Chapter 14

MULTI-CONTROLLER EXERCISE ENVIRONMENTS FOR TRAINING INDUSTRIAL CONTROL SYSTEM FIRST RESPONDERS

Joseph Daoud, Mason Rice, Stephen Dunlap and John Pecarina

Abstract When systems are targeted by cyber attacks, cyber first responders must be able to react effectively, especially when dealing with critical infrastructure assets. Training for cyber first responders is lacking and most exercise platforms are expensive, inaccessible and/or ineffective. This chapter describes a mobile training platform that incorporates a variety of programmable logic controllers in a single system that helps impart the unique skills required of industrial control system cyber first responders. The platform is modeled after a jail in the United States and was developed to maximize realism. Training scenarios are presented that cover specific cyber first responder skills and techniques. The results demonstrate that the platform is robust and highly effective for conducting sustained training exercises in curricula developed for cyber first responders.

Keywords: Industrial control systems, cyber first responders, training platform

1. Introduction

Diseases can manifest themselves in a number of ways depending on the individual. For health care professionals, this can sometimes make a diagnosis difficult and an accurate prognosis challenging. To prepare themselves to perform these tasks, medical students go through a rigorous curriculum that goes well beyond traditional classroom lectures. The curriculum involves many practical exercises, clinical rotations, internships and residency [5]. The knowledge, skills and experience gained from such a curriculum enhances a physician's ability to analyze a patient's symptoms in the context of the patient's unique medical history in order to arrive at a diagnosis and an accurate prognosis [2].

M. Rice and S. Shenoi (Eds.): Critical Infrastructure Protection XI, IFIP AICT 512, pp. 273–294, 2017.
https://doi.org/10.1007/978-3-319-70395-4_14

As in the case of human diseases, cyber threats manifest themselves in many different ways and cyber first responders must be able to diagnose and respond to the threats just like physicians. Given the similarities between the two endeavors, it is reasonable to expect that hands-on training similar to what medical students receive would be very effective for cyber first responders.

This chapter describes a training platform that is specifically designed to provide realistic training for cyber first responders. The mobile training platform incorporates several programmable logic controllers (PLCs) as well as other realistic hardware and software components that maximize the knowledge, skills and experience gained by cyber first responders during their training.

2. Background

Academic institutions, government organizations and businesses offer a variety of certifications, training courses and degree programs in the area of cyber security [9, 12]. These programs provide cyber first responders with valuable skills. However, the vast majority of these programs focus on traditional information technology (IT) systems, often neglecting operational technology (OT) systems. While there is some overlap between the two types of systems with regard to security, the differences are significant enough that cyber first responders need specialized knowledge, skills and experience to handle operational technology incidents involving industrial control systems. Two distinguishing characteristics of industrial control systems are the heavy use of proprietary software and communications protocols and the focus on safety. Almost every vendor has its own proprietary applications for interacting with its control devices (e.g., programmable logic controllers). Additionally, industrial control systems manage physical processes in which anomalies can present significant safety risks. Cyber first responders must be cognizant of these factors when conducting their activities.

Several industrial control system testbeds have been developed, but the vast majority of testbeds are geared toward research and development as opposed to education and training:

- **Sandia National Laboratories:** Sandia National Laboratories operates several facilities, including the Distributed Energy Technology Laboratory, Network Laboratory, Cryptographic Research Facility, Red Team Facility and Advanced Information Systems Laboratory [11]. All these testbeds contain real and simulated supervisory control and data acquisition (SCADA) assets for research and development in various domains. For example, the Distributed Energy Technology Laboratory houses industrial control systems used in electricity generation and distribution; however, control system security is not necessarily the primary focus of research at the laboratory [10].

- **Idaho National Laboratory:** Idaho National Laboratory has several facilities [7]. One of its cyber security facilities is intended to connect to several existing critical infrastructure testbeds, including a SCADA

testbed, power grid testbed, mock chemical mixing facility, wireless testbed and physical security testbed. The testbeds comprise a full-scale critical infrastructure test range that covers 890 square miles. Unfortunately, due to the nature of the facility, most learning opportunities are restricted to authorized individuals from government and industry [8].

- **National Institute of Standards and Technology:** The National Institute of Standards and Technology is tasked with publishing guidelines and recommended practices in many disciplines, including cyber security. The NIST industrial control system testbed was developed to enable the evaluation of security guidelines and best practices [4]. The testbed comprises real and emulated industrial control system components that can be evaluated with the appropriate security mechanisms in place.

- **SANS Institute:** The SANS CyberCity is one of a few physical industrial control system platforms that is specifically designed for security training. The CyberCity platform is used in the SANS SEC562 course, which focuses on penetration testing and kinetic cyber effects [13]. It is a 1:87 scale city with hands-on exercises involving railway switching junctions, a water reservoir and power grid. While some of the systems in CyberCity are simulated, real hardware is incorporated in the power grid system, including Allen-Bradley, Siemens and Phoenix Contact programmable logic controllers [14]. CyberCity is an effective training platform, but it is very expensive. Furthermore, it is not a mobile platform. While it can be accessed remotely for training purposes, remote training does not support intense, hands-on interactions with physical components, which is an important aspect of cyber first response training.

Effective training curricula must be in place to enable professionals to assess and react to cyber incidents involving industrial control systems. Butts and Glover [3] propose three core areas that should be covered in an industrial control system training course: (i) industrial control system principles; (ii) cyber manipulation; and (iii) response coordination. Each core area has recommended instructional blocks that cover important areas of proficiency.

The industrial control system principles core provides an introduction to common control system components, cyber-physical interactions involving these components, communications protocols and real-world configurations. The cyber manipulation core covers attack techniques that target industrial control system components and networks. The response coordination core primarily focuses on industrial control system incident response. Butts and Glover [3] recommend that all these concepts be taught via realistic scenarios involving genuine industrial control systems.

Even the best training platforms have very limited value unless they are used appropriately. Developing realistic training scenarios is an important, but difficult, task. Traditional "capture the flag" events, which are focused on gaining access, are fundamentally inadequate for industrial control systems.

The actions taken after gaining access to an industrial control system are far more important.

An effective evaluation of an industrial control system scenario must incorporate the physical process being controlled. Yoon et al. [17] leverage the NFPA 1410 format, which is used by firefighters, to develop an effective framework for evaluating the readiness of cyber first responders. This framework contains specific objectives, descriptions, evaluation criteria and accompanying references for each training scenario. Furthermore, each scenario contains a designator that describes the type of scenario and the skills addressed by the scenario. The framework proposed by Yoon and colleagues is used in this research to develop training scenarios with measurable evaluation criteria.

3. Multi-PLC Training Platform

This section describes the design considerations and implementation details of the multi programmable logic controller training platform.

3.1 Design Considerations

The training platform is designed to incorporate multiple programmable logic controller models, thereby emphasizing the differences between the individual programmable logic controllers.

Requirements. The training platform is intended to be reasonably inexpensive and mobile so that training can be conducted at multiple locations. A replica of a jail was created within a 55.32 cm × 42.39 cm × 26.97 cm Pelican 1610 case using genuine components and realistic programs. To enhance realism, the components were selected based on the design of an actual jail in the United States. Ladder logic programs for the programmable logic controllers were created to implement the same operations as the real jail. Any one of the three programmable logic controllers can be selected by the training administrator to be active at a given time. Figure 1 shows the completed training platform.

The training platform is designed to meet five criteria:

- Incorporate physical components.
- Incorporate cyber manipulation principles.
- Incorporate response coordination techniques.
- Provide hands-on experience.
- Implement effective training scenarios with measurable training evaluation metrics.

Components. Table 1 lists the main components of the platform. The pushbuttons, indicator lights and turnkey replicate components that are found

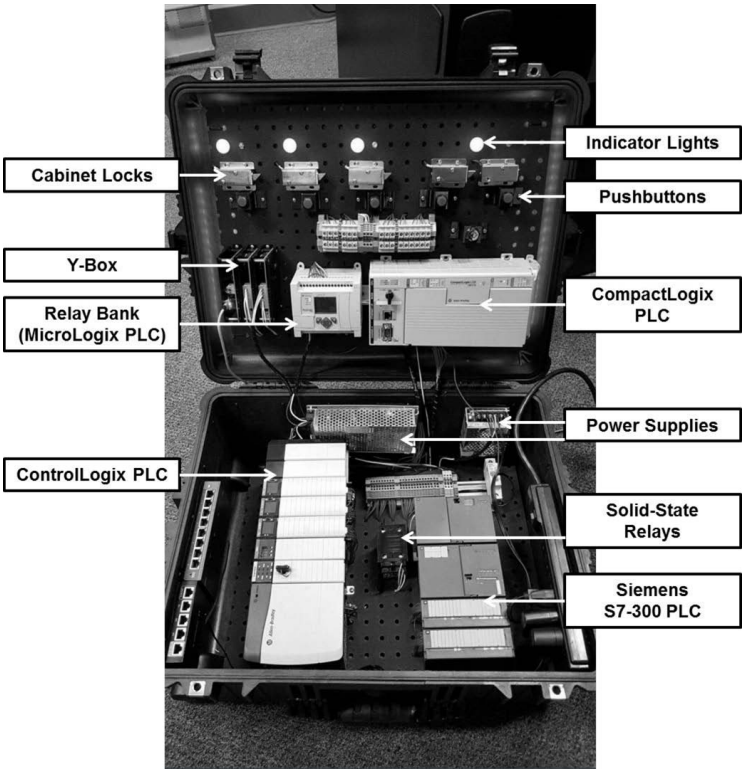

Figure 1. Training platform.

Table 1. Training platform components.

Component	**Quantity**	**Component**	**Quantity**
Cabinet Locks	5	Pushbuttons	5
Relays (Electromechanical)	5	Red Lights	4
Relays (Solid State)	3	Peg Boards	2
Power Supplies (12 V)	1	Power Supplies (24 V)	1
Network Switches	1	Routers	1
Circuit Breakers (10 A)	1	Turnkeys	1
Power Strips	1	CompactLogix PLCs	1
Siemens S7-300 PLCs	1	ControlLogix PLCs	1
Y-Boxes	1		

on the control panel at a guard station in a jail. Indicator lights are controlled based on inputs from a sensor that detects if the cell door is secure. Because the exercise platform does not have actual doors, this sensor is simulated in the

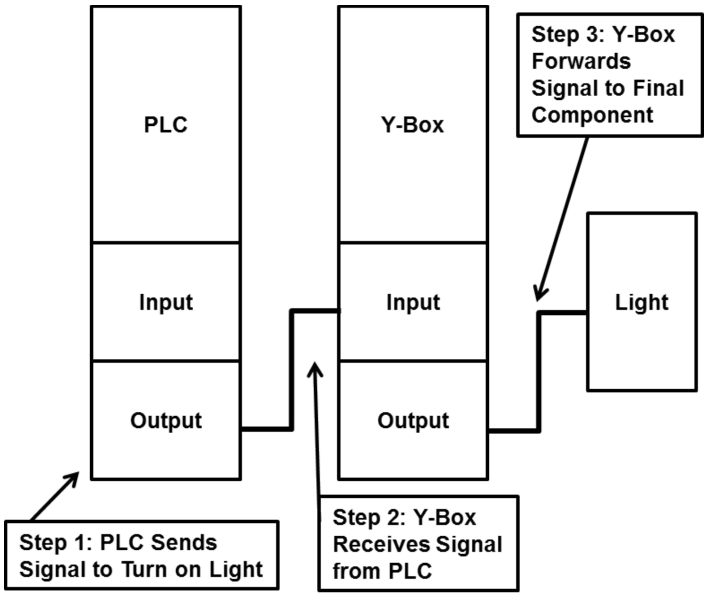

Figure 2. Wiring diagram for lights.

Y-Box code so that the Y-Box sends the sensor signals to the programmable logic controller. This is the only simulated component in the training platform. Interested readers are referred to [16] for a detailed description of the Y-Box.

The first programmable logic controller is a CompactLogix model L23E. The second is a Siemens S7-300 with one digital input module and one digital output module. The third is a ControlLogix programmable logic controller that also has one digital input module and one digital output module. Additionally, the ControlLogix programmable logic controller does not have a built-in Ethernet or CPU module; therefore, a Logix5555 CPU module and an EWEB Ethernet module are included in the seven-slot chassis. The Y-Box consists of a CPU module with one digital input module and one digital output module. The five electromechanical relays are implemented using a Micrologix programmable logic controller.

Wiring. To take full advantage of the Y-Box technology, the physical components are not wired directly to the programmable logic controller. Instead, different wiring schemes are adopted. For some applications, the Y-Box can be thought of as a "man-in-the-middle" device that receives electrical signals from the programmable logic controllers and other components, and forwards the signals to their destinations. The wiring schemes used for the lights and pushbuttons are shown in Figures 2 and 3, respectively.

The cabinet locks are wired differently because the Y-Box cannot provide sufficient electrical current to disengage the lock. In this case, the Y-Box is used

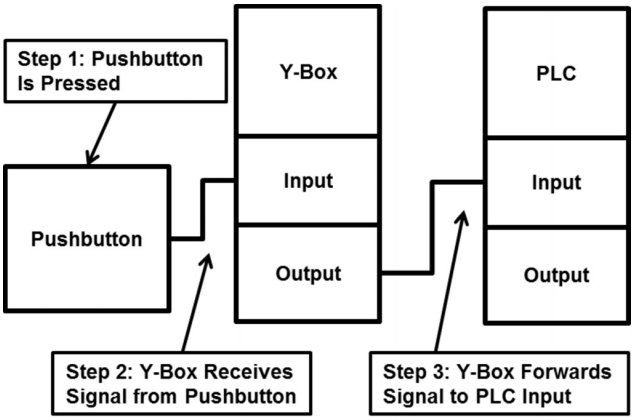

Figure 3. Wiring diagram for pushbuttons.

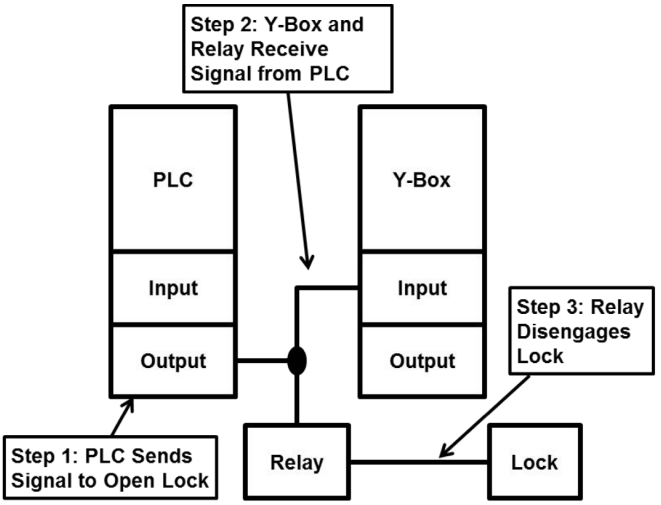

Figure 4. Wiring diagram for locks.

to monitor the signal on the line between the programmable logic controller and the relay, which ultimately powers the lock. This is accomplished by daisy chaining the programmable logic controller outputs from the relay to the Y-Box. Figure 4 shows the wiring diagram.

The other wiring challenge involves connecting all three programmable logic controllers as a single set of components. This requires the inputs and outputs of the three programmable logic controllers to be synchronized and wired together. Figure 5 shows the wiring diagram for an indicator light. The outputs of all

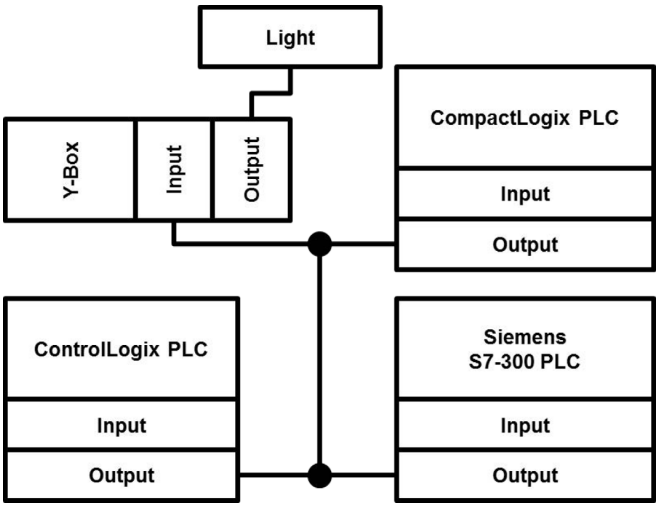

Figure 5. Wiring diagram for programmable logic controller inputs and outputs.

three programmable logic controllers are tied together, ultimately leading to a single wire that is connected to the Y-Box input module.

Programmable Logic Controller Selection. The value of having three programmable logic controllers in a single platform is lost if they cannot all assume full control over the components. Once again, the Y-Box can be leveraged to control the flow of electricity to an individual programmable logic controller while denying power to the other programmable logic controllers. This is accomplished using solid state relays controlled by a Y-Box digital output. When the solid state relay receives the control signal from the Y-Box, power is allowed to flow through the relay to its corresponding programmable logic controller, activating the device. This is the case for the ControlLogix and Siemens S7-300 programmable logic controllers. The CompactLogix programmable logic controller is slightly different from the other two controllers because it operates on 24 V DC. In this case, the relay controls power to a 24 V power supply that, in turn, powers the CompactLogix programmable logic controller. Figure 6 shows the wiring of the relays. Note that the figure is simplified and does not show the 24 V power supply.

3.2 Exercise Layout

Figure 7 shows a possible exercise layout. The following paragraphs describe the functions of each segment of the three tables in the exercise layout.

White Cell Table. An effective white cell should be aware of all the activities performed by the training participants. The simulation terminal is a

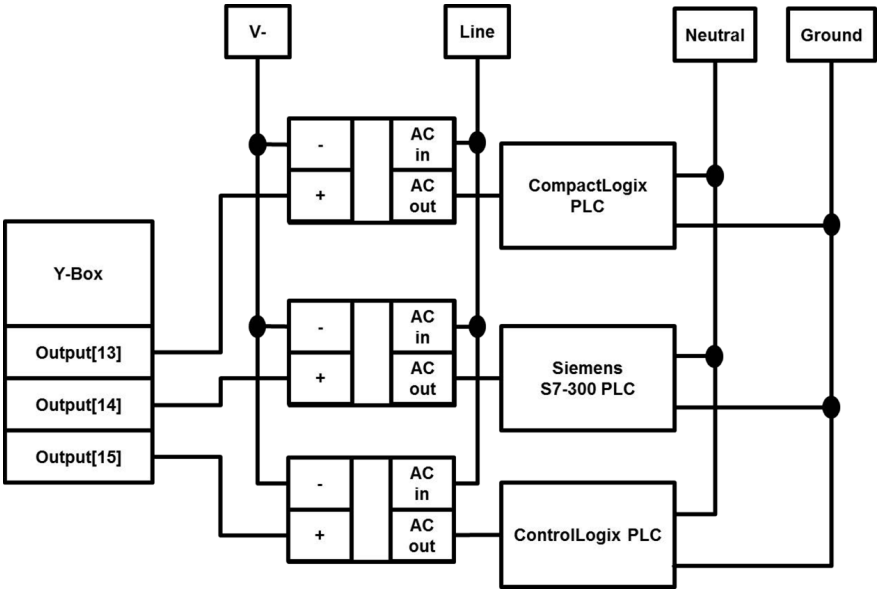

Figure 6. Wiring diagram for programmable logic controller selection.

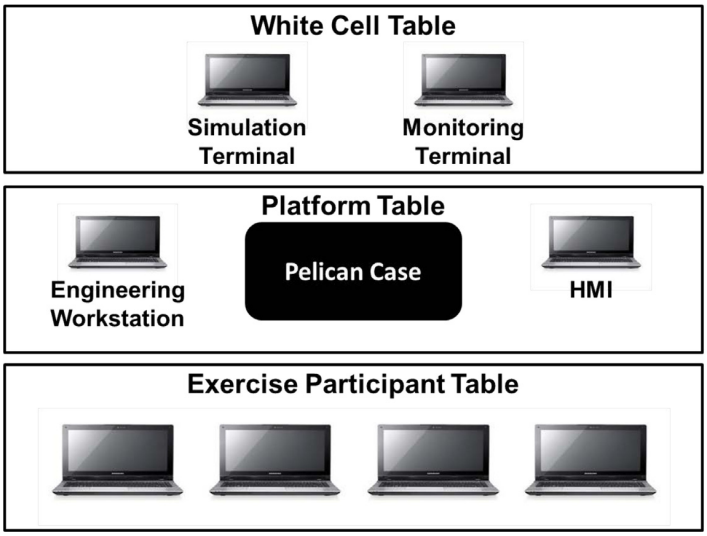

Figure 7. Exercise layout.

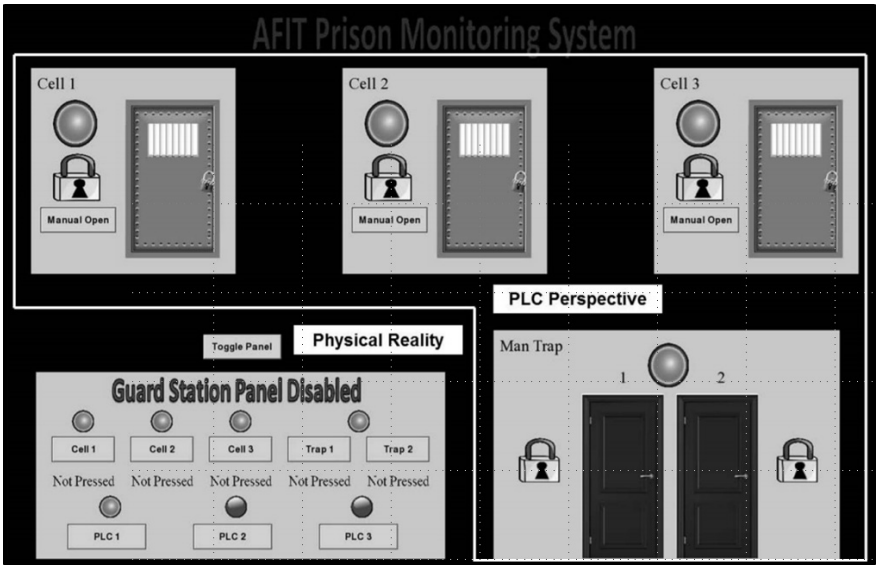

Figure 8. White cell view.

machine that runs the Y-Box software implemented in Python. The monitoring terminal runs network monitoring software and is connected to a mirrored port on the switch to capture all the traffic during the exercise. During the exercise, the white cell should watch the engineering workstation, human-machine interface, network traffic and participants. Furthermore, the white cell should watch the Y-Box software.

If a training scenario involves malware that fools the human-machine interface, it would be difficult for the white cell to maintain awareness of the state of the physical system during the training. The Y-Box overcomes this problem because it is aware of the true states of the locks, lights and pushbuttons. Figure 8 shows the Y-Box software view of the system with the physical reality of the system as well as the system from the perspective of the programmable logic controller. The figure also shows the programmable logic controller selection buttons that dictate which controller is active at any given time. It should be noted that the buttons in the software are capable of overriding the physical components in the Pelican case, enabling the white cell to manage all the aspects of the exercise at all times.

Platform Table. An engineering workstation and a human-machine interface operate beside the platform. The training platform is contained in a Pelican 1610 case. Inside the case is a fully-functioning replica of a guard station panel that closely mimics what would be found in an actual jail. Additionally, the case contains five cabinet locks, three of which represent jail cells and two that serve as mantraps. Each jail cell has a corresponding light that indicates

whether or not the cell is secure. The mantrap has only one light that indicates it is secure only when both its doors are closed and locked. The programmable logic controllers are connected to a network switch housed in the Pelican case.

Exercise Participant Table. Training participants are seated at the exercise participant table within sight of the training platform as shown in Figure 7. Laptops are provided with standard security tools (e.g., Kali Linux and Security Onion) as well as virtual machines containing proprietary software applications that interact with the programmable logic controllers. A participant may also bring any tools that he/she feels are appropriate to the exercise. From his/her table, the participant is connected to a network switch in the Pelican case and is free to interact with the platform to complete the assigned tasks. Note that the layout can be adjusted to accommodate different rooms and table sizes, and additional network switches can be added to accommodate more participants.

4. Training Scenario

One of the most important steps in securing an industrial control system is to properly segment the control network [15]. This simple task demonstrates the different implementations of similar features by the three programmable logic controllers. For this reason, a beginner-level scenario was first designed for the multi programmable logic controller training platform.

Because this scenario is intended to demonstrate the differences in programmable logic controller implementations, the scenario is simplified in several ways. First, the initial IP addresses of all three programmable logic controllers are the same (`192.168.108.205`). The new IP addresses that the participants load on the programmable logic controllers are also the same (`10.1.4.205`). Additionally, the participants need not concern themselves about whether or not changing the IP address would impact the functionality of other industrial control system components. In this scenario, it is assumed that all the other issues regarding components that are dependent on the programmable logic controller IP address have already been reconciled. More complicated scenarios can be developed to demonstrate the potential second- and third-order effects that can occur from this process.

The final scenario simplification is that no password protections exist for any of the files. In a real-world environment, it is reasonable to expect that a cyber first responder would be provided the necessary access by an asset owner to perform the tasks. While credentials are required by the ControlLogix administrative web server, they were reset to the factory-default credentials for demonstration purposes. Finally, the training scenario is implemented using the framework proposed by Yoon et al. [17].

The following are the details of the training scenario:

- **Objective:** Isolate a programmable logic controller that is located in an improperly segregated network.

- **Description:** The participant uses the relevant software and appropriate technique to change the programmable logic controller IP address from `192.168.108.205` to the new IP address `10.1.4.205`.
- **Type:** Network reconfiguration.
- **Evaluation Criteria:**
  - Identify the relevant software within five minutes.
  - Identify the appropriate technique for updating the IP address within ten minutes.
  - Update and confirm the new IP address within fifteen minutes.
  - Perform all the activities with minimal programmable logic controller downtime.
- **References:** NIST SP 800-82, Rockwell Automation EWEB module documentation, Siemens S7-300 documentation and Rockwell Automation CompactLogix documentation.

4.1 Segmentation Using a CompactLogix PLC

The first task for the participant is to interact with the CompactLogix programmable logic controller. Updating the IP address of this programmable logic controller involves the following steps:

- **Step 1:** Open the appropriate RSLogix5000 project file and access the Ethernet module properties.
- **Step 2:** Under the Port Configuration tab, enter the new IP address in the appropriate field and click Set. Confirm the update in the dialogue windows that appear.
- **Step 3:** Ensure connectivity to the new IP address (this may require routing or changing the IP address of the engineering workstation).

Step 1 requires the identification of the RSLogix5000 software. Step 2 involves the identification and update of the IP address. Step 3 ensures that the programmable logic controller is available. Since the CompactLogix programmable logic controller can have its IP address updated without downtime, the participant should receive a lower evaluation if the programmable logic controller resets or faults. Figure 9 shows the relevant dialogue window for updating the IP address.

4.2 Segmentation Using a Siemens PLC

After the participant has completed the assigned task on the CompactLogix programmable logic controller, the instructor switches control to the Siemens programmable logic controller. After the programmable logic controller has

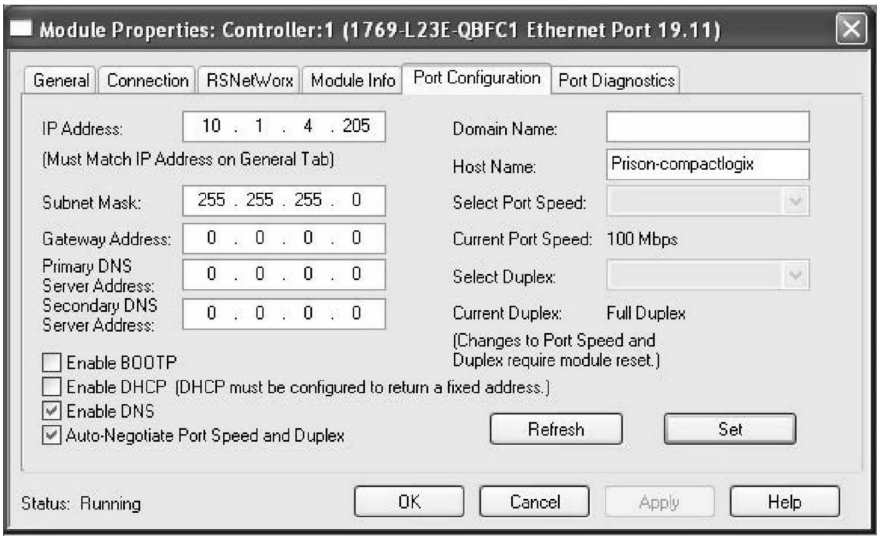

Figure 9. IP address update of the CompactLogix PLC using RSLogix5000 software.

booted, the participant must change the IP address of the Siemens controller to an isolated subnet. This is the first time that the participant is exposed to the differences between the programmable logic controllers. In particular, the programming environment for the Siemens programmable logic controller is different from that of the CompactLogix controller.

The following steps are required to complete the task on the Siemens programmable logic controller:

- **Step 1:** Open the SIMATIC project file.
- **Step 2:** Access the HW Config tab in the SIMATIC software and navigate to the Object Properties of the PN-IO module.
- **Step 3:** Under the General tab, select Properties and enter the new IP address as shown in Figure 10.
- **Step 4:** Download the new configuration to the programmable logic controller using the old IP address as the target station.
- **Step 5:** Ensure connectivity to the new IP address (this may require routing or changing the IP address of the engineering workstation).

Step 1 involves the identification of the SIMATIC software. Steps 2 through 4 involve the identification of the appropriate technique to update the IP address. Step 5 ensures that the programmable logic controller completes the download successfully with minimal downtime.

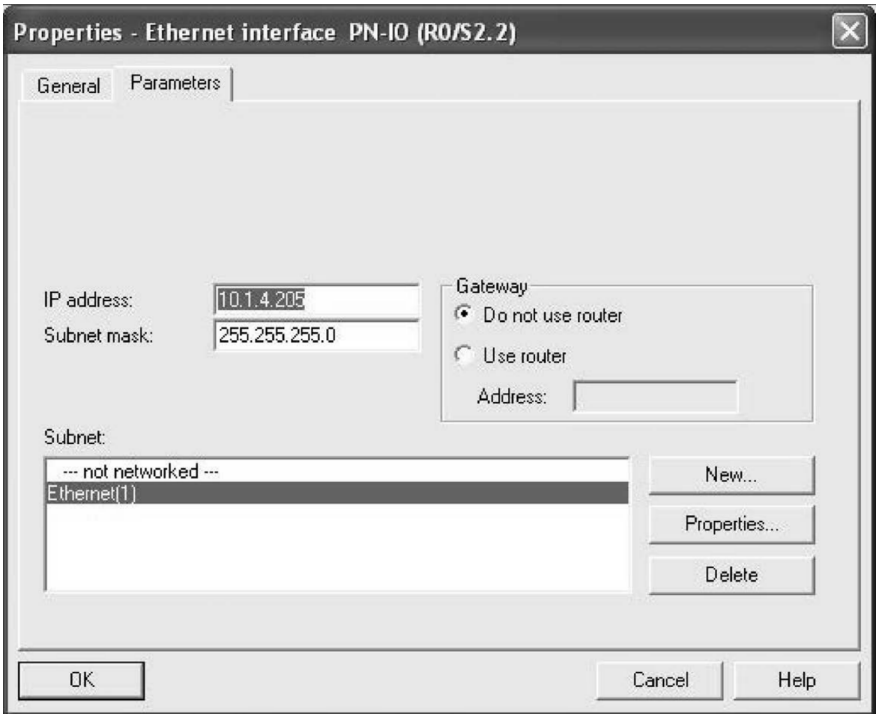

Figure 10. IP address update of the Siemens S7-300 PLC using SIMATIC software.

4.3 Segmentation Using a ControlLogix PLC

The participant now performs the required tasks using an implementation that is unique to the ControlLogix programmable logic controller. The ControlLogix programmable logic controller is equipped with a 1756-EWEB Enhanced Web Server Module that provides an administrative web interface to manage the programmable logic controller.

The following steps are involved:

- **Step 1:** Open a web browser and navigate to the IP address of the programmable logic controller.
- **Step 2:** Open the Network Configuration tab, enter the new IP address in the appropriate field and apply the changes.
- **Step 3:** Confirm that the new address is correct. Upon completion, a message is displayed that notifies the participant of the new IP address.
- **Step 4:** Ensure connectivity to the new IP address (this may require routing or changing the IP address of the engineering workstation).

Step 1 requires the participant to identify the web interface provided by the EWEB module. Step 2 involves the identification and application of the appropriate technique to perform the update. Steps 3 and 4 confirm that the change is successful. There should be no downtime when performing this task on the ControlLogix programmable logic controller.

4.4 Scenario Selection and Alternate Scenarios

The network segmentation scenario was chosen because it effectively demonstrates how different programmable logic controllers often require different techniques to perform the same task. These differences emphasize the value of a cyber first responder having experience on a variety of programmable logic controllers. The scenario incorporates several of the training items proposed by Butts and Glover [3] and implemented in the framework of Yoon et al. [17].

Note that segmenting a network is only one of many tasks that a cyber first responder may need to complete in his/her line of work and it is certainly not a difficult task. Other examples such as modifying ladder logic programs, updating firmware and applying patches are also unique to different programmable logic controllers and have varying levels of difficulty. Because the training platform incorporates real programmable logic controllers, a number of scenarios, including scenarios involving advanced topics, could be implemented with minimal reconfiguration.

Two scenarios that showcase the flexibility of the multi programmable logic controller platform are described below.

Analysis of a Malicious Implant in PLC Firmware

- **Objective:** Reverse engineer the firmware to identify and analyze a malicious implant.
- **Description:** The participant uses the relevant software and appropriate techniques to extract programmable logic controller firmware from the device and identifies malicious code given the correct version of the firmware. The participant then determines the exact functionality and purpose of the malicious code.
- **Type:** Reverse engineering.
- **Evaluation Criteria:**
  - Identify the malicious code within 45 minutes.
  - Restore the programmable logic controller firmware within 20 minutes.
  - Analyze the malicious code within 90 minutes.
- **References:** Rockwell Automation ControlLogix documentation, Siemens S7-300 documentation, Rockwell Automation CompactLogix documentation.

The reverse engineering scenario further emphasizes the differences between programmable logic controllers by requiring a participant to extract and analyze firmware from the devices (see [1] for details about reverse engineering industrial control devices). The scenario also brings up the important point that there are often similarities between programmable logic controllers. Specifically, the CompactLogix and ControlLogix programmable logic controllers have very similar firmware despite being different models. The reverse engineering scenario can be implemented with malware samples of varying complexity to accommodate and/or enhance participant abilities.

Digital Forensics of a Malfunctioning PLC

- **Objective:** Determine the cause of a malfunctioning programmable logic controller.
- **Description:** The participant uses the relevant software and appropriate techniques to identify the root cause of the programmable logic controller behavior.
- **Type:** Digital forensics.
- **Evaluation Criteria:**
  - Collect sufficient data to perform digital forensics within 30 minutes.
  - Identify the cause of the malfunction within 45 minutes.
  - Identify the corrective action within 60 minutes.
- **References:** Rockwell Automation ControlLogix documentation, Siemens S7-300 documentation, Rockwell Automation CompactLogix documentation.

The digital forensic scenario involves similar tasks as the reverse engineering scenario. It also shows that the process for conducting digital forensics on industrial control systems is identical for different programmable logic controllers (see [6] for details about this process). Despite using the same process, the data being analyzed (e.g., ladder logic program, network traffic and log files) would be different because of the operational differences between the programmable logic controllers. These operational differences mean that a cyber first responder in a real-world situation will have to focus on specific, contextualized pieces of information to effectively analyze the root cause of a malfunctioning programmable logic controller. The difficulty of this scenario can be modulated by inducing different types of programmable logic controller malfunctions ranging from simple faults to advanced malware infections. The scenario can be repeated multiple times with different symptoms to increase the participant's exposure to a variety of malfunctions.

5. Results

This section describes the principal results pertaining to training platform development.

5.1 Hardware Verification

The initial debugging of the wiring, Y-Box code and programmable logic controller code involved interacting with the physical components mounted in the Pelican case and confirming that the Y-Box and programmable logic controller behaved as intended. This process revealed that some of the variables had been coded incorrectly in the ladder logic. These variables needed their memory addresses reassigned to correct their mapping to the programmable logic controller inputs and outputs. The Y-Box software was also verified, confirming the behavior of the physical components and that the software could override the physical components to control the case autonomously.

5.2 Reliability Test

After confirming that the components were behaving correctly, an automated Python script tested the reliability of the training platform. The test involved the following steps:

- **Step 1:** Select the programmable logic controller.
- **Step 2:** Power up the selected programmable logic controller.
- **Step 3:** Wait 25 seconds for the programmable logic controller to activate.
- **Step 4:** Test all the buttons, locks and lights for functionality.
- **Step 5:** Shut down the programmable logic controller.
- **Step 6:** Reset the Y-Box parameters.

Initial runs of the reliability test encountered failures because the Python test code sent commands too quickly, which did not provide the Y-Box with adequate time to update its inputs and outputs. This issue was resolved by including "wait" commands of 25 seconds for the programmable logic controller to boot and varying amounts of time between 0.4 and 2.0 seconds for other functions (e.g., button presses, lock status updates and indicator light updates).

Step 4 is the key part of the reliability test. This step starts with the first jail cell and simulates a button press. The script then checks that the programmable logic controller responds appropriately before repeating the process for the other two cells. Next, the test code evaluates the mantrap by testing every possible combination of button presses and confirming the responses. Finally, it simulates a button press on the cell once again with the panel disabled. In this situation, the lock should not disengage and the test is considered to have failed if it does.

Table 2. Reliability test results.

Controller	Trials	Failures
CompactLogix PLC	50	0
Siemens S7-300 PLC	50	0
ControlLogix PLC	50	0
Total	150	0

The wiring scheme with the Y-Box enables electrical signals to be sent to the programmable logic controller without having to receive signals from the buttons. Furthermore, the state of the panel turnkey can be overridden by the Y-Box itself. These enable each of the functions to be simulated by the Y-Box alone. In the future, the test can be fully automated in a manner that is transparent to the programmable logic controller because the controller receives the same signals as it would under normal operation. To prevent a failure in one iteration from impacting the results of the next iteration, Step 6 resets all the Y-Box values to default values. A total of 150 iterations were performed, with each programmable logic controller tested 50 times. Table 2 shows the reliability test results.

5.3 Timing Test

Incorporating multiple programmable logic controllers in a single platform is useless if the switching between the programmable logic controllers takes a prohibitive amount of time. Ideally, control of the system should be switched from one programmable logic controller to another within the amount of time that it takes for the participant to be prepared for the next task. To evaluate this metric, the time required for each programmable logic controller to fully power up was measured and recorded by an automated Python script, which performed the following steps:

- **Step 1:** Select the programmable logic controller.
- **Step 2:** Send power to the programmable logic controller and start the timer.
- **Step 3:** Send the input command to the programmable logic controller.
- **Step 4:** Wait for the programmable logic controller to react to the input and stop the timer upon completion.
- **Step 5:** Shut down the programmable logic controller.
- **Step 6:** Reset the Y-Box parameters.

In the timing test, it is only necessary to examine the amount of time that it takes for the programmable logic controller to become responsive to an input.

Table 3. Programmable logic controller startup times (seconds).

Controller	Minimum	Maximum	Mean	Std. Dev.
CompactLogix PLC	19.547	19.688	19.629	0.030
Siemens S7-300 PLC	14.782	15.172	15.060	0.062
ControlLogix PLC	4.797	4.843	4.816	0.010

Table 3 presents the results of the timing test, which were also determined over the course of 150 trials (50 trials per programmable logic controller). The results show that the programmable logic controllers have significantly different boot times, but are very consistent across all the trials.

5.4 Functional Analysis Criteria

The multi programmable logic controller training platform is designed to meet the following criteria:

- Incorporate physical components.
- Incorporate cyber manipulation principles.
- Incorporate response coordination techniques.
- Provide hands-on experience.
- Implement effective training scenarios with measurable training evaluation metrics.

The replication of the jail system, including the programmable logic controllers running realistic ladder logic programs and the pushbuttons, locks, lights and turnkey, enables a training participant to experience many of the physical components involved in a real-world system. This addresses the need for cyber first responders to understand the physical processes underlying an industrial control system.

By creating scenarios that incorporate concepts such as reverse engineering and digital forensics, cyber manipulation principles can be effectively taught to cyber first responders. Participants can be exposed to topics like access vectors, vulnerability analysis, implanting malware, manipulating physical processes and defensive mechanisms, all of which are considered to be cyber manipulation principles [3].

Skills involved in response coordination include the ability to prioritize system components, identify attacks and understand the steps required to appropriately defend and restore a system to normal operation. Each of these skills is practiced in some way by the scenarios described in this work and can be enhanced by designing alternate scenarios using the training platform.

The entire training platform is self-contained and all the relevant software is embodied in easily-recoverable virtual machines. This enables training participants to have full access to the system for hands-on exercises without being concerned about potential damage to the system. Cyber first responders can benefit most from hands-on exercises that give them experience with realistic systems that incorporate real hardware.

5.5 Limitations

The multi programmable logic controller platform has certain limitations. First, the platform does not incorporate any analog components. Analog signals are more complicated than digital signals from a programming perspective and should be incorporated to enhance learning experiences.

Another limitation of the platform is the scale of the replica system. In a full-sized jail, there are many more components, including additional doors and alarms. The design choice leads to the trade-off between training platform cost and scale. The final limitation is that the training platform is limited to the use of one programmable logic controller at a time.

6. Conclusions

Effective industrial control system platforms are necessary for cyber first responder training. Unfortunately, most testbeds are designed for research and development activities and are not available for training purposes. Furthermore, testbeds developed for training purposes tend to be very expensive or substitute device simulations in place of genuine components. Ideal testbeds incorporate full-scale, fully-operational industrial control systems with training scenarios that impart the unique skills needed by cyber first responders to operate in real-world environments. The cost of such a testbed is prohibitively high; however, the multi programmable logic controller training platform developed in this research can impart many of the desired skills at a fraction of the cost. Furthermore, the multi programmable logic controller training platform can support scenarios ranging from basic tasks such as changing an IP address to advanced tasks involving reverse engineering and digital forensics.

The multiple programmable logic controllers incorporated in the platform provide opportunities for trainees to experience different devices, protocols and programming environments. Similar to medical students, who go through a rigorous curriculum with hands-on, real-world learning experiences, the multi programmable logic controller training platform enables cyber first responders to gain valuable hands-on experience with real control systems to significantly enhance their cyber defense skills.

Note that the views expressed in this chapter are those of the authors and do not reflect the official policy or position of the U.S. Air Force, U.S. Army, U.S. Department of Defense or U.S. Government.

Acknowledgement

This research was partially supported by the U.S. Department of Homeland Security Industrial Control Systems Cyber Emergency Response Team (ICS-CERT).

References

[1] Z. Basnight, Firmware Counterfeiting and Modification Attacks on Programmable Logic Controllers, M.S. Thesis, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, 2013.

[2] A. Bauer, Talking with your doctor about prognosis, *Cancer.Net*, August 14, 2014.

[3] J. Butts and M. Glover, How industrial control system security training is falling short, in *Critical Infrastructure Protection IX*, M. Rice and S. Shenoi (Eds.), Springer, Heidelberg, Germany, pp. 135–149, 2015.

[4] R. Candell, T. Zimmerman and K. Stouffer, An Industrial Control System Cybersecurity Performance Testbed, NISTIR 8089, National Institute of Standards and Technology, Gaithersburg, Maryland, 2015.

[5] Department of Psychiatry, New York University School of Medicine, Medical Student Education Program in Psychiatry, New York University, New York (`www.med.nyu.edu/psych/education/medical-student-education`), 2017.

[6] L. Folkerth, Forensic Analysis of Industrial Control Systems, InfoSec Reading Room, SANS Institute, Bethesda, Maryland (`www.sans.org/reading-room/whitepapers/forensics/forensic-analysis-industrial-control-systems-36277`), 2015.

[7] Idaho National Laboratory, INL Cyber Security Research: Defending the Network Against Hackers, Fact Sheets: 21st Century Science and Technology, Idaho Falls, Idaho (`www.inl.gov/research/inl-cyber-security-research`), 2014.

[8] Idaho National Laboratory, University Partnerships, Idaho Falls, Idaho (`www.inl.gov/inl-initiatives/education`), 2016.

[9] International Information System Security Certification Consortium ($(ISC)^2$), $(ISC)^2$ Information Security Certification Programs, Clearwater, Florida (`www.isc2.org/credentials/default.aspx`), 2016.

[10] Sandia National Laboratories, Distributed Energy Technology Laboratory, Albuquerque, New Mexico (`energy.sandia.gov/wp-content/gallery/uploads/DETL_Factsheet_SAND2010-3643_Aug2011.pdf`), 2011.

[11] Sandia National Laboratories, SCADA Testbeds, Albuquerque, New Mexico (`energy.sandia.gov/energy/ssrei/gridmod/cyber-security-for-electric-infrastructure/scada-systems/testbeds`), 2016.

[12] SANS Institute, ICS Training Courses, Bethesda, Maryland (ics.sans.org/training/courses), 2017.

[13] SANS Institute, SEC562: CyberCity Hands-On Kinetic Cyber Range Exercise, Bethesda, Maryland (www.sans.org/course/cybercity-hands-on-kinetic-cyber-range-exercise), 2017.

[14] E. Skoudis, How to build a completely hackable city in five steps: And why you should build your skills in this arena, presented at *SANS Pen Test Hackfest*, 2013.

[15] K. Stouffer, J. Falco and K. Scarfone, Guide to Industrial Control Systems (ICS) Security, NIST Special Publication 800-82, National Institute of Standards and Technology, Gaithersburg, Maryland, 2011.

[16] J. Yoon, Framework for Evaluating the Readiness of Cyber First Responders for Industrial Control Systems, M.S. Thesis, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, 2016.

[17] J. Yoon, S. Dunlap, J. Butts, M. Rice and B. Ramsey, Evaluating the readiness of cyber first responders responsible for critical infrastructure protection, *International Journal of Critical Infrastructure Protection*, vol. 13, pp. 19–27, 2016.