Industrial Control System Simulation and Data Logging for Intrusion Detection System Research

Thomas H. Morris
Distributed Analytics and Security Institute
Mississippi State University
Starkville, MS, USA
morris@ece.msstate.edu

Zach Thornton
Distributed Analytics and Security Institute
Mississippi State University
Starkville, MS, USA
zach@dasi.msstate.edu

Ian Turnipseed
Distributed Analytics and Security Institute
Mississippi State University
Starkville, MS, USA
ian@dasi.msstate.edu

ABSTRACT

Industrial control system intrusion detection is a popular topic of research for several years, and many intrusion detection systems (IDS) have been proposed in literature. IDS researchers lack a common framework to train and test proposed algorithms. This leads to an inability to properly compare proposed IDS and limits research progress. This paper documents 2 approaches to data sharing for the industrial control system IDS research community. First, a network traffic data log captured from a gas pipeline is presented. The gas pipeline data log was captured in a laboratory and includes artifacts of normal operation and cyber-attacks. Second, an expandable virtual gas pipeline is presented which includes a human machine interface, programmable logic controller, Modbus/TCP communication, and a Simulink based gas pipeline model. The virtual gas pipeline provides the ability to model cyber-attacks and normal behavior. IDS solutions can overlay the virtual gas pipeline for training and testing.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General – *Security and protection (e.g., firewalls)*.

General Terms

Security

Keywords

Industrial Control System, SCADA, Intrusion Detection System, Cyber Security.

1. INTRODUCTION

Supervisory Control and Data Acquisition (SCADA) systems are computer-based process control systems that interconnect and monitor remote physical processes. SCADA systems have a strategic importance due to the fact that they are adopted by the critical infrastructure of nations. Any damage to critical infrastructure may have an 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. The Stuxnet [1] worm targeted industrial control systems and altered system behavior at the client and server level. A disgruntled engineer penetrated a sewage control system in Maroochi Australia and caused approximately 264,000 gallons of raw sewage leak in to nearby rivers [2]. Finally, in 2003, the Davis-Besse nuclear plant in Oak Harbor Ohio was attacked by the Slammer Worm which caused a safety monitoring system of the plant to go offline for approximately five hours [3].

IDS researchers need tools and data to facilitate research. First, IDS researchers commonly use data logs which include attack and normal artifacts from a system to train and test classifiers used to detect cyber-attacks. The 1999 DARPA dataset produced by MIT's Lincoln Labs was created with the intent for researchers to test viable Intrusion Detection Systems (IDS) for effectiveness. The dataset has been a vital part in furthering research for evaluating computer network IDSs and providing a benchmark for other researchers to compare and validate results, but the dataset was found to contain unintended patterns that led algorithms to easily learn differences between scenarios [4]. Currently, no commonly shared data logs exist for industrial control system IDS research. As a result researchers commonly develop a set of cyber-attacks against a locally owned system, capture data logs, and then train and test their IDS. These data logs are not commonly shared which makes comparison of proposed IDS difficult. This paper presents a set of data logs captured from a laboratory scale gas pipeline system described in [5]. The data logs include labeled network transactions during normal operation and during 35 cyber-attacks. Second, in addition to data logs, IDS researchers need a common platform to model industrial control systems and cyber-attacks against these systems. This paper describes a virtual gas pipeline built using Python. The virtual pipeline which includes a human machine interface (HMI), a virtual physical process, a virtual programmable logic controller (PLC), and a virtual network. The behavior of the virtual pipeline is compared to a laboratory scale gas pipeline. In addition to modeling the gas pipeline, the underlying components of the virtual pipeline constitute a platform for modeling other industrial control systems. Because the modeled systems are virtual, there is no physical limit on the size of modelled systems. As such, the platform allows modelling of industrial control systems at scale.

The rest of this paper is organized as follows. Section 2 describes the gas pipeline data logs and section 3 describes the virtual gas pipeline. Finally, conclusions and future work are provided.

2. GAS PIPELINE DATA LOGS

The data logs described in this paper are a second iteration of previous data logs described by Morris et al. in [6] which were found to contain unintended patterns. These unintended patterns cause machine learning algorithms to build a model which does not match real system behavior and leads to overly optimistic classification accuracy. For this work, a new test bed architecture was developed which randomizes system state by making period control changes from the HMI, randomizes attack order, and randomizes attack attributes. This new randomization minimizes the presence of unintended patterns in the data logs.

Modbus packets include header and payload. For Modbus over Serial Line a packet includes a device address, function code, payload, and a cyclic redundancy code (CRC) or linear redundancy code (LRC). Modbus/TCP packets include a Modbus Application Protocol header (MBAP) header, function code, and payload. The MBAP header includes a transaction identifier, protocol identifier, length, and device identifier. The device identifier is similar to the Modbus over Serial Line address. The data logs described in this work are taken from a Modbus over Serial Line, however, they can be safely used as proxy for Modbus/TCP data with the exception that there is no transaction identifier, protocol identifier, and length field. The transaction identifier is generally a count of transactions. The protocol identifier is always 0 for legal Modbus/TCP packets, and the length is the number of bytes in the payload plus 1 byte for the function code.

Inside the payload Modbus/TCP and Modbus over Serial Line packets are identical. Modbus read and write commands are the most common command types. Read and write payload includes additional packet attributes such as coil or register addresses, quantities of requested or returned coils or registers, coil or register contents, error codes, and exception codes. Some exceptional commands, such as the Diagnostic, file record access, mask write, and read FIFO commands include sub function codes, and other attributes to describe specific queries and responses.

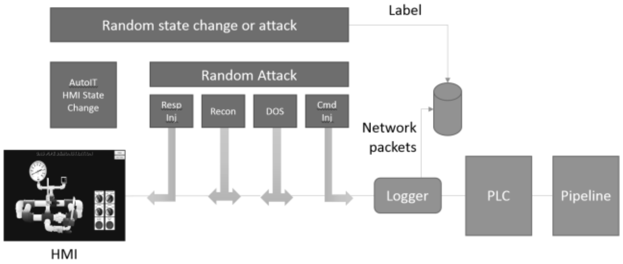


Figure 1: Test Bed Architecture

SCADA systems have very regular communication patterns. Often the same limited set of read and write commands will be repeated in a loop. For example, the gas pipeline system used for this work repeats the same two commands in a loop. First, it writes the contents of all registers and coils used for control. Next, a Modbus read holding register command is used to read the measured state of the system. These two commands are each followed by a response. This regularity leads to a set of commands in which all device addresses are constant, each of the 4 packets always have the same length, and each of the 4 packets always have the same function code. This lack of variation is expected. These regular patterns can be exploited by machine learning algorithms which build a model of normal behavior and detect abnormal deviations. In general deviations in header attribute values and attributes which describe payload contents are indicative of an attack.

For a given system read and write command payload describes coils and register contents to monitor and control the system. This information effectively represents the intended control state of the measured system. This state information should change to represent normal variation of control and measured system state.

shows the architecture of the test bed used to collect data logs. The test bed first randomly chooses whether to change system control state or to initiate a cyber-attack. An AutoIt automation and scripting language [7] script was written to interact with the test bed HMI software. When a control state change was selected, the AutoIt script randomly chooses from multiple legal system states and initiates the mouse clicks in the HMI to effect the state change. The gas pipeline includes a system control mode input with 3 states; off, manual control, or automatic control. In automatic mode the control scheme controls pressure by turning a pump on or off or by opening and closing a relief valve using a solenoid. In automatic mode, a proportional integral derivative (PID) controller is used to control the pump or solenoid depending upon the control scheme chosen. Six PID parameters can be set from the HMI; pressure set point, gain, reset rate, rate, dead band, and cycle time. In manual mode, the HMI can be used to manually change the pump state and manually control the relief valve state by opening or closing the solenoid. When the test bed chooses to change the gas pipeline control state, the AutoIt script chooses a legal combination of system control mode, control scheme, and PID set points. The gas pipeline has one physical limitation. The pump must periodically rest to allow it to cool. This is not normal for gas pipelines, but, is required for the lab system. Because of this, the AutoIt script forced the pump to have a 20% duty cycle. This results in periods of the pump being off in the data logs.

Table 1: Cyber-attacks 1-12

Attack Name	Number	Type	Description
Setpoint Attacks	1-2	MPCI	Changes the pressure set point outside and inside of the range of normal operation.
PID Gain Attacks	3-4	MPCI	Changes the gain outside and inside of the range of normal operation.
PID Reset Rate Attacks	5-6	MPCI	Changes the reset rate outside and inside of the range of normal operation.
PID Rate Attacks	7-8	MPCI	Changes the rate outside and inside of the range of normal operation.
PID Deadband Attacks	9-10	MPCI	Changes the dead band outside and inside of the range of normal operation.
PID Cycle Time Attacks	11-12	MPCI	Changes the cycle time outside and inside of the range of normal operation.

When the AutoIt script chooses to execute a cyber-attack, a random attack is chosen from 4 categories; response injection, reconnaissance, denial of service (DOS), and command injection. 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. 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. Response injection attacks alter the response from server to client and thereby provide false system state information. NMRI attacks lack sophistication. NMRI attacks leverage the ability to inject or alter response packets in the network, but, lack information about the process being monitored and controlled.

CMRI attacks add a level of sophistication above that of the NMRI attacks. CMRI require more 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. CMRI attacks are designed to appear like normal process functionality. These attacks can be used to mask other process changes. Because these attacks project a state of normalcy they are more difficult to detect.

Table 2: Cyber-attacks 13-23

Attack Name	Number	Type	Description
Pump Attack	13	MSCI	Randomly changes the state of the pump.
Solenoid Attack	14	MSCI	Randomly changes the state of the solenoid.
System Mode Attack	15	MSCI	Randomly changes the system mode.
Critical Condition Attacks	16-17	MSCI	Places the system in a Critical Condition. This condition is not included in normal activity.
Bad CRC Attack	18	DOS	Sends MODBUS packets with incorrect CRC values. This can cause denial of service.
Clean Registers Attack	19	MFCI	Cleans registers in the slave device.
Device Scan Attack	20	Recon	Scan for all possible devices controlled by the master.
Force Listen Attack	21	MFCI	Forces the slave to only listen.
Restart Attack	22	MFCI	Restart communication on the device.
Read Id Attack	23	Recon	Read ID of slave device. The data about the device is not recorded, but is performed as if it were being recorded.

Command injection attacks inject false control and configuration commands into a control system to alter system behavior. The potential impacts of malicious command injections include loss of process control, interruption of device communications, unauthorized modification of device configuration, and unauthorized modification of process set points. Malicious State Command Injection (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 require a single injected command or multiple injected commands. Malicious Parameter Command Injection (MPCI) attacks alter PLC set points. Malicious Function Code Injection (MFCI) attacks transmit commands which misuse protocol network parameters to alter network behavior. Denial of Service (DOS) attacks target communication links or attempt to disable programs running on system endpoints which control the system, log data, and govern communications.

In total 35 cyber-attacks were used in creation of the data logs. Tables 1-3 provide brief descriptions of the cyber-attacks. In total the data log contains records from 214,580 Modbus network packets with 60,048 packets associated with a cyber-attack. In the data logs, each packet is labeled with the attack number shown in Tables 1-3 or the label 0 for packets associated with a normal event (no attack).

Table 3: Cyber-attack 24-35

Attack Name	Number	Type	Description
Function Code Scan Attack	24	Recon	Scans for possible functions that are being used on the system. The data about the device is not recorded, but is performed as if it were being recorded.
Rise/Fall Attacks	25-26	CMRI	Sends back pressure readings which create trends on the pressure reading's graph.
Slope Attacks	27-28	CMRI	Randomly increases/decreases pressure reading by a random slope.
Random Value Attacks	29-31	NMRI	Random pressure measurements are sent to the master.
Negative Pressure Attack	32	NMRI	Sends back a negative pressure reading from the slave.
Fast Attacks	33-34	CMRI	Sends back a high set point then a low setpoint which changes "fast"
Slow Attack	35	CMRI	Sends back a high setpoint then a low setpoint which changes "slow"

The data logs were stored in two formats; raw and Attribute Relationship File Format (ARFF). Raw data log entries include raw packet contents in Modbus ASCII format, attack category, specific attack, source, destination, and a time stamp. The raw data is parsed into individual features in the ARFF data log. Table 4 lists all features in the ARFF data logs. The data logs are available by request from the author.

3. VIRTUAL GAS PIPELINE

Fundamental risks to SCADA systems can be identified and detected with research into the patterns, attack vectors, and impacts related to malicious activity. Traditionally, such research has required access to a real control system or a test bed environment that includes a scaled physical model and accompanying hardware, software, and information communications technologies (ICT) which form the complete cyber physical system. Such lab scale test beds, as used to create the previous ICS data logs, present two limitations for researchers. First, only researchers with physical access to the test bed can engage in SCADA intrusion detection research. Second, such test beds are expensive, difficult to expand, and difficult to maintain.

A virtual SCADA laboratory was developed as a potential solution to these difficulties. The virtual SCADA laboratory is portable, distributable, and expandable. The virtual SCADA laboratory closely models commercial SCADA products, is able to communicate with commercial SCADA products, is easily expandable, and is run in a virtual computing environment. The model of the physical process, initially a curve fit of a physical lab process, lacked fidelity to the original model in all operating modes. The original laboratory, with improvements to the physical process model using Simulink, is described below.

3.1 Virtual Pipeline Components

The virtual pipeline consists of 4 components; a virtual process, a PLC simulation, a network simulation, and an HMI. The virtual process, the PLC, and HMI all run in a separate virtual machines (VM). The virtual PLC and the HMI communicate via Modbus/TCP over a virtual network provided by the VM platform.

Table 4: Features in ARFF Data Log

Attribute	Description
address	The station address of the MODBUS slave device. This address is the same on a query and response to a given slave device.
function	MODBUS function code.
length	The length of the MODBUS packet.
setpoint	The pressure set point when the system is in the Automatic system mode.
gain	PID gain.
reset rate	PID reset rate.
deadband	PID dead band.
cycle time	PID cycle time.
rate	PID rate.
system mode	The system’s mode automatic (**2**), manual (**1**), or off (**0**).
control scheme	The control scheme is either pump (**0**) or solenoid (**1**). This determines which mechanism is used to regulate the set point.
pump	Pump control; on (**1**) or off (**0**). Only used in manual mode.
solenoid	Relief valve control; opened (**1**) or closed (**0**). Only used in manual mode.
pressure measurement	Pressure measurement.
command response	Command (**1**) or response (**0**).
time	Time stamp.
binary result	Binary class; attack (**1**) or normal (**0**).
Attack category	Category of attack (**0-7**).
specific result	Specific attack (**0-35**)

3.2 Process Simulation

Simulation of the physical processes forms the base component of this laboratory. In real world control systems, the process itself is typically a physical/chemical/mechanical phenomenon which must be measured and controlled. The process is usually described by complex sets of multi-order differential equations. To simplify process simulation Matlab Simulink was used to model the gas pipeline, a pump, and a relief valve. The SimHydraulics Simulink package was used to model pipeline components [8].

The virtual system models a pump, a valve, a pipeline, a fluid, and fluid flow. For the modeled physical system, the source of fluid (in the field of fluid dynamics, air is considered a fluid [9]) was air from the compressor. Because Simulink requires a source, and open air is not an option, a reservoir of non-descript fluid was chosen as the source of the fluid. There is also a valve between the valve controlled by the virtual PLC and the return reservoir. This valve is used to simulate load and changes position at least once a second. shows the design of the system in Simulink.

Not pictured in is the interface to the PLC which models sensor and actuator connections to the physical process. Simulink contains libraries for communicating using UDP packets. JSON attribute-value pairs are sent between the virtual PLC and Simulink process simulation to model connections between a real PLC and actuators and sensors. The process simulation implements separate ports for each actuator and sensor.

3.3 PLC Simulation

A central component of the virtual pipeline is the simulation of the PLC hardware and software. In real world systems, a typical PLC controller is programmed to perform 4 steps in an infinite loop: read inputs, analyze current state, calculate responses, and write outputs. This process is what the PLC simulation seeks to emulate.

Almost all programming in PLCs is written in a language known as Ladder Logic [10]. The virtual PLC simulates the behavior of PLC ladder logic via a Python routine. Each data read, calculation, and output setting takes place one at a time, emulating each rung of a ladder logic program.

The virtual PLC communicates with other networked devices using Modbus/TCP by utilizing the modbus_tk Python libraries. This enables the virtual PLC to communicate with external devices including physical PLCs and commercial HMI using a standard SCADA communication protocol and allows researchers and students to view, capture, analyze, and route the traffic just as in a real SCADA system. The implementation of Modbus/TCP within this simulation is indistinguishable from Modbus traffic of real SCADA devices. Both Wireshark and Snort classify the simulated network traffic as Modbus/TCP packets.

3.4 Human Machine Interface (HMI)

The third critical component of the virtual pipeline is the human machine interface (HMI) which is used to remotely monitor and control the physical process.

Two separate HMIs were used. The first HMI, shown in Figure 2, uses GE iFix [11] software. This HMI is the same set of HMI screens used for the modeled physical process. Since the virtual pipeline exactly models the sensors, controls, and PLC ladder logic from the physical system, no changes were required to use the iFix HMI.

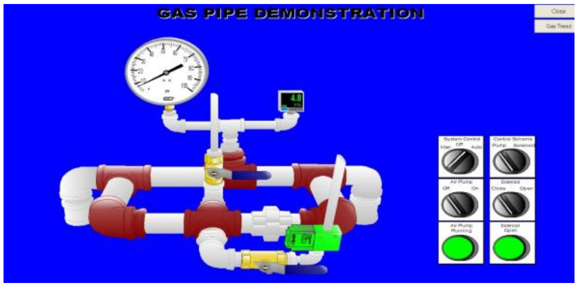
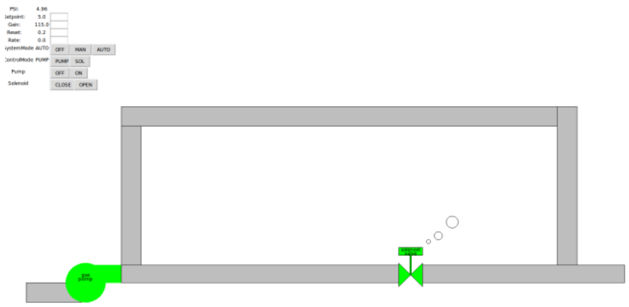


Figure 2: GE iFix HMI

The GE iFix software is proprietary, and therefore not distributable by the authors. As such, a second HMI was developed using the Python TkInter libraries for sharing with other researchers and educators. Figure 3 shows below the TkInter HMI.



PSI: 4.96
Setpoint: 5.0
Gain: 115.0
Reset: 0.2
Rate: 0.0
SystemMode AUTO
ControlMode PUMP
Pump
Solenoid

Figure 3: TkInter HMI

Both HMI are functionally equivalent. Both provide the same control inputs and display the same system measurements.

3.5 Results

The virtual pipeline was compared to the physical model. Normal, startup, and attack behaviors were compared and contrasted.

3.5.1 Normal Operation

Figure 4 and Figure 5 show the physical and virtual pipelines regulating pressure around a set point using the relief valve control scheme. The pressure set point of both systems is 15 PSI, the time range is 8 minutes, and the pressure scale on the graph is 0-25 PSI.

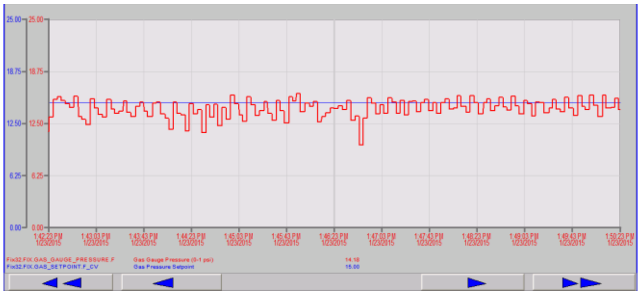


Figure 4: Physical Pipeline Pressure Regulation

As can be seen, the pressure changes in the virtual pipeline simulation resemble the physical pipeline. The frequency of pressure change in the virtual system is faster than the real system. The behavior differences occur because the modeled pump size, the relief valve size, and the diameter and length of the pipe all differ between the two systems. Since both the physical pipeline and virtual pipeline are simulated pipelines it was not important to exactly match behaviors between the two simulations. Both accurate model real gas pipelines for the purposes of intrusion detection system and SCADA security research.

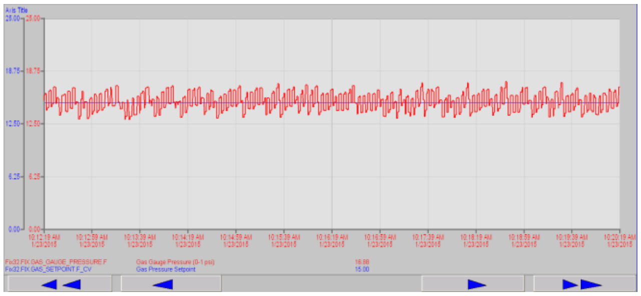


Figure 5: Virtual Pipeline Pressure Regulation

3.5.2 Startup Operation

Figure 6 plots pipeline pressure during startup for both the physical and virtual pipeline. During startup both pipeline models incrementally gain pressure with a stair step pattern. The stair step pattern is an artifact of the measurement frequency. The slope of the virtual pipeline pressure is higher than the physical mode. This results from a difference in modeled pump size, relief valve size, and pipeline diameter and length.

3.5.3 Attack Operation

Command Injection attacks are a category of attacks used by attackers to maliciously adjust settings within a SCADA system. One way to implement such an attack is to impersonate a SCADA client, inject a command into the system by sending the command to the server, and modify settings such as pressure set point, PID parameters, or relief valve control state.

One such attack is an Altered Control Set Point attack. In this attack, the attacker purports to be a MODBUS device with a unique MODBUS device number, acts as a client, and sends a command to the server to alter the set point of the system. The server perceives this command as an authentic, alters the set point, and begins adjusting the process actuators to achieve the new set point.

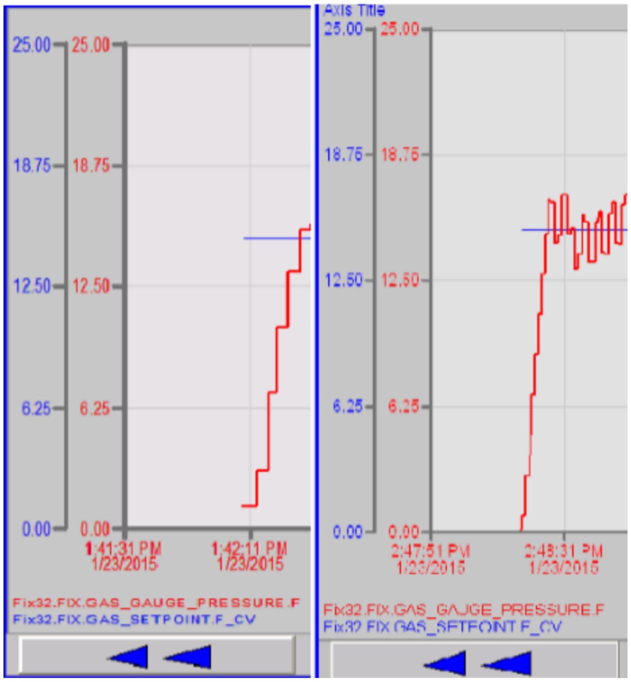


Figure 6: Physical and Virtual Pipeline Pressure during System Startup

In Figure 7, the blue line plots the pressure set point and the red line plots the measured pressure. The set point is increased twice, then decreased, then increased. The measured pressure follows the set point. This attack was not compared to the physical system. The pump in the physical system is too small to achieve similar pressures.

4. CONCLUSIONS

This paper provides an overview of two tools useful for industrial control system IDS researchers. First, a set of labeled network data logs captured while a laboratory scale gas pipeline was in normal states and under cyber-attack is described. The data logs include artifacts of 35 cyber-attacks and can be used to train and test classifiers used by IDS. The data logs are available from the authors of this paper and will facilitate comparison of different IDS implementations. Second, a virtual gas pipeline is described. The virtual gas pipeline includes a HMI, PLC, virtual physical process, and network connections. The virtual pipeline behaves similarly to a laboratory scale pipeline and uses the industry standard Modbus/TCP network protocol.

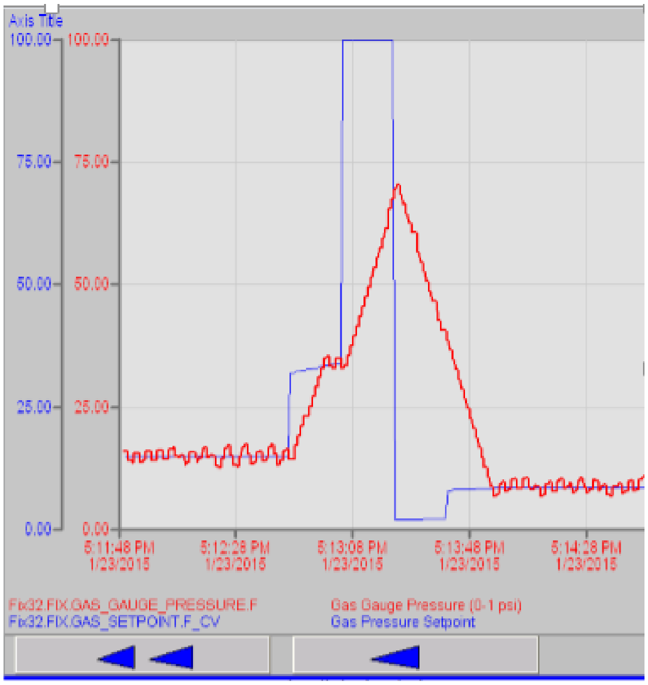


Figure 7: Virtual Pipeline Responding to Command Injection Attack

The virtual pipeline components can be used to model many types of industrial control systems. One future work is to model an interstate gas pipeline to allow investigation of IDS performance and to enable cyber-attack impact studies at scale. Another future work is to map the 35 cyber-attacks presented in the data log section of this work to a virtual interstate gas pipeline system and capture data logs from the larger system.

5. ACKNOWLEDGMENTS

This work was supported both by the National Science Foundation Secure and Trustworthy Cyberspace program under Grant No. 1315726, and by the Pacific Northwest National Laboratory, under U.S. Department of Energy Contract DE-AC05-76RL01830.

6. REFERENCES

[1] N. Falliere, L. O'Murchu, and E. Chien, W32.Stuxnet Dossier, Symantec Technical Report 1.4, 2011.

[2] J. Slay and M. Miller, Lessons Learned From the Maroochy Water Breach, Critical Infrastructure Protection, eds. E. Goetz and S. Shenoi, New York: Springer, vol. 253, pp. 73–82, 2007

[3] K. Poulsen, Slammer Worm Crashed Ohio Nuke Plant Network. http://www.securityfocus.com/news/6767. 2009

[4] McHugh, J. Testing Intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory, ACM Transactions on Information and System Security (TISSEC), v.3 n.4, p.262-294, Nov. 2000

[5] Morris, T. Srivastava, A., Reaves, B., Gao, W., Pavurapu, K., Reddi, R. A Control System Testbed to Validate Critical Infrastructure Protection Concepts. International Journal of Critical Infrastructure Protection (2011). Elsevier. doi:10.1016/j.ijcip.2011.06.005

[6] Morris, T., Gao, W., Industrial Control System Network Traffic Data sets to Facilitate Intrusion Detection System Research, Critical Infrastructure Protection VIII, Sujeet Shenoi and Johnathan Butts, Eds. IFIP Advances in Information and Communication Technology, Springer Berlin Heidelberg, Volume 441, 2014, pp 65-78.

[7] Brand, J., Balvanz, J. Automation is a Breeze with AutoIt. Proceedings of the 33rd Annual ACM SIGUCCS Conference on User Services, pp. 12-15. 2005.

[8] Dabney, James B., and Thomas L. Harman. *Mastering Simulink 4*. Prentice Hall PTR, 2001.

[9] Hutchinson, J., The Physics of Flight, University of California Museum of Paleontology, January 1996.

[10] Pollard, J. Ladder Logic Remains the PLC Language of Choice, *Control Engineering* vol. 41, no. 5, pp. 77-79, 1994.

[11] Cong-Jiang, L., Control System of GE iFix and SIMATIC PLC in Alkali-callback Evaporator, *Light Industry Machinery* 1, 2008.