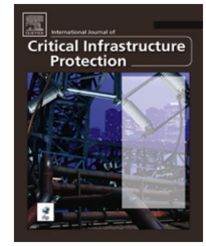
Available online at www.sciencedirect.com

SciVerse ScienceDirect

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



A control system testbed to validate critical infrastructure protection concepts

Thomas Morris[a,*], *Anurag Srivastava*[b], *Bradley Reaves*[a], *Wei Gao*[a], *Kalyan Pavurapu*[a], *Ram Reddi*[a]

[a] *Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, Mississippi 39762, USA*
[b] *School of Electrical Engineering and Computer Science, Washington State University, Pullman, Washington 99164, USA*

ARTICLE INFO

Article history:
Received 3 January 2011
Received in revised form
19 June 2011
Accepted 20 June 2011
Published online 29 June 2011

Keywords:
Testbed
Industrial control system
SCADA
Smart grid
Cybersecurity

ABSTRACT

This paper describes the Mississippi State University SCADA Security Laboratory and Power and Energy Research laboratory. This laboratory combines model control systems from multiple critical infrastructure industries to create a testbed with functional physical processes controlled by commercial hardware and software over common industrial control system routable and non-routable networks. Laboratory exercises, functional demonstrations, and lecture material from the testbed have been integrated into a newly developed industrial control system cybersecurity course, into multiple other engineering and computer science courses, and into a series of short courses targeted to industry. Integration into the classroom allows the testbed to provide a workforce development function, prepares graduate students for research activities, and raises the profile of this research area with students. The testbed enables a research process in which cybersecurity vulnerabilities are discovered, exploits are used to understand the implications of the vulnerability on controlled physical processes, identified problems are classified by criticality and similarities in type and effect, and finally cybersecurity mitigations are developed and validated against within the testbed. Overviews of research enabled by the testbed are provided, including descriptions of software and network vulnerability research, a description of forensic data logger capability developed using the testbed to retrofit existing serial port MODBUS and DNP3 devices, and a description of intrusion detection research which leverages unique characteristics of industrial control systems.

1. Introduction

University, government, and industry based researchers have begun to develop testbed solutions that model industrial control systems to support cybersecurity research and development in this domain. The majority of testbeds documented in the literature model control systems and their associated networks using various simulation technologies. This paper describes a university based testbed, which includes commercial hardware and software and includes functioning physical processes to model contemporary control systems found in critical infrastructure. Pedagogical uses of the testbed are described to demonstrate the utility of the testbed to educate students from a cross-section of disciplines in the challenges and methods of industrial control system cybersecurity and to raise the profile of this research

* *Corresponding author.* Tel.: +1 662 325 3199; fax: +1 662 325 2298.
E-mail addresses: morris@ece.msstate.edu (T. Morris), asrivast@eecs.wsu.edu (A. Srivastava), bgr39@ece.msstate (B. Reaves), wg135@ece.msstate.edu (W. Gao), kp240@ece.msstate.edu (K. Pavurapu), rr370@ece.msstate.edu (R. Reddi).

1874-5482/$ - see front matter
doi:10.1016/j.ijcip.2011.06.005

problem amongst students. Finally, the paper provides abbreviated overviews of research enabled by the testbed to demonstrate the effectiveness of a testbed with commercial hardware and software controlling functional physical processes.

The testbed includes laboratory scale control systems from multiple critical industries including, petrochemical manufacturing, gas pipeline operation, electricity transmission, factory systems, steel manufacturing, and heating ventilating and air conditioning (HVAC) in the form of 7 industrial control systems built with commercially available hardware and software. The control systems include remote terminal units (RTU), programmable logic controllers (PLC), sensors, actuators, and human machine interface (HMI) software, commonly found in industrial critical infrastructure applications. Each system controls a functional laboratory scale physical process. The physical processes include a water storage tank (which also models a petroleum storage tank), a raised water tower, a factory conveyor belt, a gas pipeline, an industrial blower, and a steel rolling operation, and a smart grid transmission control system.

A remote connection exists to a second facility on the university campus which houses electric transmission substation devices and control center software and systems. The electric transmission substation and control center facility contains protection relays, phasor measurement units, phasor data concentrators, a synchrophasor vector processor, programmable logic controllers, a substation GPS clock, a Real Time Digital Simulator, OSISoft PI Historian. Wireshark is used in the power lab to capture and store network traffic at all layers including industrial control system specific application layer protocols MODBUS, DNP3, GOOSE, and IEEE C37.118. Wireshark enables capturing data during normal operation and while systems are under simulated attack without affecting either the operation of the control system or the simulated attack. The power lab includes a MU-4000 Analyzer, a network tester capable of performing denial of service testing, network congestion testing, protocol mutation testing, and tests against commonly known network vulnerabilities.

The combined testbed is used to support university based research and development in support of identifying existing industrial control system vulnerabilities, developing vulnerability taxonomies to identify common cybersecurity deficiencies in need of solutions development, and to serve as a platform for validating research cybersecurity solutions which serve industry and government. Research and development in the two labs has primarily included software and network cybersecurity. Specific software related projects include laboratory research to identify common human machine interface (HMI) vulnerabilities. HMI software provides means for operators to remotely monitor and control physical processes. HMI software may reside on a computer housed within a physical and electronic security perimeter or may access processes through remote connections over the internet using VPN or over secure dial-up solutions. HMI software vulnerability analysis has identified multiple vulnerabilities related to insecure password storage, and privilege escalation. Specific network related projects include laboratory research to discover communications vulnerabilities related to the use of wireless systems in industrial control systems, vulnerability analysis of the Serial MODBUS and Serial DNP3 protocols, development of a tool to log Serial MODBUS and Serial DNP3 network traffic for post-incident forensic analysis, and industrial control system intrusion detection system research.

The testbed enables research by allowing researchers first to investigate cybersecurity vulnerabilities on functional control systems. Researchers typically implement exploits and attack the systems in the testbed to understand the implications of the vulnerability. Vulnerabilities can be ranked by criticality and classified by similarities in type and effect. Researchers then develop cybersecurity mitigations for vulnerabilities which are implemented and validated against the previously generated exploits using the testbed.

The combined testbed is also used for pedagogical purposes. First, a graduate course has been developed to teach industrial control system cybersecurity concepts. This class is available to students across multiple disciplines including all fields of engineering, computer science, and management information systems. The testbed described in this paper has been used to support development of classroom material, to provide live demonstrations of control systems in use, to provide live demonstrations of control systems under cyber attack, and for laboratory exercises for students. Second, concepts and experiences learned from research activities using the testbed have been integrated into multiple other classes. Classes which added material related to the SCADA testbed include Data Communications and Computer Networking, Software Engineering Senior Design, Introduction to Software Engineering, Information and Computer Security, Operating Systems, Power System Operation and Control, Power System Modeling and Simulation, and Cryptography and Network Security. Finally, researchers are currently developing material for a series of workforce development short courses. The electric utility industry currently has an aging workforce and is working to replace retiring workers. Additionally, the emergence of the Smart Grid is changing the business practices at utilities. Both these factors combine to create a need for short courses to train existing utility employees and prepare new employees.

The body of this paper includes a section discussing related industrial control system testbeds, detailed descriptions of the laboratory scale control systems available in the testbed, a section on the pedagogical impact of the testbed, a section on cyber security research conducted with the testbed, followed by a discussion of future works and conclusions. A glossary defining abbreviations used in this paper is provided in Table 1.

2. Related works

The Idaho National Labs (INL) National SCADA Testbed Program is a large scale testbed program dedicated to control system cybersecurity assessment, standards improvement, outreach, and training. The INL SCADA Testbed includes a full scale electric power grid. The INL power grid includes a 61 mile 128 kV transmission loop, 13.8 kV distribution lines, 7 substations, with more than 3000 monitoring and control points in the system. Additionally, the INL SCADA Testbed

Table 1 – Glossary of terms.

Acronym	Definition
SCADA	Supervisory control and data acquisition
ICS	Industrial control system
HMI	Human machine interface
MTU	Master terminal unit
RTU	Remote terminal unit
PLC	Programmable logic controller
DNP3	Distributed network protocol
MODBUS	Modicon bus protocol
RTDS	Real time digital simulator
PMU	Phasor measurement unit
PDC	Phasor data concentrator
IDS	Intrusion detection system

includes a wireless testbed facility for simulation and test of TCP/IP, ATM, 802.11, GSM, and modem communication signals. Supported wired communication standards include ICCP, MODBUS, DNP3, and other proprietary and public domain protocols. Finally, a Cyber Testbed is available to support testing of firewalls and virtual private networks (VPN). Noted research outcomes from the INL SCADA Testbed Program include published taxonomies of common industrial control vulnerabilities [1], published lessons learned from security assessments control systems [2], participation in standards enhancement and development, and development of recommended procurement language for wireless systems in the advanced metering infrastructure [3]. INL activities primarily involve security assessments, outreach, training, and standards development for the electric power industry. INL partners with industry software and equipment vendors for cyber security assessments of products. INL SCADA Testbed facilities are not made available for university led research.

A European SCADA Testbed has been proposed to provide real world testing facilities for SCADA manufacturers, academic researchers, and other stake holders [4]. A European Network of SCADA Security Test Centres for Critical Energy Infrastructures (ESTEC) [5] has been established to define a design for a European SCADA testbed. ESTEC proposes to provide facilities and resources to identify SCADA vulnerabilities, attack simulation, experiment execution, and standards development and testing.

The British Columbia Institute of Technology (BCIT) houses a SCADA testbed known as the Industrial Instrumentation Process Laboratory. The BCIT lab includes a fully operational distillation column, evaporator, a batch pulp digester, a chemical blending reaction process, and power boiler. The BCIT lab includes a variety of SCADA equipment including Emerson DeltaV and Provox distributed control systems, F&P MC5000 controllers, Foxboro IA digital control systems, Rockwell PLC-5s, Groupe Schnieder 984 and Quantum programmable logic controllers, Honeywell TDC 3000 distributed control systems, BaileyNet90 distributed control system, and GE/Fanuc Series 90/70, Series 90/30 programmable logic controllers with Genius I/O [6].

Various strategies have been used by academic researchers to support testing of proposed industrial control system cybersecurity solutions including small scale SCADA installations and simulated SCADA environments. Multiple research groups have proposed simulation based systems to model control systems, their communication networks, and cybersecurity attacks against a control system. In [7] Montague discusses the emergence of simulation in the industrial control system domain for process design and modeling, operator training, and incident response modeling. Most SCADA security testbeds include a process simulator, network simulators, and a mechanism to initiate a cyber attack on the system. In [8] Davis et al. discuss a simulation based approach to model electric power system by integration of the PowerWorld simulator, simulated network clients, simulated control and power system measurement information, and the Real time Immersive Network Simulation Environment for Network Security Exercises (RINSE) to simulate network traffic and cyber security attacks. In [9] Giani et al. describe a simulated SCADA security testbed. This testbed uses MathWorks Simulink to model a control system and implementation platforms. The authors plan to extend this simulation model to include simulated attacks. Giani et al. also plan to implement a SCADA testbed with commercial hardware and software. In [10] Queiroz et al. propose an architecture for a simulated SCADA security testbed. This system will use LEGO Mindstorm NXT devices to simulate hardware devices such as sensors and actuators. OMNET++ is used to model discrete events, the *libModbus* C library to model MODBUS network traffic, and the INET framework for TCP/IP support. In [11] Chabukswar et al. discuss simulating SCADA cybersecurity attacks. Chabukswar uses the C2WindTunnel framework to integrate results from other simulators. OMNET++, Simulink, and NetworkSim. are subcomponents in the overall simulation environment. In [12] Bergman discusses a simulation framework for SCADA security testbed which models the power grid. This test bed uses the PowerWorld power world to model the electric generation, transmission, and load. Power world is connected to the RINSE network simulator a DNP3 over TCP protocol simulator. Finally, virtual relays are models to act as control system intelligent electronic devices (IED) in the test bed. Simulated testbeds offer a low cost means to model industrial control systems the effects of cyber security attacks on such systems. Simulated testbeds lack the ability to completely model the interactions of control system components.

Two significant testbeds were found which use commercial control system hardware and software. In [13] Fovino et al. provide details of the laboratory used for SCADA cyber security research which includes components from a functional Turbo-Gas Power Plant (modeling a type of electric power generation system). This testbed includes physical apparatus such as pipes, valves, sensors and pumps; common control system equipment from multiple vendors; a process network to interconnect devices related to power generation, a simulated office intranet which is connected to the process network via RADIUS server, a demilitarized zone which houses high speed database servers, a separate observer network to collect information from the control system related to attacks without affecting control system state, and a connection to the external internet. This test bed is very complete model of a electric power generation facility. In [14] Hahn et al. document a test bed which models two electric substations connected to a control center. A power source with auto-transformers and a circuit load is used to model high voltage lines. The substations include software base remote terminal units connected to two over current protection relays. The control center includes human machine interface software and a historian server. Common industrial control system communication protocols used in this test bed include DNP3 and IEC 61850. The substations and control center are connected via VPN server device which also includes basic firewall capabilities. Both of the above testbeds model electric power system components; the first models electric power generation and the second models electric power transmission. The MSU testbed described in this paper includes hardware and software to model a electric power transmission system. A real time digital simulator (RTDS) is available to provide large scale real time power system simulations in a hardware-in-the-loop configuration. Additionally, the testbed includes phasor measurement units, phasor data concentrators, and a commercial energy management system (EMS) to enable Wide Area Monitoring System (WAMS) and Special Protection Schemes (SPS) research.

Researchers in academia, industry and government have primarily concentrated control system cybersecurity research on systems which use routable protocols. The testbed described in this paper includes systems which use routable protocols and systems which use serial based communication. We have shown that serial based systems also are subject to cybersecurity vulnerabilities and attacks [15]. This knowledge has led to research related to retrofit cybersecurity solutions for serial based SCADA systems.

The testbed described in this paper is smaller than the INL SCADA Testbed. The INL cybersecurity assessment mission requires acquisition of many ICS hardware and software products for evaluation. The testbed described in this paper supports an academic and research mission. These goals are accomplished with a testbed built with commercial hardware and software which models control systems from a diverse set of critical industries, with multiple control schemes, and multiple network protocols.

3. Testbed contents and capabilities

The physical processes and control systems in the testbed model systems found in multiple critical industries including electric power transmission, gas distribution, water storage and distribution, manufacturing, mining, and steel manufacturing.

The control systems in the testbed can be divided into 2 categories based upon communication type. First, 5 control systems use serial port communication. Second, 2 control systems use Ethernet communications.

3.1. *Serial port control systems*

The testbed serial port control systems include a water storage tank, a raised water tower, a factory conveyor belt, a gas pipeline, and an industrial blower. These 5 control systems represent a diverse set of industries and control schemes.

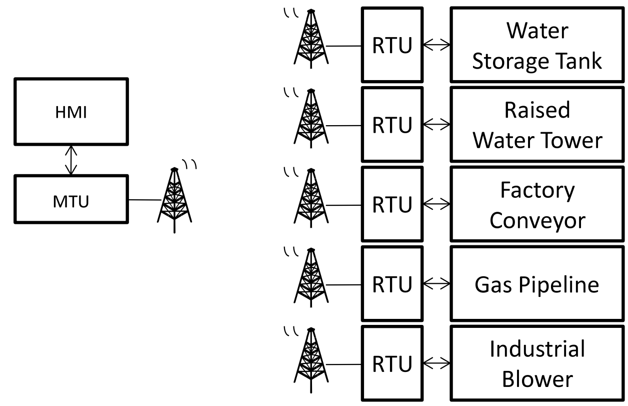


Fig. 1 – Serial port control systems.

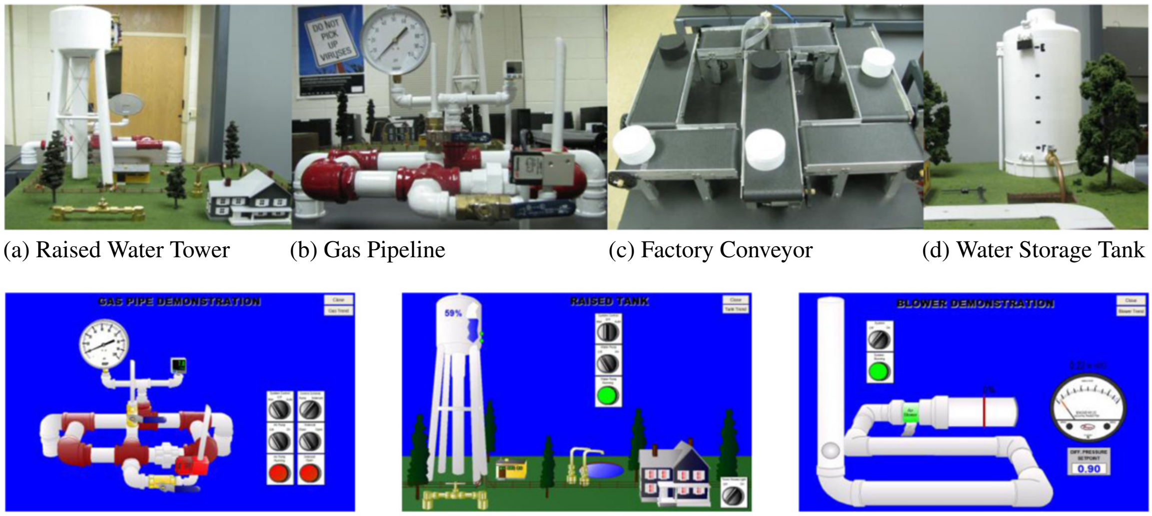
Fig. 1 shows a high level schematic of the 5 serial port control systems. A single HMI is used to control all 5 control systems. The control systems may be operated individually or simultaneously in any grouping.

Fig. 2 shows pictures of the serial port control systems and associated HMI. The actual industrial blower and the HMI screens for the water storage tank and factory conveyor are not shown to conserve space. The actual industrial blower is similar to the drawing on its HMI screen and the HMIs for water storage tank and factory conveyor are similar to the HMI screens shown for the other systems.

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 3 communication protocol MODBUS ASCII, MODBUS RTU, and DNP3. All three communication protocol 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 2 EIA-232 UART. The first UART is connected by serial port cable to the HMI host. The second serial port UART is connected to a industrial 900 MHz radio. Commands from the HMI are received on the HMI port and forwarded to the radio port. The industrial 900 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 may be 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.



(a) Raised Water Tower (b) Gas Pipeline (c) Factory Conveyor (d) Water Storage Tank

(e) Gas Pipeline HMI (f) Raised Water Tower HMI (g) Industrial Blower HMI

Fig. 2 – Pictures of testbed serial port control systems and HMI.

RTU ladder logic for the 5 serial port control systems use a common configuration. RTU include 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, 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.

Each serial port control system is configurable. The testbed includes Telepace ScadaPack Programming Software to modify ladder logic and Telepace C Compiler to compile C programs for use on the Control Microsystems, Inc. A wire bridge is as available on each system to add digital and analog inputs and outputs to the system. New monitoring and control logic associated with new inputs and outputs can be added using ladder logic. The GE/Fanuc iFix HMI screens can be configured to add or remove visualizations or controls. C programs can be used to implement control algorithms or to make network stack changes. SCADAPack LP PLC. One useful network stack change is to modify MODBUS network protocol data units by extending them with a digital signature or by encrypting them. The Telepace C Compiler supports adding a MODBUS extension handler which calls a custom C function when a MODBUS packet with an unknown function code is received. The MODBUS extension handler can decrypt or validate a digital signature before calling *modbusProcessCommand* function to process the MODBUS command. Responses can be intercepted in a similar manner and signed or encrypted before transmission.

Water storage tank

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. 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 alarm state.

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 2 s. 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 L and 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.

Fig. 2(d) shows the water storage tank. The water storage tank HMI screen shows the water level as a percentage of full capacity in blue on the tank and on the moving full/empty gauge on the tank. There are also 2 knobs on the HMI screen which are turned with a mouse press. The upper knob changes system state. Allowed settings are *off*, *manual*, and *automatic*. The lower knob changes the pump state. Allowed settings are *on* and *off*. The HMI screen also includes a system on/off indicator light. Also, the tank shown on the HMI screen includes a moving water level gauge which rises and falls with water level and a percent full indicator.

Raised water tower

The raised water tower control system models water towers used to provide water pressure in water distribution systems in thousands of communities throughout the United States of America. Cyber penetration of control systems monitoring and controlling a raised water tower can lead to a denial of water service with potential economic and public health ramifications.

The raised water tower control system contains a raised water storage tank, a secondary water tank, a pump to move water from the secondary tank to the raised water tank, a gravity fed pipe which allows water to flow from the raised tank to secondary tank, and 4 sensors which provide discrete water levels in the raised tank. The secondary tank is not a feature of a typical water tower. The secondary tank is used to provide a destination for water when it leaves the raised tank and a source for water to fill the raised tank. The raise water tower control system is a closed loop control system.

Ladder logic is used to program the raised water tower MTU and RTU. The MTU ladder logic forwards command and responses to the RTU and HMI respectively. The RTU ladder logic includes 3 setpoint registers; system mode, pump on/off, and town house lights on/off. The RTU ladder logic also includes 8 output registers which store process parameters; pump state, water level, HH, HI, LO, and LL water levels, alarm state, and town house light state. The HH, HI, LO, and LL water level values are set by 4 water level sensors attached to the side of the water tank. Each signal provides a binary indication of the presence of water. If water is touching the sensor the output value is on and if water is not touching the sensor output value is off. The 4 water level sensors are arranged vertically, with LL being the lowest water level and HH the highest water level.

An operator uses the HMI to monitor and remotely control the raised water tower tank. The operator can place the system in automatic or manual mode. In both modes, the HMI polls the RTU ever 2 s. 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 2 setpoints; 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 output values for the 4 water level sensors; HH, HI, LO, LL. If the communication link between the MTU and RTU is broken the RTU will continue operating with the last configured setpoints.

The raised water tower includes automatic and manual control modes. In automatic mode, the RTU ladder logic program attempts to maintain water level between the fixed L and H water levels and alarms if water level exceeds the HH level or is below the LL level. In manual mode, the pump state is controlled manually by the HMI. If due operator error, the water level rises to the HH setpoint or falls to the LL setpoint the alarm is triggered.

Fig. 2 shows the raised water tank and its HMI screen. The HMI screen is animated to show water level, pump state, and the light state. Knobs on the HMI control various setpoints.

Factory conveyor belt

The factory conveyor belt control system models conveyor belt and sorting control systems used to in the manufacturing industry. Cyber penetration of control systems monitoring and controlling a factory based control systems can lead to loss of productivity by shutting down factory operations or by causing errors in the manufacturing process. Both cases may lead to financial loss by the affected factory and to financial loss at other businesses which supply parts to the affected factory or which are supplied by the affected factory.

The factory conveyor belt control system contains a 6 conveyor belts each with 2 motors capable of driving the conveyor belt in forward or reverse direction, a light sensor to detect white objects, a light sensor to detect black objects, and a diverter used to sort white and black pucks by color.

Ladder logic is used to program the factory conveyor belt MTU and RTU. The MTU ladder logic forwards command and responses to the RTU and HMI respectively. The RTU ladder logic includes 4 setpoint registers; a system mode setpoint register, conveyor on/off setpoint register, a motor direction setpoint register, and a diverter control setpoint register. The system mode setpoint register allows an operator to place the system into manual, automatic, or off states. The motor direction setpoint register allows an operator to cause the conveyor motors operate in forward or reverse direction. The conveyor on/off setpoint register allows an operator to start and stop conveyor movement. The diverter setpoint register allows an operator to move the diverter between two positions; left and right. The conveyor on/off, motor direction, and diverter setpoint registers only affect system operation when the system mode is set to manual. The RTU ladder logic also includes 4 output registers which store process parameters; diverter state, motor state, white sensor state, and black sensor state. In automatic mode the diverter state register is updated on the detection of a white or black puck in front of the white or black light sensor. If the black sensor detects a black puck, the diverter is moved to the left to cause the moving puck to be pushed to the right. If the white sensor detects a white puck, the diverter is moved to the right to cause the moving puck to be pushed to the left. In manual mode, the diverter state matches the diverter control setpoint register. In automatic mode, the motor state register is set to forward. In manual mode the motor state register matches the motor direction setpoint register.

The factory conveyor belt sorts black and white pucks which ride on the conveyor belt. White pucks are detected by the white light sensor and cause the diverter to move to its right position, which in turn causes the puck to move to left. Black pucks are detected by the black light sensor and cause the diverter to move to its left position, which in turn causes the puck to move to right. Pucks move along the outer diameter of the conveyor system in a loop and are returned to beginning of the conveyor belt to be resorted.

An operator uses the HMI to monitor and remotely control the factory conveyor belt. The pucks are shown traversing the conveyor system on the HMI screen. The operator can place the system in automatic or manual mode. In both modes, the HMI polls the RTU ever 1 s. Each poll is performed by a MODBUS command being sent from the HMI to the RTU. The MODBUS command requests to read the diverter state, the motor state, the white sensor state, and the black sensor state. The same MODBUS command includes values for the 4 setpoints; system mode, motor direction, conveyor on/off, and diverter control. 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 with the values contained in the output registers. If the communication link between the MTU and RTU is broken the RTU will continue operating with the last configured setpoints.

Fig. 2 shows the physical conveyor belt with pucks and the diverter at the top of the screen, the conveyor belt HMI screen is not shown. The HMI screen is animated to show puck location. Knobs on the HMI control various setpoints.

Gas pipeline

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 visibility and loss of control of the gas pipeline. Both cases may lead to financial loss by affecting billing systems and may lead to physical harm to the gas pipeline and to individuals in the vicinity of the gas pipeline at the time of an incident. In 1991 a Bellingham, Washington process control system became unresponsive and failed to open a pressure relief valve [16]. Due to this failure, the pipeline ruptured and leaked over 250,000 gallons of gasoline into two nearby streams. The leaked gasoline later ignited and created an explosion which killed three persons and injured eight others. The exact cause of this incident is unknown because of a lack of available forensic evidence detailing system activity around the time of the accident. However, in an analysis of the incident, Abrams and Weiss show that the human machine interface software used by operators to monitor and control the pipeline became unresponsive around the time of the pipeline rupture and subsequent explosion. According to Abrams and Weiss the system normally updated screen information, including pipeline pressure, every five to seven seconds, however the system did not update for over 20 min around the time of the incident. This left operators unaware of the exact condition of the process control system and unable to intercede to prevent the ultimate failure.

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 an operator 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 3 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.

Fig. 2 shows the actual gas pipeline system. The pipeline contains analog and digital pressure gauges. The solenoid controlled relief valve is shown in the foreground. The gas pipeline HMI screen is animated to show gas pressure measurements, pump state, and relief valve state. Knobs on the HMI control various setpoints. A second HMI screen allows an operator to adjust the pressure setpoint and the PID parameters.

Industrial blower

The industrial blower control system models an industrial blower used to force air through an exhaust system. Similar industrial blowers are used to evacuate gasses from mines, in heating, ventilating, and air conditioning (HVAC) control systems, and in chemical exhaust hoods. Cyber penetration of control systems monitoring and controlling an industrial blower control systems can lead to loss of visibility and loss of control of the industrial blower. Both cases may lead to safety issues for individuals in areas protected by the industrial blower. For HVAC systems loss of system monitoring and control can lead to an inefficient and or unresponsive air conditioning system.

The industrial blower control system forces air through an air duct. A portion of the air duct is vertical and clear. A small ball inside the vertical portion of duct can be suspended at varying heights by adjusting the PID controller setpoint. A pump is used to push air through the system. A damper valve is used to control the amount of air forced into the primary duct which houses the small ball and the amount of air which is released prior to entry into the primary duct. The system can be turned on or off. Additionally, a PID controller (with requisite PID setpoint, gain, reset, rate, dead band, and cycle time values) is used to control air flow and the height of the small ball in the vertical duct.

Ladder logic is used to program the industrial blower MTU and RTU. The MTU ladder logic forwards commands and responses to the RTU and HMI respectively. The RTU ladder logic includes 7 setpoint registers; a system mode, a PID setpoint, a PID gain, a PID reset, a PID rate, a PID dead band, and a PID cycle time. The system mode setpoint register allows an operator to turn the system on or off. The PID setpoint provides a target ball height in the vertical duct. 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 3 output registers which store process parameters; air pressure measured in inches of H_2O, system state, pump state, and damper position.

When the system is on, the RTU ladder logic program attempts to maintain air pressure in the duct by adjusting the damper position. The PID controller calculates a new damper position value based on the measure pressure and the PID gain, reset, rate, dead band, and cycle time setpoints. Adjusting the air flow has a secondary effect of raising and lowering a floating ball in the vertical portion of the air duct.

Fig. 2 shows the industrial blower HMI screen. The HMI screen is animated to damper position, air flow, the PID setpoint, and pump state. A virtual knob on the HMI turns the system on and off.

3.2. Ethernet based control systems

Many modern control systems use Ethernet based communications infrastructures. These Ethernet networks may be electronically isolated networks dedicated to control system interactions. Such systems are said to have an air gap. Air gapped systems are believed to be more cybersecure due to their isolation. However, air gap systems can still be vulnerable to penetration through unintended network connections or through other attack vectors like infected USB drives [17]. Many control system networks are not electronically isolated. Such networks may have connections to corporate intranets and also to the internet. The trend of using Ethernet based communications for control systems has led to increased cybersecurity awareness and research and development in the control system domain.

The testbed includes an Ethernet network which connects two control systems: a steel rolling operation and a smart grid transmission control system.

Steel rolling operation

The steel rolling operation control system models a four high stand steel rolling operation. These steel rolling operations are used to press sheet metal to add strength via strain hardening and to improving surface finish. Cyber penetration of control systems monitoring and controlling a steel rolling operation can lead to loss of visibility and loss of control of the roller. Additionally, penetration of such a control system can lead to economic loss if an attacker is able to adjust process parameters which affect steel finish quality such as strength or surface finish by changing roll speed at either the entry or exit roll.

The model steel rolling operation is partially simulated. Commercial HMI and RTU are connected to industrial AC single phase motors which act as the motors controlling the rolling operation. The motors turn a dummy load (large weights) rather than actual steel. In its current configuration, the exit roll has the same length as the entry roll, representing a surface finishing case rather than a strengthening operation. Converting to a simulated strengthening operation would only require control scheme changes with no additional hardware.

The steel rolling operation is controlled by an MTU, RTU, and local and remote HMI sharing communication links. Both the MTU and RTU for the steel rolling operation are Allen Bradley Compact Logix L35E PLCs. The master terminal unit is connected to the Factory Talk View 5.0 HMI. Researchers recently added a wireless connection between this master terminal unit and a second lab space housing a Smart Grid transmission control system. Researchers developed new remote HMI screens in addition to the HMI and ladder logic to remotely monitor and control systems in the Smart Grid transmission control system portion of the testbed.

The steel rolling operation contains two local HMI that provide an interface for an operator to monitor and control the operation. The first local HMI is a panel HMI available on the front panel of the case enclosing the system RTU. Operators may monitor and control the steel rolling process directly from the front panel HMI. The second local HMI runs on a PC using Factory Talk View 5.0. The first HMI is hard-wired to the RTU and variable frequency drive motor controllers. The second HMI is connected to the RTU via Ethernet physical link; it communicates with the RTU using

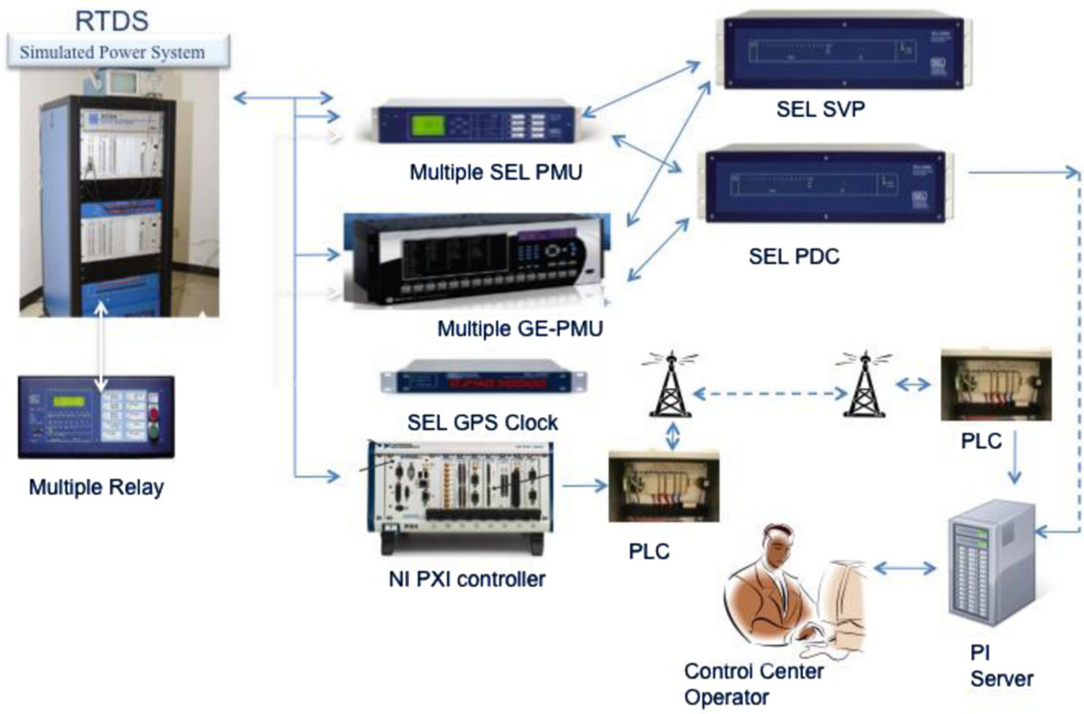


Fig. 3 – Electric substation control system.

Rockwell's EtherNet/IP[1] protocol. Full visibility and control of the steel rolling operation is available with both HMI. The PC HMI forwards EtherNet/IP commands from the operator to the RTU. The RTU executes the commands and then responds. The communication link consists of an Ethernet (IEEE 802.3) physical layer, IP network layer, TCP or UDP transport layer and the Common Industrial Protocol (CIP) application layer protocols.

The steel rolling operation includes manual and automatic control schemes. In manual mode, an operator can set the roll direction setpoint to forward, reverse, or off. Additionally, a variable frequency drive (VFD) speed setpoint is available to set roll motor speed. Manual mode is primarily available for establishing roll tension when loading a new roll and for process debugging. In automatic mode, the system begins to roll the steel by controlling the entry and exit roll motor direction and speed. Steel is rolled from the entry roll to the exit roll through the four high stand. The length remaining and length rolled output register provide information and process progress. The process has completed when remaining length is 0 ft.

The steel rolling operation control system is configurable. The testbed includes Allen Bradley RSLogix 5000 Programming Software to compile and write ladder logic programs for use on the L35E PLCs. A wire bridge is as available on each system to add digital and analog inputs and outputs to the system. New monitoring and control logic associated with new inputs and outputs can be added using ladder logic. The Factory Talk View 5.0 HMI screens can be configured to add or remove visualizations or controls.

Smart grid transmission control system

The Smart Grid transmission control system [18] provides a second environment for research in industrial control system cybersecurity. Commercial devices in the Smart Grid transmission portion of the testbed include phasor measurement units, a phasor data concentrator, a synchrophasor vector processor, protection relays, controllers, a substation GPS clock, an Omicron relay test and calibration device, a Real Time Digital Simulator (RTDS), and OSISoft PI Historian. Additionally, the lab includes amplifiers to provide inputs needed for some of the PMUs (Fig. 3).

The RTDS, protection relays, and phasor measurement units are connected to RTDS in a hardware-in-the-loop configuration. The RTDS is used to simulate transmission and distribution high voltage conditions and scenarios. The protection relays can be configured to monitor bus voltage, current, frequency, and phase conditions to detect faults. Phasor measurement units continuously measure bus voltage and phase, and they transmit measurements to the phasor data concentrator, synchrophasor vector processor, and to the OSI PI historian. Protection relays, phasor measurement units, phasor data concentrators, and the synchrophasor vector processor can be configured over the network using software provided by the various product vendors.

[1] EtherNet/IP is an abbreviation of Ethernet *Industrial Protocol* and is an implementation of the Common Industrial Protocol application layer protocol using Ethernet and the Internet Protocol to carry industrial control systems communications in the. It should not be confused with the network layer Internet Protocol (IP) over Ethernet.

This portion of the testbed also includes a MU Dynamics MU-4000 Analyzer. The MU-4000 is an appliance designed to test network appliances for cybersecurity vulnerabilities. The MU-4000 includes test suites for protocol mutation (also known as fuzzing), denial of service testing, and a test suite of published vulnerabilities. Additionally, a Zenwall 10 Access Control Module and a Cisco 5510 Adaptive Security Appliance are in this portion of the test bed as well. The MU-4000 is used to search for network vulnerabilities, and firewall rules to mitigate vulnerabilities can be developed and validated using the Zenwall 10 and Cisco 5510 firewalls.

Devices in the Smart Grid transmission control system are connected via a common Ethernet network. Industrial control system communication standards in use in the Smart Grid transmission control system include IEEE C37.118, MODBUS/TCP, DNP3, Generic Object Oriented Substation Events (GOOSE), and EtherNet/IP.

4. Pedagogy

The testbeds described in this paper have had a significant pedagogical impact at the university. Integration into the classroom allows the testbed to provide a workforce development function, prepares graduate students for research activities, and raises the profile of this research area with students. A graduate course covering industrial control system security has been developed and taught. Additionally, material related to the testbed and research performed with the testbed has been integrated into projects and lecture materials of multiple of classes, raising the profile of this research problem. Finally, researchers are currently developing a set of short courses to train existing electric utility employees. These courses will train new employees on technologies related to the Smart Grid, including cybersecurity issues.

4.1. *Industrial control system security class*

An industrial control system security class has been developed and offered to graduate students. Material from the course is developed from knowledge gained by research activities in the testbed. Industrial control systems are used to manage physical process by engineers from multiple disciplines. Additionally, professionals from other disciplines including computer science, software engineering, and management information systems may be involved in the commissioning and continuing support of industrial control systems. Finally, cross-functional teams of engineers, computer scientists, security engineers, and management information systems graduates may be called upon to develop, review, and maintain security policies regarding industrial control systems. As such, the industrial control system security course is offered to students in engineering, computer science, and management information systems departments.

The course begins with an introduction to control systems. The phrase “industrial control system” has different meanings to students from different backgrounds. Students are provided with a written description of the control systems available in the testbed. The written descriptions from this paper were originally developed for the class. Detailed descriptions are provided to assist students in comprehending controlled physical processes, the role of individual items in the control system, and the flow of data and control information in the system. Early in the class, students are given a tour of labs which house the testbed. Students are given demos of the control systems showing normal operation and demonstrating control system cyber security attacks. Students are also encouraged to use the human machine interface to control processes to allow for hands-on learning.

Because students come from multiple disciplines, the course includes a set of lectures on basic cybersecurity concepts. Lectures introduce the concepts of confidentiality, integrity, and assurance. Lectures also introduce basic security mechanisms including encryption, authentication, authorization, firewalls, virus protection, certificates, hashes, and digital signatures. After the introduction, all of the cybersecurity concepts are related to securing industrial control systems.

The course introduces multiple network protocols including DNP3, MODBUS, IEC 61850, IEEE C37.118, and Ether/IP. Security features and potential vulnerabilities are discussed for each protocol.

The course includes content on cyber security requirements and recommendations from multiple industry sources. North American Electric Reliability Corporation (NERC) Critical Infrastructure Protection Standards 002-009 [19] are discussed in detail. Risk assessments are also discussed in detail. This portion of the course includes introduction to the Department of Homeland Security National Cyber Security Division Cyber Security Evaluation Tool (CSET) [20] and other risk assessment methodologies. The electronic security perimeter (ESP) concept is discussed in detail, including details on how ESP are achieved in industry. Students are also introduced to NISTIR 7628 Guidelines for Smart Grid Cyber Security [21]. Students walk through the NISTIR 7628 guidelines to develop a set of security requirements for different sub-systems of the smart grid.

Code developed for the testbed to capture and transmit MODBUS and DNP3 link layer protocol data units will be used for a lab activity in the next offering of the class. Students will develop a program to inject false measurements into control systems. Working projects will be demonstrated on the testbed.

Students are required to read and summarize two papers from current literature on industrial control system security. Finally, students prepare a term paper written as a proposal for future research related to industrial control system security. In the first semester the course was offered, research proposals varied from purely software based information technology projects, to network security research proposals, to hardware related items. The research proposal is intended to encourage a subset of students to perform work related to a thesis in this domain.

4.2. *Integration into other courses*

Concepts related to the testbed have been introduced into the computer science, software engineering, computer engineering, and electrical engineering curriculum by the addition of small modules of lecture material and/or laboratory exercises to multiple courses.

Cryptography and network security — students are presented with information about weak cryptographic algorithms found in some industrial control system applications [22].

Data communication and computer networks — Students develop a program to capture and print MODBUS or DNP3 data link layer protocol data units.

Information and computer security — A case study on HMI password security is presented when reviewing the Saltzer and Schroeder concept of least privilege [22].

Special topics: smart grid — This course will cover cross-disciplinary subjects on smart grid. Topics include smart grid standards, optimization and control of the smart grid, communication, advanced metering infrastructure, cyber and physical security systems; microgrids and distributed energy resources, demand response, and system visualization.

4.3. Workforce development

Researchers are currently developing a series of short courses related to the testbed which will be provided as continuing education opportunities for electric power employees. Courses include the following:

Understanding synchrophasors — Introduction to the concept of synchrophasors and how synchrophasors can be used by power system dispatchers and operators.

Cybersecurity concepts for smart grid — North American Electric Reliability Corporation (NERC) Critical Infrastructure Protection Standards 002–009 [19] are discussed in detail. Discussion NISTIR 7628 Guidelines for Smart Grid Cyber Security is also provided [21].

RTDS training — Train power system employees on use of the Real Time Digital Simulator (RTDS) [23], a power system simulator.

5. Cyber security research thrusts

The assets in the testbed provide researchers with unique capabilities to support research across multiple industrial control system (ICS) cyber security domains using the following research process. First, researchers use the testbed to identify hardware, software, and network cybersecurity vulnerabilities. Exploits are developed to study the impact of vulnerabilities on control systems and the controlled physical processes. Then researchers create vulnerability taxonomies which classify vulnerabilities into similar groups based upon control scheme, type of physical process, type of communication, and impact to the controlled physical process. Such taxonomies allow researchers to identify problems which require research to develop cybersecurity mitigation strategies. Finally, researchers implement cybersecurity mitigations and validate their effectiveness using the testbed and the aforementioned exploits. Research outcomes include vulnerability advisories, cybersecurity risk analysis techniques, and cybersecurity tools and architectures.

The following sections provide abbreviated overviews of research enabled by the testbed.

5.1. Vulnerability discovery — software vulnerabilities

The recent discovery of the W32.Stuxnet [24] worm that has infected over 100,000 computers in over 155 countries [17] highlights the need to ensure human machine interface software and the hardware devices which host HMI software are free of vulnerabilities and protected against zero-day attacks.

Researchers are developing a taxonomy of software vulnerabilities related to industrial control systems. This taxonomy is being used to derive new research directions for securing such systems. Developing the software taxonomy requires access to commercial HMI software which is available in the testbed.

HMI software often use usernames and passwords to provide a level of role based access control with different privilege levels assigned to various personnel. Research using the testbed has led to multiple vulnerability discoveries.

Results from the study of one HMI revealed a password recovery vulnerability, insecure transmission of remote authentication credentials over a network, and the ability to bypass authentication routines to elevate privilege [22]. HMI passwords were on the HMI host in an insecure manner. Passwords were stored after XOR with a fixed key. The XOR process obfuscated passwords, however, the fixed key is recoverable with a chosen plaintext attack. Further examination of the same HMI found that the mechanism to support remote authentication passed the password file over the network encrypted in the manner described above to the remote client attempting to login. This practice provides an easy means for attackers to access the password file and enables the aforementioned password recovery vulnerability. Finally, review of the studied HMI revealed that the HMI did not include mechanisms to prevent replacement of executable code related to the HMI from outside of the HMI running context. Researchers leveraged this vulnerability to replace a DLL which included authentication routines with an altered copy which bypassed all authentication checks and gave all users elevated privileges. These vulnerabilities were reported to the software developer and to US-CERT as VU#310355 US-CERT.

After the above discovery HMI products from multiple vendors were purchased to continue to study this investigation. Multiple subsequent vulnerabilities have been found and reported to software developers and appropriate critical infrastructure protection authorities. These results are not documented in this paper because the ethical disclosure process is ongoing.

5.2. Vulnerability discovery — network vulnerabilities

The testbed includes both Ethernet and serial based networks with wired and wireless links. Multiple application layers are represented including MODBUS, DNP3, Ether/IP, GOOSE, and IEEE C37.118. Researchers have used the tested bed to discover proprietary radio vulnerability and to explore the effects of network attacks which compromise integrity and availability. Discovery of the proprietary radio vulnerability would not have been possible without the testbed because no specifications or models are available for review. Furthermore, simulation models can be developed which combine models of physical processes, control system devices, and control system networks. However, in many cases it is easier and more informative to implement a laboratory scale control system and physical process such as is used in the testbed.

Proprietary radio vulnerability

The testbed was used to discover a vulnerability which allows an attacker discover, infiltrate, and subsequently attack control systems which use a popular brand of industrial radio [15]. The radios tested for this work are used for RS-232 serial communications between two devices (presumably an MTU and RTU, but in general any RS-232 device). A pair of radios are used to replace a null modem cable; each radio has one serial port which is connected to an end device (MTU or RTU), and the radios autonomously negotiate communications between themselves.

The radios provide a number of parameters for configuring transmission and network characteristics. Parameters include node type (master, slave, repeater), a network identifier all nodes of a network share, a frequency key for choosing the frequency hopping pattern, and several parameters related to data rate and packet size. Devices with matching parameters autonegotiate a connection. The radios use frequency hopping spread spectrum modulation. Frequency hops occur 200 times per second.

Discovery of a radio network is the first step required to enable an attack. To discover a wireless radio of which uses the tested radios an attacker must first possess a radio of the same type. Such radios are available for sale on the internet and easily interface with a laptop computer. The discovery attack consists of attempting to connect to all allowable network identifier and data rate combinations (12,288 combinations). There are a limited number of available frequencies used in the frequency hopping pattern. Two radios with different frequency hopping keys will intermittently have overlapping actual transmission frequencies for short periods of time. During these overlaps, if the network identifier and data rate parameters match, some data from valid network member nodes will be accepted by the rogue radio and placed in its output buffer. This spurious data is used to identify an existing network using a given network identifier and data rate pair. Searching the entire network identifier and data rate space requires approximately 6.375 s per combination meaning a network can be scanned in 21 h. A second scan is sometimes required to eliminate false positives yielding a total scan time of 42 h. The radios may use a serial number list in place of network identifier. A similar attack to that listed above can be used to find networks configured to use serial number lists.

Once a radio network is discovered it can be infiltrated using an exhaustive search of the remaining unknown radio parameters. Radios configured as a slave and with all configuration parameters matching an existing network will autonegotiate a connection and join the network as a slave. Experimentation using the testbed has shown that a radio network with an unknown set of connection parameters can always be found (search all possible parameter values) in approximately thirty nine days. Thirty nine days requires a significant investment by an attacker; however, since industrial control system nodes are not mobile such an attack is feasible.

The tested radios have a dynamic key substitution encryption feature which is not configurable. The encryption feature is not able to stop a radio with matching connection parameter from connecting to the network.

Once matching connection parameters are known multiple attacks are possible including denial of service attacks, eaves dropping, and data injection.

The radio tested for this work is a proprietary system. Detailed state diagrams, network stack source code, and or simulation models are not available to perform vulnerability studies. As such, the testbed was required to find the vulnerability using reverse engineering and laboratory testing techniques. Furthermore, implementing the discovery and infiltration attacks with the testbed made it possible to demonstrate denial of service and response injection attacks on live control systems.

Network attacks against control system integrity and availability

Researchers have used the testbed to develop three classes of attacks against industrial control system network datagram availability and integrity. These attacks are used to identify industrial control system vulnerabilities, to highlight ICS research needs, and to support development of intrusion detection systems targeted for industrial control systems. First, command injection attacks are attacks against the RTU. Network traffic is injected which includes false control information. Command injection attacks may change individual setpoints on the RTU or overwrite firmware on the RTU. Second, response injection attacks are against the HMI or MTU. Response injection attacks inject network traffic to send false responses or measurements to a control system HMI or MTU after a data request. Third, denial of service (DOS) attacks disrupt the communication link between the RTU and MTU or HMI.

A set of command injection attacks have been developed for use against the serial based Modbus control systems in the testbed. Each of the serial port control systems includes setpoints which can be altered by a command injection attack. For example, a command injection attack can change the following setpoints in the water storage tank system: system mode, pump override, HH and LL alarm water levels, and H and L water levels. Similar command injection attacks are possible for the other serial based control systems in the testbed. Ethernet control systems are also subject to command injection attacks. Any network appliance connected to an Ethernet network can transmit packets which mimic control system commands. Command injection attacks were developed against the temper mill control system, which uses the Ether/IP protocol, which change the roll length, entry roll mode, entry roll manual, exit roll mode, exit roll manual, entry roll speed, and exit roll speed setpoints.

HMI send commands to read setpoint values from the RTU and to read measurements from analog and digital inputs connected to the RTU. MODBUS, DNP3, EtherNet/IP and many other control system communication protocols do not authenticate responses from RTU. Response injection attacks replace valid responses with falsified responses. Any RTU output value or stored setpoint value can be falsified with a response injection attack. Multiple classes of response injection attacks were developed for each control system in the testbed. For the water storage tank false response attacks were developed to inject negative water values, water values which generate alarms, random water level values, and false water level measurements which are within the ranged defined by the H and L set points. For the water tower system false response attacks were developed for false alarm state, false pump state, false light state, for illegal or impossible combinations of water level sensors, and for valid but false water level sensor values. For the gas pipeline system false response attacks were developed to falsify pump state, falsify relief valve state, for negative pressure measurements, for random pressure measurements, for zero pressure measurements, and physically impossible pressure trends, and for physically possible but false pressure trends.

One denial of service attack was developed against the serial port control system which uses the aforementioned proprietary radio network. Slaves connected to the radio network use a carrier sense back off mechanism before transmitting. A rogue slave may transmit continuously and stop all other slave devices from transmitting which stops RTU measurements from being transmitted. A rogue slave that penetrates the wireless network via methods described earlier in this paper can deny all other slave the right to transmit data.

The testbed includes a Mu Dynamics MU-4000 Analyzer, a network tester used to test network appliances for denial of service and protocol mutation vulnerabilities. The MU-4000 includes a suite of denial of service attacks which have been used to generate denial of service attacks against multiple devices in the testbed. Denial of service attacks can be launched which send increasing volumes of network traffic on multiple network layers. Data link layer attacks include ARP floods. Network layer attacks include IP floods with random higher layer protocols, IP floods with random header values and payload sizes, IP floods with large IP fragments, IP fragment overlapping attacks (aka. tear drop attack), and ICMP flooding (aka. ping flood or Smurf attack) Transport layer attacks include TCP SYN/FIN floods to designated TCP ports, TCP SYN port scan flooding to random TCP ports, TCP LAND attack flood, UDP port flooding, and UDP multicast flooding. TCP and UDP denial of service attacks should be repeated against all ports implemented on a network appliance.

The Mu Dynamics MU-4000 Analyzer can also be used to generate protocol mutation attacks. Devices often exhibit unknown behavior including hanging or resetting when confronted with unexpected network traffic. Protocol mutation exhaustively searches through possible combinations of values which are possible for all packet fields (address, length, payload, CRC, etc.) for a given protocol. Fields which can be mutated include all fields in a packet header, packet payload, and packet trailer. The MU-4000 was used to attack multiple devices in the testbed. Mutated protocols include ICMP, ARP, IP, TCP, DNP3, MODBUS, and IEEE C37.118.

Man in the middle (MITM) attacks against Ethernet based control systems can be used to inject false commands, inject false responses, create replay attacks, and to execute denial of service attacks. The Ettercap tool was used to create an ARP poisoning based MITM attack between a phasor measurement unit and phasor data concentrator in the testbed. Ettercap supports active packet content analysis and filtering. Traffic from the phasor measurement unit was captured by the MITM node and the voltage magnitude for all synchrophasor measurements was doubled before retransmission to the phasor data concentrator. The same MITM setup can be used to inject completely false synchrophasor measurements or to simply accept the stream of synchrophasors and not forward the data causing a denial of service attack. This type of MITM attack can be executed against any Ethernet based control system without properly configured intrusion detection and prevention systems.

Control system intrusion detection and intrusion prevention system researchers often first develop a set of attacks against an available control system testbed, then develop an intrusion detection system, and then validate the intrusion detection system against the set of attacks. This methodology makes it difficult to compare the effectiveness of different intrusion detection and prevention approaches. Command injection, response injection, and denial of service attacks were developed and executed against the control systems in the testbed to create a database of known attacks. This database is intended for use for development and testing of intrusion detection algorithms developed for industrial control systems. The authors intend to share the aforementioned attack database with other vetted researchers.

A serial MODBUS and DNP3 network traffic data logger developed for the testbed (described in the section) was used to create network traffic logs of systems operating normally and systems subjected to the attacks described above. The data logger is a bump-in-the-wire device which introduces extra latency for serial port packets. All packets are slowed equally and the latency has been shown, using the testbed, not affect control system functionality. Therefore, the data logger is an acceptable platform to capture live system data without interfering with the ongoing experiment. Data logs are post-processed to mark packets as malicious or non-malicious. For Ethernet based attacks Wireshark [25] running in promiscuous mode on a separate data logger PC was used to capture network traffic from the attack scenarios and to capture traffic from normal operation. For Ethernet log files, attackers can generally be identified by IP address, MAC address, or evil bit (asserting unused bit in IP header to indicate attack) enabling automated marking of data logged transactions as attack or normal transactions. All of the data logs described above are used to enable intrusion detection research.

The attacks described above were implemented against commercial control system hardware devices controlling functional physical processes. Captured data logs therefore represent reactions from real devices and network stacks rather than models.

5.3. *Serial MODBUS and DNP3 network traffic data logger*

The possibility of command injection, response injection, and denial of service attacks outlined above raises the need for a device to capture and store SCADA control system network traffic for post-incident forensic analysis and to serve as an input to real time and post-incident intrusion detection systems. Such a data logger was developed and validated with the testbed [26].

The data logger uses a bump-in-wire configuration which allows it to serve as a retrofit module that works with legacy SCADA systems and does not negatively impact the normal functionality of the SCADA control system. The data logger is compatible with serial MODBUS (RTU and ASCII modes) and serial DNP3 networks.

Network transactions are stored indefinitely for post-incident retrieval. All network transactions captured by the data logger are time stamped. Time stamping allows engineers to develop an incident time line from logged transactions. Stored network transactions are signed with an HMAC which ensures stored transactions are irrefutably stored by the data logger device and not written to the device at a later time or by means other than via the data logger device. Stored network transactions are encrypted to ensure that lost or stolen data logs may not be deciphered by unauthorized parties.

The data logger can be used in a software configuration or a standalone hardware configuration. The software configuration runs in a virtual machine on the HMI host platform. The HMI is connected to the data logger virtual machine which captures and stores network transactions then forwards transactions to the host platforms physical network interface. The software configuration is effectively a software bump-in-the-wire. The standalone hardware version of the data logger is intended as a physical bump-in-the-wire device. The standalone version was prototyped on a Xilinx FPGA development board in which case the data logger process runs on a Microblaze CPU clocked at 120 MHz.

The data logger was tested with the testbed to confirm that inclusion as a software or hardware bump-in-the-wire produced no harmful side effects against the SCADA control system. The serial port HMI in the testbed poll RTU every 1–2 s. SCADA data logger latency must be less than the polling period to allow the control system to operate normally. Adherence to this requirement was proven by retrofitting all serial port control systems in the testbed with the developed data logger device and confirming continued control system functionality.

The standalone data logger stores transactions on a compact flash card for the embedded version and hard disk drive for the virtual machine version. Both versions of the data logger store the captured MODBUS or DNP3 datagram, a time stamp, a nonce, and digital signature derived using HMAC-SHA1 algorithm.

Electric transmission system situational awareness algorithms poll SCADA devices every 2–4 s. For the 2 s polling case, the stand alone data logger can store 2.8 years of MODBUS ASCII transactions per gigabyte, 5.0 years of MODBUS RTU transactions per gigabyte, and 4.5 years of DNP3 transactions per gigabyte. Since, virtual machine version runs on the HMI on the host PC, it has access to that system's hard disk drive providing significantly more storage capacity. Average latency for MODBUS RTU transactions captured and forwarded by the virtual machine data logger and the stand alone hardware data logger had latency (measure as time between first byte transmission and first byte received) averaged less than 50 ms which compares favorably to average latency without presence of data logger which averaged approximately 35 ms.

The serial MODBUS and DNP3 network traffic data logger project demonstrates that it is feasible to build a retrofit data logger for serial control systems which supports logging SCADA network traffic for post-incident forensic analysis and provides an input sensor for intrusion detection systems designed to monitor SCADA control system network traffic for illicit penetrations.

5.4. *Response and measurement injection intrusion detection*

Many attacks on industrial control system availability and integrity can be detected using a signature based intrusion detection system (IDS). For example, a denial of service attack previously described continuously transmits network traffic from a rogue slave which has penetrated the wireless network. A signature based IDS can be programmed alarm if it detects continuous transmissions greater than a set length. MODBUS and DNP3 protocol accept the first response from a slave with matching address as the legitimate response from a command. Researchers have demonstrated with the testbed that rogue slaves can transmit responses faster than legitimate devices and therefore inject false responses in a control system's HMI feedback control loop. Ethernet and serial MODBUS and DNP3 protocols are susceptible to this attack. However, such an attack is detectable by signature IDS because network packet inspection will detect 2 responses for a given command.

Signature based IDS require prior knowledge of threats to develop signatures. New attacks and variants on existing attacks can be missed by signature based IDS. Also, certain attacks are difficult for a signature IDS to detect. For instance, illicitly injected commands and responses may appear valid to a signature based IDS. Replay attacks are also difficult for a signature based IDS to detect.

Statistical IDS can be used to classify network activity into normal and abnormal categories. To test the effectiveness of statistical IDS in SCADA control system networks a set of false response injection attacks were developed against the water storage tank control system [27]. Six classes of response injections were developed against the water storage tank control system (Table 2).

Researchers used the testbed to implement an exploit to inject the false responses onto the water tank control system network and cause the water tank HMI to accept and display false water level values. The serial MODBUS and DNP3 network traffic data logger was used to capture network traffic while systems were under attack and while operating normally. Individual MODBUS responses were marked as either a normal response or attack response.

A back propagation algorithm was used to build a three stage neural network which classifies logged SCADA network transactions as normal or attack. The neural network used four input features. The first input feature was the measured water level in the tank. The second input feature was command response frequency. The third input feature is the mode of operation of the control system. The mode of operation can be set as either automatic or manual. In

Table 2 – Water tank control system false response injection classes.

Response injection class	Description
Negative	Water level is negative. This and impossible water level value, though it can be injected.
HH alarm	Water level is above the HH alarm setpoint. This may be a single false measurement or a group of false measurement.
Above H setpoint	Water level is above the H setpoint though below the HH alarm setpoint. This may be a single false measurement or a group of false measurement.
Below L setpoint	Water level is below the L setpoint though above the LL alarm setpoint. This may be a single false measurement or a group of false measurement.
LL alarm	Water level is below the LL alarm setpoint. This may be a single false measurement or a group of false measurement.
Random	Water level is a random value or group of random values. This may be a single false measurement or a group of false measurement.
Replay	Water level is a value or set of values capture from previous activity and retransmitted and current measurements.

Table 3 – Neural network classification results.

Exploit scenario	False positive (%)	False negative (%)	Accuracy (%)
Negative	0.0	0.0	100.0
HH alarm	4.5	0.0	95.5
Above H setpoint	2.3	3.0	94.7
Below L setpoint	2.4	3.0	94.6
LL alarm	3.2	0.0	96.8
Random	6.2	8.9	84.9
Replay	45.1	42.7	12.1

automatic mode, the RTU automatically maintains the water level between the L and H set points. In manual mode, the RTU waits for input from the MTU to control the water level. The fourth input feature is the state of the water tank pump. The pump can be set to on or off. During testing the water relief valve was held open to allow water to continually flow out of the tank. This caused the water level to cycle up and down as the pump is turned on and off.

Results showed that the neural network could successfully detect system behavior changes related to false response injection attacks (see Table 3). One weakness of the neural network classifier was an inability to detect control system replay attacks derived from previously captured network traffic.

Work continues in the testbed to identify new input features which produce increase intrusion detection accuracy and decreased false positives and false negatives. Researchers are also considering other classifier types, including support vector machines and decision trees.

This work attempts to leverage physical properties of the processes being controlled to detect illegal or impossible variations in process measurements. Having access to control systems with physical processes from multiple critical industries allows researchers to develop ideas for intrusion detection for one control system and physical process and then study how the ideas transfer to protect control systems from other critical industries. For instance, researchers are currently studying the ability to predict future measurements from past results based upon the slope and acceleration of a measurement trend line. The distance from the predicted measurement to the actual measurement is used as a classifier input feature. This idea was developed for the water tank control system and then found to also have merit on the gas pipeline control system.

6. Conclusions

The Mississippi State University (MSU) SCADA Security Lab and the MSU Power and Energy Research Lab form a testbed for industrial control system cyber vulnerability discovery and solutions research. The labs are built with commercial hardware and software devices from multiple vendors. The testbed includes commercial equipment and software monitor and control laboratory scale physical processes from multiple critical infrastructure protection industries including electricity infrastructure, gas pipelines, factory systems, water storage and distribution, and mining. Equipment and software include human machine interfaces, master terminal units, remote terminal units, programmable logic controllers, industrial serial to Ethernet converters, industrial radios, industrial routers, industrial firewalls, energy management system, historians, digital protection relays, phasor measurement units, and phasor data concentrators. To demonstrate the utility of such a test bed details on classroom integration and overviews of research conducted within the testbeds are provided.

REFERENCES

[1] Common Cyber Security Vulnerabilities Observed in Control System Assessments by the INL NSTB Program, Idaho National Laboratory, Idaho Falls, Idaho 83415, November 2008. http://www.inl.gov/scada/publications/d/inl_nstb_common_vulnerabilities.pdf.

[2] R. Fink, D. Spencer, R. Wells, Lessons learned from cyber security assessments of SCADA and energy management systems, Idaho National Laboratory, Idaho Falls, Idaho 83415, September 2006. http://www.inl.gov/scada/publications/d/nstb_lessons_learned_from_cyber_security_assessments.pdf.

[3] Wireless Procurement Language in Support of Advanced Metering Infrastructure Security, Idaho National Laboratory, Idaho Falls, Idaho 83415, August 2009. http://www.inl.gov/scada/publications/d/inl-ext-09-15658_ami_proc_language.pdf.

[4] H. Christiansson, E. Luiijf, Creating a European SCADA security testbed, in: E. Goetz, S. Shenoi (Eds.), IFIP International Federation for Information Processing, in: Critical Infrastructure Protection, vol. 253, Springer, Boston, pp. 237–247.

[5] ESTEC Project. http://www.estec-project.eu/.

[6] Industrial Instrumentation Process Lab. http://www.bcit.ca/appliedresearch/tc/facilities/industrial.shtml.

[7] Jim Montague, Simulation breaks out, in: Control Global, Sep-2010, pp. 52–61.
[8] C. Davis, J. Tate, H. Okhravi, C. Grier, T. Overbye, D. Nicol, SCADA cyber security testbed development, in: Power Symposium, 2006. NAPS 2006. 38th North American, 2006, pp. 483–488.
[9] Annarita Giani, Gabor Karsai, Tanya Roosta, Aakash Shah, Bruno Sinopoli, Jon Wiley, A testbed for secure and robust SCADA systems, in: 14th IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS'08, WIP session, 2008.
[10] C. Queiroz, A. Mahmood, Jiankun Hu, Z. Tari, Xinghuo Yu, Building a SCADA security testbed, in: Network and System Security, 2009. NSS'09. Third International Conference on, 2009, pp. 357–364.
[11] Rohan Chabukswar, Bruno Sinpoli, Gabor Karsai, Annarita Giani, Himanshu Neema, Andrew Davis, Simulation of network attacks on SCADA systems, in: First Workshop on Secure Control Systems, 2010.
[12] David C. Bergman, Power grid simulation, evaluation, and test framework, Master's, University of Illinois, 2010.
[13] I. Fovino, M. Masera, L. Guidi, G. Carpi, An experimental platform for assessing SCADA vulnerabilities and countermeasures in power plants, in: Human System Interactions, HSI, 2010 3rd Conference on, 2010, pp. 679–686.
[14] A. Hahn, et al. Development of the PowerCyber SCADA security testbed, in: Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research — CSIIRW'10, 2010, p. 1.
[15] B. Reaves, T. Morris, Discovery, infiltration, and denial of service in a process control system wireless network, in: IEEE eCrime Researchers Summit, Tacoma, WA, October 20–21, 2009.
[16] M. Abrams, J. Weiss, Bellingham, Washington, Control system cyber security case study, National Institute of Standards, Information Technology Laboratory, August, 2007.
[17] N. Falliere, L. Murchu, E. Chien, W32.Stuxnet Dossier, Version 1.3, Symantec Security Repsonse, November 2010. http://tinyurl.com/36y7jzb.
[18] R. Reddi, A. Srivastava, Real time testbed development for power system operation, control and cyber security, in: North American Power Symposium, NAPS, 2010, 2010, pp. 1–6.
[19] Critical Infrastructure Protection Standards 002-3–009-3, North American Electric Reliability Corporation, December 16, 2009. Available online at http://www.nerc.com/page.php?cid=2---20.
[20] National Cyber Security Division (NCSD), Cyber Security Evaluation Tool (CSET), Department of Homeland Security. Available online at http://www.us-cert.gov/control_systems/satool.html.
[21] The Smart Grid Interoperability Panel Cyber Security Working Group, NISTIR 7628 Guidelines for Smart Grid Cyber Security, National Institute of Standards, September 2010. Available online http://csrc.nist.gov/publications/PubsNISTIRs.html.
[22] R. McGrew, R. Vaughn, Discovering vulnerabilities in control system human-machine interface software, Journal of Systems and Software 82 (4) (2009) 583–589.
[23] RTDS Technologies, Real Time Digital Simulator. Available online at http://www.rtds.com/index/index.html.
[24] L. O'Murchu, Last-minute paper: an indepth look into Stuxnet, in: The 20th Virus Bulletin International Conference, Vancouver, BC, Canada, September 29–October 1, 2010.
[25] Wireshark Network Protocol Analyzer. Available online at http://www.wireshark.org/.
[26] T. Morris, K. Pavurapu, A retrofit network transaction data logger for SCADA control systems, in: 2010 IEEE international conference on power and energy, PECON, Kuala Lampur, Malaysia, November 29, 2010.
[27] W. Gao, T. Morris, B. Reaves, D. Richey, On SCADA control system command and response injection and intrusion detection, in: The Proceedings of 2010 IEEE eCrime Researchers Summit, Dallas, TX, Oct 18–20, 2010.