A Remotely Accessible, Configurable, Instrumented ICS Lab for Attack, Defend, and Forensics Research and Education

James H. Jones, Jr., and Peggy S. Brouse
George Mason University

Abstract

Industrial Control Systems drive many of our critical infrastructures, and the security of these systems directly affects the availability and reliability of services provided by those critical infrastructures. Most ICS elements were not designed with security in mind, and in fact were never intended for connection to untrusted networks. We are now engaged in a fast-paced and very important race to secure these systems from both defensive and forensic points of view. Preparing students for this work requires a thorough grounding in scientific principles, a full understanding of engineering processes, and experiential learning with actual devices and environments. While an engineering education address the first two requirements, in this work we aim to address this third requirement for realistic hands-on activities. We constructed a remotely accessible, instrumented, and re-configurable ICS system on which students can learn by performing offensive, defensive, and forensic cyber security exercises.

Keywords

cyber security; digital forensics; cyber physical systems; industrial control systems; programmable logic controller

Introduction

Industrial Control Systems (ICS) are ubiquitous in production environments, including but not limited to those supporting critical infrastructures such as defense, energy, chemical, water, oil, gas, transportation, and others. Increasingly, the devices making up an ICS are connected over IP networks and run full or derivative mainstream operating systems, exposing the devices and the elements they control to more vulnerabilities and remote attackers[1,2]. This increased attack surface, combined with motivated, skilled, and well-funded adversaries, puts our critical infrastructures and the people and assets that depend on them at greater risk than at any time in our history. Securing these systems and defending them against attackers requires an understanding of how the systems work, what vulnerabilities are exposed, and how they are attacked in practice[3]. Such an understanding cannot be acquired through textbooks or demonstrations, but rather is best learned in a hands-on, operationally realistic environment using real hardware and software where students can attack and defend these systems without fear of actual damage to equipment or personnel. Unfortunately, such laboratory environments are expensive and relatively few[4], and the handful of portable training kits are limited in scope and still require colocation of student and kit. We designed and implemented a remotely accessible ICS security lab, consisting of actual ICS hardware and software controlling real but harmless devices. Students, instructors, and practitioners can remotely configure the lab to select different network architectures, access points, control workstations, PLCs, and devices. Users then access

the lab systems remotely to run penetration tests, experiment with attacks and exploits, implement detection and defensive measures, and capture data for subsequent forensic analysis. The lab as implemented includes multiple cameras for visual monitoring, network and system instrumentation, and 12 TB of network attached storage for virtual machine images and data collection.

Related Work

Considerable work has been performed to establish learning and exercise environments for cyber attack and defend activities. These environments can be loosely classified as virtual and physical, where virtual environments may be further subdivided into simulated and emulated.

Simulated environments use models of the systems being exercised, where such models necessarily exhibit a subset of the actual system behavior and have inherent fidelity limitations[5,6]. Emulators typically have higher fidelity but still not 100%, and for fields such as cyber-physical security such limitations may matter[7]. Simulated cyber security exercises frequently show up in the form of games, where the game designer has encoded certain behaviors independent of an actual running system. Such simulations have considerable value at the early educational stages, or to demonstrate a particular point, but they fall short when exploring outside the parameters of the game. Emulators are closer representations of the target system and enable a more exploratory learning environment[8], although it can be difficult to know the limitations of that fidelity. With a simulated game the parameters are clear, but an emulated environment may not specify parameters. A simple example might be an exercise involving a mobile device emulator. If attacking a specific application on the emulated device, the code may execute as expected and the attack works or not as it would on a physical device. However, if exploring the residual digital artifacts remaining on the media after the failed or successful attack, the physical media and the device interaction with it is crucial[9]; an emulated flash memory device running on a host SSD or spinning magnetic disk will likely not behave identically to the on board flash in a physical device. Emulated environments in the form of virtual machines are popular in cyber security and engineering education as they provide rapid provisioning, version control, sandboxing, and preservation at a low cost relative to physical systems. For example, we can quickly provision a few dozen virtual machines for a blue-on-red exercise, and allow the students to attack at will knowing that boundaries are in place, no systems will be permanently damaged during the exercise, we can instrument the whole thing, and we can restore systems to any prior state in a few minutes. Virtual machine hypervisors emulate the physical hardware required by the running guest OS or device, and do so in such a way that it can be difficult to establish whether a given system is running in a virtual or bare metal environment. Hypervisor guests are not limited to full operating systems, as virtual machines have been developed for PLCs[10]. As with any emulation, fidelity eventually breaks down and it is incumbent upon the instructor or exercise designer to know where that breakdown happens and take steps to mitigate the risk of incorrect results or false conclusions.

Physical environments for education are common in other fields such as chemistry or biology, where students work with real chemicals or animals to facilitate the learning process, although this is changing with the availability of high fidelity simulations for activities such as animal dissection. However, just as we would not want to be the first "real animal" that our surgeon ever cut, we also don't want a critical infrastructure system to be the first "real ICS" that a recent

cyber security graduate touches. The cyber security field grew up with technological availability for simulations and emulations, and in fact such availability was a catalyst for the field's growth since learning environments were cheap and easy to obtain. However, this virtual training environment has come at a cost, and the cost is magnified as we move to an ever increasingly cyber-physical world. The challenges in cyber security are no longer corralled in the processor, memory, and disks of a computer in an office or the networks that connect them, but now live among us in our cars, planes, houses, factories, and power plants. In this exposed environment, physical access matters, and physical access can't be simulated or emulated. In the ICS domain, training kits[11,12] are available which provide one or more devices, a PLC, and an interface and software to communicate with the PLC and control the devices. These are excellent learning tools as far as they go, but now must be physically transported to the students (or vice versa), and they lack flexibility and instrumentation. Similarly, various ICS (SCADA) testbeds have been established, but these are targeted at a particular industry or application, e.g. energy[13] or man-in-the-middle attacks[14]. In this work, we aim to leverage the benefits of real, physical devices while addressing some of the key limitations.

Lab Architecture, Construction, and Operation

The ICS lab was designed to be remotely accessible, sandboxed, configurable, and physically realistic. The lab is also designed to support red and blue team operations as well as forensic activities[15], and the lab is remotely configurable and instrumented for observation, data collection, artifact storage, and analysis. The lab project is funded by a grant from the US Army Reserves Cyber Soldier Development - Cyber Private Public Partnership Initiative (P3i) BAA-004-2016, award # H98230-16-1-0356.

The lab architecture is shown in Figure 1 and components are explained in the bullets that follow:

Devices: The input and output devices are switches and lightboards as well as actual hardware with moving parts disabled or removed and replaced with sensors to indicate speed, state, etc. In this way, lab users can manipulate the devices and observe their state without risk of physical destruction. These extra sensors are exposed to the monitor system and not the ICS server or PLCs.

PLCs: The PLCs are actual hardware that can control the devices. The current version of the lab uses two each of the following: Allen-Bradley MicroLogix 1000, Siemens S7 (one -200 and one -1200), Raspberry Pi, and OpenPLC (Arduino).

ICS server: The ICS server hosts the virtual machines that control the PLCs and hence the devices through PLC software (Step7, RSLogix, etc.) or other means. The ICS server VMs are available to lab users for offensive, defensive, and forensic purposes.

Configuration Workstation: The configuration workstation controls the lab configuration, specifically which VM is connected to which PLC and associated devices. This configuration is controlled through a combination of hardware and software, labeled "A-B-C switch" in the figure, one between the ICS server and the PLCs and one between the PLCs and the devices. The configuration workstation also controls whether the PLCs are

directly connected to the LAN or are connected to the LAN only via a serial interface to the ICS server VM (both configurations are found in practice). The configuration workstation is accessible to remote lab administrators.

Attacker: The attacker system is also available to the lab users and provides a platform for them to attack the components of the ICS setup. This is designed to be the most realistic scenario, where an attacker has gained a foothold on the internal network but has not yet compromised the control workstation or PLC. The attacker system contains virtual machines for common attack platforms (e.g., Kali Linux) as well as standard workstations and allows lab users to instantiate and preserve new virtual machines as desired.

Monitor: The monitor system is accessible to lab administrators and lab users doing research. The monitor provides a means to monitor and capture network traffic, and to monitor device and other system states from an external or infrastructure point of view. The monitor system also provides a mechanism to capture the non-volatile storage and RAM of the devices, PLCs, and ICS VMs.

NAS device: A hardware de-duplicating NAS device provides 10TB of redundant disk space for local storage of VM templates and experimental data. Some of this storage is directly available to lab users for specific experiments.

Gateway Server: The gateway server acts as a firewall and handles all incoming requests, authenticates users and forwards the traffic to the appropriate lab system.

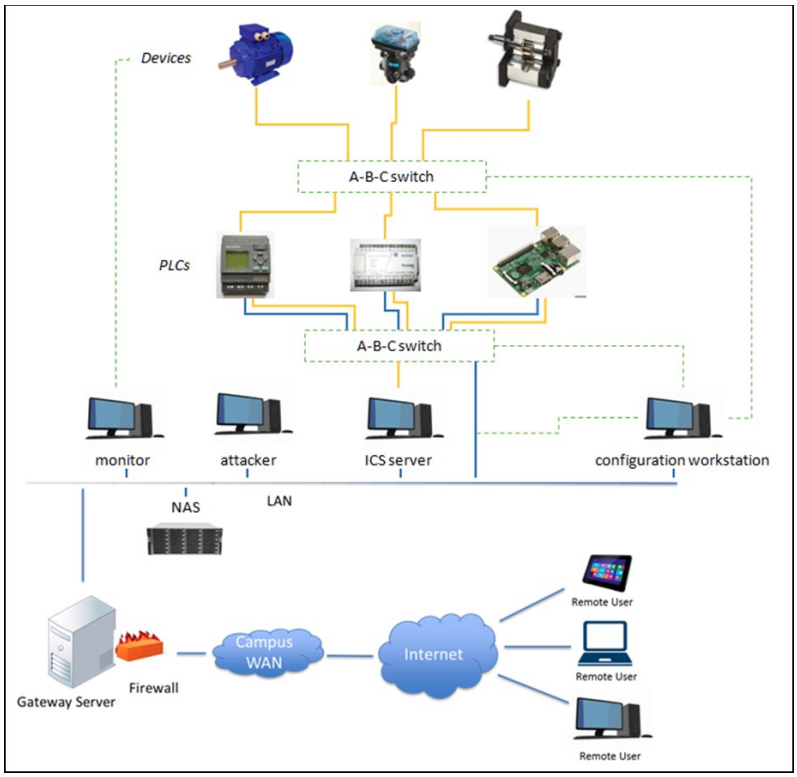

Figure 1: Lab Architecture

Sample Exercise

A student exercise was constructed and executed during the Fall semester 2017 and is described below to demonstrate the configuration and use of the lab. This exercise was conducted over two weeks as one of seven modules in a 1-credit 400-level Cyber Vulnerabilities Lab course. For this lab, students worked in 4-5 person teams and were provided access to a Windows 10 Control Workstation with a connected PLC (Allen-Bradley MicroLogix1000) and devices (I/O board with switches and lights) as depicted in Figure 2. In a real scenario, Control Workstation access is often gained via a phishing campaign or other exploit against another system, then the Control Workstation is breached; we assumed that step had already occurred. The lab consists of reconnaissance and manipulation activities, followed by some analytic work, and ending with considerations for how to attack (and defend) such a system.

Figure 2: Student Exercise Setup

The Control Workstation was preconfigured with Wireshark with USBPcap (for network and USB sniffing), RSLinx (for connecting to the PLC), RSLogix (for programming the PLC), and CMS (a camera app for viewing the I/O boards). Once connected, the students ran the Wireshark application on the local Ethernet interface to capture traffic and get a sense of the network they were on. Analysis of this pcap file was part of their submission. The students also ran Wireshark on the local USB interface to capture serial communications between the Control Workstation and the PLC. Analysis of this pcap file was also part of their submission.

Students then targeted the PLC. Using the local programming application (RSLogix), the students retrieved the program currently running on the PLC. Once they had the program, they could view and decipher the implemented ladder logic (Figure 4), comparing I/O states to those viewed with the camera (Figure 3). Setting the programming application to "Live" mode, they could manipulate the Input and/or Output values at will and without physically touching the switches. For example, the lab instructions guided the students to set a "Force ON" for Toggle 2, which turned the Green light (second from left) on although the toggle switched physically remained in the OFF position (Figure 5). In an attack scenario, this is analogous to turning on the pump or opening the gate without the operator touching the physical switch.

Finally, the students modified the running program offline to cross-wire the toggle switches so that Toggle 1 controlled Light 2, and Toggle 2 controlled Light 1. They then loaded the program and ran it live on the PLC. In an attack scenario, this is analogous to swapping the effect of two physical operator switches, e.g., turn on the ignitor in boiler 1 instead of boiler 2 and vice versa.

The remainder of the exercise occurred offline. The students were asked to analyze their captured network and USB/serial traffic as well as a provided binary ladder logic program, firmware image[16], and partial memory dump of the PLC. Finally, the students were asked to discuss at least three possible attack vectors to compromise and control the PLC equipment along with potential defensive measures for each.

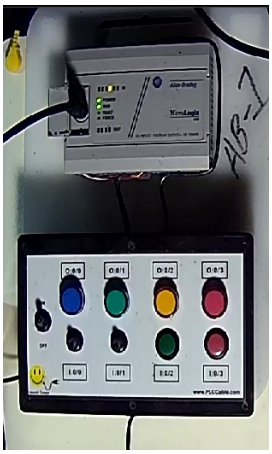

Figure 3: Camera View

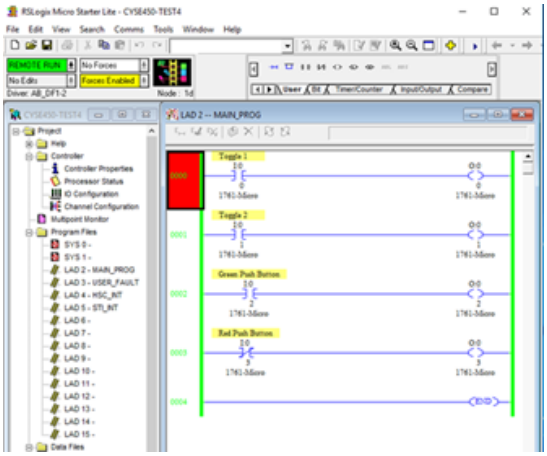

Figure 4: Running Ladder Logic Program

Figure 5: After Force ON

The students worked in teams, and their submissions reflected the diverse nature of the teams. All teams successfully navigated the guided online exercises in essentially the same manner. Their submission diverged considerably in terms of tools, techniques, and results for the offline analytic portion of the exercise, where tools ranged from simple hex editors and string extractors to attempted use of decompilers and research into protocol specifications. Similarly, the speculative final portion of the assignment yielded considerable variation in submissions. Various teams noted the physical limitation of PLC security and the exposure to unauthorized forcing outputs, the possibility of logic bombs and firmware modification, and exposure to memory manipulation and man in the middle attacks through unauthenticated communications.

Conclusions and Future Work

The ICS lab provides a flexible education platform in which instructors can craft specific exercises and students can gain hands on experience in real world scenarios with real equipment. Similarly, the lab provides a platform for security research by enabling rapid reconfiguration and data collection via integrated instrumentation.

Future work on the lab will focus on both capabilities and research. Regarding capabilities, instrumentation is currently manual and we have not developed a cross-platform method for imaging RAM in a PLC. We are investigating this to at least construct methods for capturing RAM from each type of PLC even if a cross-platform method is not found. We are also working to automate the instrumentation functions, imagining a "record" button which will capture network, protocol, firmware, processor, and RAM contents and activity at regular intervals for later analysis and in support of both educational and research activities.

Regarding research, we plan to explore attacks and possible defenses where a PLC's register (data) memory and/or execution (program) memory are manipulated to affect PLC operation. In addition to known attack vectors and malware, we are exploring additional techniques which leverage Ethernet, USB/serial, and PLC-to-PLC communications. Concurrently, we will use the lab's instrumentation capability to collect full and partial forensic artifacts associated with such activity to facilitate post-event detection and analysis in the field.

References

1 Spenneberg, Ralf, Maik Brüggemann, and Hendrik Schwartke. "Plc-blaster: A worm living solely in the plc." Black Hat Asia, Marina Bay Sands, Singapore (2016).

2 Govil, Naman, Anand Agrawal, and Nils Ole Tippenhauer. "On Ladder Logic Bombs in Industrial Control Systems." arXiv preprint arXiv:1702.05241 (2017).

3 Morris, Thomas H., et al. "Engineering future cyber-physical energy systems: Challenges, research needs, and roadmap." North American Power Symposium (NAPS), 2009. IEEE, 2009.

4 Aragó, Antonio Sánchez, et al. "SCADA Laboratory and test-bed as a service for critical infrastructure protection." Proceedings of the 2nd International Symposium for ICS & SCADA Cyber Security Research. 2014.

5 Barbosa, R.R.R., Sadre, R., Pras, A.: Difficulties in modeling SCADA traffic: a comparative analysis. In: Taft, N., Ricciato, F. (eds.) PAM 2012. LNCS, vol. 7192, pp. 126–135. Springer, Heidelberg (2012).

6 Queiroz, Carlos, et al. "Building a SCADA security testbed." Network and System Security, 2009. NSS'09. Third International Conference on. IEEE, 2009.

7 Alves, Thiago, Rishabh Das, and Thomas Morris. "Virtualization of Industrial Control System Testbeds for Cybersecurity." Proceedings of the 2nd Annual Industrial Control System Security Workshop. ACM, 2016.

8 Lemay, Antoine, José Fernandez, and Scott Knight. "An isolated virtual cluster for SCADA network security research." Proceedings of the 1st international symposium for ICS & SCADA cyber security research. 2013.

9 Van Vliet, Pieter, M-T. Kechadi, and Nhien-An Le-Khac. "Forensics in industrial control system: a case study." Conference on Cybersecurity of Industrial Control Systems. Springer International Publishing, 2015.

10 Thamrin, Norashikin M., and Mohd Mukhlis Ismail. "Development of virtual machine for Programmable Logic Controller (PLC) by using STEPS™ programming method." System Engineering and Technology (ICSET), 2011 IEEE International Conference on. IEEE, 2011.

11 Barrett, Michael. "The design of a portable programmable logic controller (PLC) training system for use outside of the automation laboratory." International Symposium for Engineering Education. 2008.

12 Burhan, I., S. Talib, and A. A. Azman. "Design and fabrication of programmable logic controller kit with multiple output module for teaching and learning purposes." Signal Processing and its Applications (CSPA), 2012 IEEE 8th International Colloquium on. IEEE, 2012.

13 Hahn, Adam, et al. "Development of the PowerCyber SCADA security testbed." Proceedings of the sixth annual workshop on cyber security and information intelligence research. ACM, 2010.

14 Yang, Yi, et al. "Man-in-the-middle attack test-bed investigating cyber-security vulnerabilities in smart grid SCADA systems." (2012): 138-138.

15 Wu, Tina, and Jason RC Nurse. "Exploring The Use Of PLC Debugging Tools For Digital Forensic Investigations On SCADA Systems." The Journal of Digital Forensics, Security and Law: JDFSL 10.4 (2015): 79.

16 Basnight, Zachry, et al. "Analysis of programmable logic controller firmware for threat assessment and forensic investigation." Proceedings of the 8th international conference on information warfare and Security: ICIW. 2013.

James H. Jones, Jr.

Dr. Jim Jones is an Associate Professor in the Digital Forensics and Cyber Analysis program in the Electrical and Computer Engineering Department at George Mason University in Fairfax, Virginia. Dr. Jones earned his Bachelor's degree from Georgia Tech (Industrial and Systems Engineering, 1989), Master's degree from Clemson University (Mathematical Sciences, 1995), and PhD from George Mason University (Computational Sciences and Informatics, 2008). He has been a cyber security practitioner, researcher, and educator for over 20 years. His research interests are focused on digital artifact persistence, extraction, analysis, and manipulation.

Peggy S. Brouse

Dr. Peggy Brouse is an Associate Professor in the Systems Engineering and Operations Research department and the Director of the Cyber Security Engineering undergraduate program at George Mason University. Before Mason, Dr. Brouse worked as a Program Director at the MITRE Corporation, a database analyst at CACI, and systems analyst at the U.S. Army Computer Systems Command. Dr. Brouse is a member of the IEEE, ASEE and INCOSE. Within INCOSE, she served as the director of the Academic Forum. Degrees earned: BS in Computer Science, American University; MBA, Marymount University and Ph.D. INFT, George Mason University.