Network traffic features describe the communications patterns in SCADA systems. Compared with traditional enterprise networks, SCADA network topologies and services are relatively static. Note that some attacks against SCADA systems may change network communications patterns. As such, network traffic features are used to describe normal traffic patterns in order to detect malicious activity. Network traffic features include the device address, function code, length of packet, packet error checking information and time intervals between packets. Payload content features describe the current state of the SCADA system; they are useful for detecting attacks that cause devices (e.g., PLCs) to behave abnormally. Payload content features include sensor measurements, supervisory control inputs and distributed control states.

5.2 Network Traffic Features

Table 3 lists the ten attributes that comprise the network traffic features. The first and second attributes are the command device address and response device address. Note that the MODBUS serial command address is one byte long, with each server having a unique device address. As such, the command and response device addresses should match during normal operations. An address mismatch is an indicator of a reconnaissance attack. MODBUS serial

Table 3. Attacks on MODBUS systems.

Attribute	**Description**
command_address	Device ID in command packet
response_address	Device ID in response packet
command_memory	Memory start position in command packet
response_memory	Memory start position in response packet
command_memory_count	Number of memory bytes for R/W command
response_memory_count	Number of memory bytes for R/W response
command_length	Total length of command packet
response_length	Total length of response packet
time	Time interval between two packets
crc_rate	CRC error rate

systems are configured so that all the slave devices (servers) see all the master transactions. Each slave must check the device address to discern the intended recipient before acting on a packet. Based on the system configuration, the set of device addresses that a slave device should encounter is fixed; device addresses not specified in the configuration are anomalous.

The command memory, response memory, command memory count and response memory count include internal memory addresses and field sizes for read and write commands. The memory of a MODBUS server is grouped into data blocks called coils, discrete inputs, holding registers and input registers. Coils and discrete inputs represent a single, read-only Boolean bit with authorized values of `0x00` and `0xFF`. Holding and input registers are 16-bit words; holding registers are read/write capable while input registers are read only. Each data block may have its own set of contiguous address space or the data blocks may share a common memory address space based on vendor implementation. The command memory and response memory features are coil or register read/write start addresses taken from command and response packets, respectively. The command and response memory count features are the numbers of objects to be read and written, respectively.

The command and response packet length features provide the lengths of the MODBUS query and response frames, respectively. The MODBUS protocol data unit (PDU) is limited to 253 bytes with an additional three bytes for device ID and CRC fields, resulting in a 256-byte packet. In the gas pipeline and water storage tank systems, the master repeatedly performs a block write to a fixed memory address followed by a block read from a 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. Note, however, that many of the described attacks have different packet lengths. As such, the packet length feature provides a means to detect many attacks.

The time interval attribute is a measurement of the time between a MODBUS query and its response. The MODBUS protocol is a request-response protocol and the time interval varies only slightly during normal operations.

Table 4. List of common payload attributes.

Feature Name	**Description**
comm_fun	Value of command function code
response_fun	Value of response function code
sub_function	Value of sub-function code in the command/response
measurement	Pipeline pressure or water level
control_mode	Automatic, manual or shutdown
pump_state	Compressor/pump state
manual_pump_setting	Manual mode compressor/pump setting
label	Manual classification of the instance

The malicious command injection, malicious response injection and DOS attacks often result in significantly different time interval measurements due to the nature of the attacks.

The last attribute is the command/response CRC error rate. This attribute measures the rates of CRC errors identified in command and response packets. Because SCADA network traffic patterns are relatively static, the normal command and response CRC error rates are expected to stay somewhat constant. In a normal system, the error rates should be low; however, the rates are expected to increase when a system is subjected to a denial-of-service attack such as the invalid CRC attack.

5.3 Payload Content Features

The payload content features differ for the gas pipeline and water storage system data sets due to different control schemes and different measured variables. The attributes common to both systems are listed in Table 4. During normal operations, the response function code matches the command function code if there is no error. If there is an error, the response sub-function code is the command function code value plus `0x80`. The measurement attribute provides the current value of the gas pipeline pressure or water tank level. The naive malicious response injection attack and the complex malicious response injection attack influence process measurements by manipulating the expected values. The system control mode is determined based on data in a command packet. The system control mode can place the system in the shutdown, manual or automatic modes; zero represents the shutdown mode, one represents the manual mode and two represents the automatic mode. A malicious state command injection attack can attempt to modify the system operating mode or shut down the system. The gas pipeline system/water storage tank system use a compressor/pump to add air/water, respectively, to maintain the desired setpoint. If the compressor/pump state has a value of one, then the compressor/pump is on; if it is zero, the compressor/pump is off. When a system is in the automatic mode, the PLC logic controls the compressor/pump state. A malicious complex response injection attack may modify this value in

Table 5. Unique features of the gas pipeline system data sets.

Feature Name	**Description**
set_point	Target pressure in the gas pipeline
control_scheme	Control scheme of the gas pipeline
solenoid_state	State of solenoid used to open the gas relief valve
gain	Gain parameter value of the PID controller
reset	Reset parameter value of the PID controller
dead_band	Dead band parameter value of the PID controller
rate	Rate parameter value of the PID controller
cycletime	Cycle time parameter value of the PID controller

order to mask the actual compressor/pump working state. Note that, in the manual mode, the compressor/pump state is controlled by the manual compressor/pump setting value. A malicious state command injection attack may change the compressor/pump mode continually or intermittently.

Table 5 shows the eight attributes that are specific to the gas pipeline system. The initial attribute identifies the setpoint for the nominal gas pressure. The second attribute identifies the operating mode of the system. In the automatic mode, the PLC logic attempts to maintain the gas pressure in the pipeline using a PID control scheme by selecting if the compressor or the relief valve is activated. If the control scheme is zero, then the compressor is activated to increase pressure; if the control scheme is one, then the relief valve is activated using a solenoid to decrease the pressure. In the manual mode, the operator controls the pressure by sending commands to start the compressor or open the relief valve. Additionally, there are five attributes related to the PID controller. The gain, reset, dead band, rate and cycle time impact PID controller behavior and should be fixed during system operation. A malicious parameter command injection attack tries to modify these parameters to interrupt normal control operations.

Table 6. Unique features of the water storage system data sets.

Feature Name	**Description**
HH	Value of HH setpoint
H	Value of H setpoint
L	Value of L setpoint
LL	Value of LL setpoint

Table 6 shows the four attributes that are specific to the water storage tank system: HH, H, L and LL. In the automatic mode, the PLC logic maintains the water level between the L and H setpoints using an on/off controller scheme. When the sensors detect that the water level has reached the L level, the PLC logic turns the water pump on. Alternatively, when the sensors determine that

Table 7. Instance classification values.

Label Name	**Label Value**	**Label Description**
Normal	0	Instance is not part of an attack
NMRI	1	Naive malicious response injection attack
CMRI	2	Complex malicious response injection attack
MSCI	3	Malicious state command injection attack
MPCI	4	Malicious parameter command injection attack
MFCI	5	Malicious function command injection attack
DoS	6	Denial-of-service attack
Reconnaissance	7	Reconnaissance attack

the water level has reached the H level, the PLC logic turns the water pump off. Note that the water storage tank includes a manual drainage valve that allows water to drain out of the tank when the valve is open. If the manual drainage valve is open, the water level in the tank oscillates between the H and L setpoints continuously as the pump cycles on and off to compensate. When the manual drainage valve is closed, the pump stays on until the water level reaches the H setpoint, at which point it turns off and maintains a constant level. Due to a system fault, if the water level rises to the HH setpoint or falls to the LL setpoint, then an alarm is triggered at the human machine interface that monitors the water storage tank. In the manual mode, the pump state is controlled manually by the human machine interface (i.e., an operator can manually activate and deactivate the pump).

Table 7 lists the eight possible label values. Recall that each data set instance is labeled as normal or according to its attack class. The labeling scheme was chosen to match the KDD Cup 1999 Data Set [6], which identified attacks by class. Note that specific attacks in each attack class have similar exploit methods and similar impact on the SCADA system.

5.4 Discussion

The data sets described in this paper are relevant to other SCADA systems – systems that use protocols other than MODBUS as well as systems other than gas pipelines and water storage tanks. The features in the data sets are divided into two groups in a similar manner as SCADA protocols divide packets into network traffic related fields and content fields. Indeed, other protocols include similar, albeit not identical, network traffic information such as addresses, function codes, payloads and checksums. Additionally, most SCADA protocols tend to adhere to query-response traffic patterns similar to MODBUS. The content features in the data sets include remote commands and system states similar to how other types of systems monitor and update system settings. As such, the data sets provide a framework to measure the accuracy of intrusion detection approaches designed for a variety of SCADA systems.

6. Conclusions

Researchers have developed numerous intrusion detection approaches for detecting attacks against SCADA systems. To date, researchers have generally engaged unique threat models and the associated network traffic data sets to train and validate their intrusion detection systems. This leads to a situation where researchers cannot independently verify the results of other research efforts, cannot compare the effectiveness of intrusion detection systems against each other and ultimately cannot adequately judge the quality of intrusion detection systems.

The four data sets developed in this research include network traffic, process control and process measurement features from two laboratory-scale SCADA systems. Data Set I contains transactions from a gas pipeline system while Data Set II contains transactions from a water storage tank system. The data sets were generated from network flow records captured with a serial port data logger in a laboratory environment. A set of 28 attacks was used to create the data sets; the attacks were grouped into four categories: reconnaissance, response injection, command injection and denial-of-service attacks. Reduced size data sets corresponding to Data Sets I and II were also created. Data Set III is a gas pipeline system data set containing 10% of the instances in Data Set I while Data Set IV is a water storage tank system data set containing 10% of the instances in Data Set II. The four data sets comprising normal and attack traffic can be used by security researchers to compare different SCADA intrusion detection approaches and implementations.

References

[1] A. Carcano, A. Coletta, M. Guglielmi, M. Masera, I. Fovino and A. Trombetta, A multidimensional critical state analysis for detecting intrusions in SCADA systems, *IEEE Transactions on Industrial Informatics*, vol. 7(2), pp. 179–186, 2011.

[2] S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner and A. Valdes, Using model-based intrusion detection for SCADA networks, *Proceedings of the SCADA Security Scientific Symposium*, 2007.

[3] N. Falliere, L. O'Murchu and E. Chien, W32.Stuxnet Dossier, Version 1.4, Symantec, Mountain View, California, 2011.

[4] W. Gao, Cyber Threats, Attacks and Intrusion Detection in Supervisory Control and Data Acquisition Networks, Ph.D. Dissertation, Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, Mississippi, 2014.

[5] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. Witten, The WEKA data mining software: An update, *ACM SIGKDD Explorations*, vol. 11(1), pp. 10–18, 2009.

[6] S. Hettich and S. Bay, The UCI KDD Archive, Department of Information and Computer Science, University of California at Irvine, Irvine, California (`kdd.ics.uci.edu`), 1999.

[7] O. Linda, M. Manic and M. McQueen, Improving control system cyber-state awareness using known secure sensor measurements, *Proceedings of the Seventh International Conference on Critical Information Infrastructures Security*, pp. 46–58, 2012.

[8] O. Linda, T. Vollmer and M. Manic, Neural network based intrusion detection system for critical infrastructures, *Proceedings of the International Joint Conference on Neural Networks*, pp. 1827–1834, 2009.

[9] T. Morris, A. Srivastava, B. Reaves, W. Gao, K. Pavurapu and R. Reddi, A control system testbed to validate critical infrastructure protection concepts, *International Journal of Critical Infrastructure Protection*, vol. 4(2), pp. 88–103, 2011.

[10] P. Oman and M. Phillips, Intrusion detection and event monitoring in SCADA networks, in *Critical Infrastructure Protection*, E. Goetz and S. Shenoi (Eds.), Springer, Boston, Massachusetts, pp. 161–173, 2008.

[11] K. Poulsen, Slammer worm crashed Ohio nuke plant network, *SecurityFocus*, Symantec, Mountain View, California (`www.securityfocus.com/news/6767`), August 19, 2003.

[12] J. Rrushi and K. Kang, Detecting anomalies in process control networks, in *Critical Infrastructure Protection III*, C. Palmer and S. Shenoi (Eds.), Springer, Heidelberg, Germany, pp. 151–165, 2009.

[13] J. Slay and M. Miller, Lessons learned from the Maroochy water breach, in *Critical Infrastructure Protection*, E. Goetz and S. Shenoi (Eds.), Springer, Boston, Massachusetts, pp. 73–82, 2008.

[14] C. Ten, J. Hong and C. Liu, Anomaly detection for cybersecurity of substations, *IEEE Transactions on Smart Grid*, vol. 2(4), pp. 865–873, 2011.

[15] A. Valdes and S. Cheung, Communication pattern anomaly detection in process control systems, *Proceedings of the IEEE Conference on Technologies for Homeland Security*, pp. 22–29, 2009.

[16] D. Yang, A. Usynin and J. Hines, Anomaly-based intrusion detection for SCADA systems, presented at the *IAEA Technical Meeting on Cyber Security of Nuclear Power Plant Instrumentation and Control and Information Systems*, 2006.

[17] Y. Zhang, L. Wang, W. Sun, R. Green and M. Alam, Distributed intrusion detection system in a multi-layer network architecture of smart grids, *IEEE Transactions on Smart Grid*, vol. 2(4), pp. 796–808, 2011.