Designing and Developing Virtual and Real Testbeds for Industrial Control Systems Education and Training

A Thesis

Presented in Partial Fulfillment of the Requirements for the

Degree of Master of Science

with a

Major in Computer Science

in the

College of Graduate Studies

University of Idaho

by

Matthew J. Kirkland

Major Professor: Daniel Conte de Leon, Ph.D.

Committee Members: Michael Haney, Ph.D.; Yacine Chakhchoukh, Ph.D.

Department Administrator: Terence Soule, Ph.D.

December 2019

Authorization to Submit Thesis

This thesis of Matthew J. Kirkland, submitted for the degree of Master of Science with a Major in Computer Science and titled "Designing and Developing Virtual and Real Testbeds for Industrial Control Systems Education and Training," has been reviewed in final form. Permission, as indicated by the signatures and dates below is now granted to submit final copies for the College of Graduate Studies for approval.

Major Professor:	Daniel Conte de Leon, Ph.D.	Date
Committee Members:	Michael Haney, Ph.D.	Date
	Yacine Chakhchoukh, Ph.D.	Date
Department Chair:	Terence Soule, Ph.D.	Date

Abstract

Industrial Control Systems (ICS) are increasingly being targeted by cyber-attacks. The demand for highly qualified cybersecurity professionals, needed to secure these systems, is at an all-time high. Current ICS cybersecurity education and training on ICS is costly and not currently available at scale. This thesis attempts to solve this problem by providing accessible training testbeds for ICS cybersecurity education. This proposed solution is covered in the span of 3 major contributions.

(1) A thorough and structured literature review on the state of ICS testbeds was performed. (2) Based on those findings, a virtualized testbed, vWaterLab, meeting the common criteria of modern educational systems related to ICS, was created. (3) A detailed risk-informed design of a physically-enabled iteration of vWaterLab, called CacTiE, is provided. CacTiE, having built upon the lessons learned with vWaterLab, is more accessible and economic for smaller organizations. The construction of CacTiE is outlined and discussed.

Acknowledgments

Thank you to Dr. Conte de Leon and the members of my committee, Dr. Haney and Dr. Chakhchoukh, for their support and advice throughout my graduate education. Their input has greatly improved the quality of this thesis. I would also like to thank my fellow students and professors in the Computer Science program at University of Idaho.

I want to thank my family and friends which have been a crucial support, without which, this work would not have come to fruition. I am thankful for the wisdom of my uncle Cory which has guided me throughout my college career. Thank you to my mother who taught me the meaning of hard work and to never give up. Thank you to my best friend in the world, Josh Nelson, his strength of character has been guiding beacon for the past 10 years.

Thank you to the NSF CyberCorps®: Scholarship for Service Program for their mentoring and financial support. This material is based upon work supported by the National Science Foundation under Grant No. 1565572. I am thankful for the fellowship provided by my colleagues in the University of Idaho SFS program.

Dedication

I dedicate this thesis to my girlfriend Shannon Hurley, for her endless love and support.

Table of Contents

Authorization to Submit Thesis . . . ii
Abstract . . . iii
Acknowledgements . . . iv
Dedication . . . v
Table of Contents . . . vi
List of Tables . . . xi
List of Figures . . . xii
List of Acronyms . . . xiii
Chapter 1: Introduction . . . 1
The Problem . . . 1
The Proposed Solution . . . 1
Format of Thesis . . . 2
Chapter 2: Survey of Water Testbeds . . . 3
Chapter Introduction . . . 3
Problem Definition . . . 3
Current Solutions . . . 3
Outline of this Chapter . . . 4
Methods . . . 4
Syntactic Search . . . 4
Semantic Filter . . . 5
Supplemental Papers . . . 6
Current Water Testbeds . . . 6
The SWaT Testbed . . . 6
University of Louisville Hybrid System . . . 8
University of Florida Training Module . . . 9

Water Security Testbed (WSTB) 10
SecWater Framework 11
Research Applications 12
Threat Detection 12
Control System Hardening 15
Education & Training 18
Chapter Conclusion 19
Chapter 3: Education and Training with Virtualized Water Control Systems 21
Chapter Introduction 21
Problem 21
Proposed Solution, Objectives, and Contribution 21
Outline of this Chapter 22
Literature Review on Educational ICS Cybersecurity Testbeds . . 22
Needed Skills for ICS Cybersecurity 23
Instructional Methods 23
Bloom's Taxonomy 23
Hands-On 24
Lecture, Practical, and Debriefing 25
Project-based Learning 25
Blended Training 25
Educational ICS Testbed Characteristics 26
Control System Zones and Network 26
Programmable Logic Controllers (PLC) 27
Human Machine Interface (HMI) 28
ICS Protocols 28
Virtualization 28

ICS Security Training and Education . . . 29
SANS ICS Training . . . 29
ICS-CERT . . . 29
Certifications . . . 29
Educational Institutions . . . 30
Competitions . . . 30
Description of vWaterLab . . . 30
Overview . . . 30
Design . . . 31
Compute Infrastructure . . . 32
VM: pfSense Firewall . . . 33
VM: Engineering and Enterprise Windows 10 Workstations . . . 34
VM: Windows DC/LDAP Server . . . 34
VM: SCADA/HMI with ScadaBR . . . 34
VM: Ubuntu with OpenPLC . . . 35
VM: Ubuntu Server with Python: I/O modeling . . . 35
Tutorial I: PLC Pentesting using MODBUS . . . 36
Overview . . . 36
Educational Markers . . . 36
Tutorial II: Water Testbed Vulnerability Assessment . . . 37
Overview . . . 37
Educational Markers . . . 37
Chapter Conclusion . . . 37
Chapter 4: Risk Informed Design: Small-Scale, Educational Water Treatment Cyber Testbed . . . 39
Chapter Introduction . . . 39
Problem . . . 39

Current Solution 39
CacTiE: Cost-effective, Compact, Cybersecurity Testbed for ICS Education 40
Approach & Objectives 40
Contributions 41
Outline of this Chapter 41
CacTiE: Design 41
Lessons Learned from vWaterLab 41
Overview 43
CacTiE's Water Processing 43
Network Map 44
Architecture 44
CacTiE: Risk Assessment 47
Motivation 47
State of Risk in ICS Testbed Design 47
Risk Methodologies Reviewed 48
Regulations & Standards 50
Input Data 51
Proposed Method 51
HAZOP 52
HRA 54
Fault Tree Analysis 56
Results 58
Discussion 58
CacTiE: Risk Management 59
Identified Risks 59
Mitigation Goals 59

Assumptions 60
Resulting Recommendations 60
Societal Impacts 60
Discussion 61
CacTiE: Construction 61
Physical Assembly 61
CLICK Ethernet PLC and C-More Micro HMI 62
Wiring 63
Network Configuration 64
SCADA and Workstations 64
BoM 64
Literature Review 65
Procedure 65
Chapter Conclusion 66
Chapter 5: Summary and Conclusions 68
Bibliography 69

List of Tables

2.1 Syntactic keywords used in the relevance searching 4
2.2 Syntactic keyword search results 5

3.1 Skills and knowledge needed for ICS cybersecurity 24

4.1 Economic progammable logic controllers 42
4.2 List of guidewords [1] 53
4.3 HAZOP for CacTiE 55
4.4 HRA for CacTiE startup procedure 56
4.5 BoM for CacTiE 67

List of Figures

2.1 Semantic context of papers' abstracts 5

3.1 vWaterLab's design diagram 31
3.2 vWaterLab's network implementation diagram 32

4.1 Facsimilie process modeled by CacTiE 43
4.2 Network diagram of CacTiE 45
4.3 PLC, HMI, and SCADA/Workstation device implementations 46
4.4 The selected network appliances 46
4.5 The selected I/O devices 46
4.6 Fault tree for PLC failure 57
4.7 Fault tree for a pump failure 58
4.9 CacTiE phusical infrastructure 61
4.10 The programmed HMI 63

List of Acronyms

CoGS College of Graduate Studies

NSF National Science Foundation

SFS CyberCorps®: Scholarship for Service

VM Virtual Machine

PLC Programmable Logic Controller

SCADA Supervisory Control and Data Acquisition

ICS Industrial Control System

CPS Cyber Physical System

HMI Human Machine Interface

DLR Device Level Ring

CIP Common Industrial Protocol

EPA U.S. Environmentatl Protection Agency

MTU Master Terminal Unit

RTU Remote Terminal Unit

IP Internet Protocol

IO Input and Output

WSTB Water Security Testbed

PVC Polyvinyl Chloride

PEX Crosslinked Polyethylene

IDS Intrusion Detection System

BIDS Behaviorial Intrusion Detection System

PPM Physical Process Monitoring

SD State Dependent

SA State Agnostic

SSMP Single Stage Multi-Point

PE Pattern Engine

DP Data Preparator

DaD Distributed Attack Detection

ODM Orthogonal Defense Mechanism

MITM Man-in-the-middle

DDoS Distributed Denial of Service

DeC Design-centric

DaC Data-centric

ARM Association Rule Mining

SWaT Secure Water Treatment Testbed

WSTB Water Security Testbed

NIST National Institute of Standards and Technology

LAN Local Area Network

WAN Wide Area Network

CTF Capture the Flag

RADICL Reconfigurable Attack, Defend Instructional Computing Laboratory

DNS Domain Name Service

DHCP Dynamic Host Configuration Protocol

IPS Intrusion Prevention System

VLAN Virtual Local Area Network

VPN Virtual Private Network

AD Active Directory

GP Group Policy

LDAP Lightweight Directory Access Protocol

LD Ladder Diagram

IL Instruction List

FBD Function Block Diagram

SFC Sequential Function Chart

ST Structured Text

HiL Hardware-in-the-Loop

CDC Center for Disease Control and Prevention

CacTiE Cost-effective, compAct, Cybersecurity Testbed for ICS Education

AC Alternating Current

DC Direct Current

ENISA European Network and Information Security Agency

ISO International Organization for Standardization

NEMA National Electronics Manufacturers Association

HAZOP Hazard and Operability Study

FMEA Failure-modes and Effects Analysis

HRA Human Reliability Analysis

Chapter 1: Introduction

1.1 The Problem

Computers help make our lives easier by making processes and people more efficient. The natural implementation for many businesses is the adaptation of technology to increase efficiency and make businesses more profitable. The critical infrastructure sectors have followed in suit to deliver more reliable power, gas, water, and other services more efficiently. However, as technology has evolved, so have the threats to computing devices and the software that operates on them. As a result, cyber-attacks on technology have only increased. The threats to critical infrastructure have expanded and increased in frequency. Making the problem worse, there is a world-wide shortage of professionals that have the training to help thwart cyber-criminals let alone those trained in critical infrastructure security.

The automation and control technology behind Critical infrastructure is called Industrial Control Systems (ICS). These systems differ from modern information technology in a variety of ways. One way is that it is more robust because it must operate at a much higher level of up-time than traditional technology and it must do so in harsh conditions. This makes the technology expensive and harder to obtain. For this reason, national labs and large Universities have developed testbeds to perform control system cybersecurity research. However, this solution only allows a select few to use the testbeds, let alone for educational purposes.

1.2 The Proposed Solution

This research directly answers the problem by creating both a virtualized, vWaterLab, and a physically-enabled, CacTiE, educational testbed. These testbeds provide a variety of easily-available architecture solutions for ICS cybersecurity training by focusing on

low-cost, small-scale, simplicity, and Open-Source design. Further, educational material is developed for the vWaterLab testbed in the form of two ICS cybersecurity tutorials.

1.3 Format of Thesis

First, we discuss the current state of water-based testbeds and educational ICS testbeds in general. The water sector is selected due to the large area of need compared to the power grid research community. Second, a proof of concept is developed using the virtualization technology available at the University of Idaho's RADICL [2]. Third, a risk assessment is performed on the preliminary design for the physical testbed, CacTiE. The design is improved based on the results of a Hazards and Operability Analysis (HAZOP) [1], Human Reliability Analysis (HRA) [3], and Fault Tree analysis [3]. Fourth, CacTiE's risk influenced design is implemented and documented. Finally, a conclusion of the thesis and the above topics is performed.

Chapter 2: Survey of Water Testbeds

2.1 Chapter Introduction

2.1.1 Problem Definition

In the world of critical infrastructure, one of the prominent concerns is cyber-attacks the likes of which have been seen in Ukraine in 2015 and 2017. Attacks on critical infrastructure are a grave concern given the challenging nature of the defense of control systems. Much of the challenge lies in the fact that control system technology was designed and implemented in a time where cybersecurity was not a consideration. This tends to limit the defense of control system technology to "bolt-on" solutions and struggles to address security flaws that are rooted in their design. Further, this technology controls many essential processes that must have zero downtime. Meaning that patches or other enterprise security solutions are infeasible in application to industrial control systems (ICS). Additionally, the technology is expensive due to the high resiliency needs of control system technology compared to that seen in enterprise environments. These factors contribute to the complex nature of control system security.

2.1.2 Current Solutions

One of the more promising solutions is the usage of testbeds to develop secure technology and encourage research in cybersecurity for control systems. Testbeds allow for a myriad of experimentation that is not feasible on real infrastructure due to the following constraints:

- 100% availability is necessary in critical processes.
- Interference and damage may cause cascading failures that permeate to other infrastructure.

- It is dangerous to reveal vulnerabilities in real critical infrastructure.

As a result, the majority of research in this area is limited to simulations, testbeds, and other forms of abstraction.

2.1.3 Outline of this Chapter

This chapter discusses the current state-of-the-art in water sector cybersecurity. Section 2.2 will discuss the process of selecting and filtering the papers used in this survey. Section 2.3 will summarize current water sector testbeds and frameworks used. Section 2.4 will discuss the range of cybersecurity research being performed in the water sector. This research is broken down into 3 categories: threat detection, control system hardening, and education and training. Section 2.5 gives an overview of the survey itself and concluding thoughts.

2.2 Methods

2.2.1 Syntactic Search

The papers represented in this survey were collected using the following search engines: ACM Digital Library, IEEE Xplore, Google Scholar, Elsevier, and Springer. The keywords used for this purpose are listed in table 2.1.

Set	*Keyword(s)*
A	"Cyber-Physical Systems" "Security Testbed"
B	"water distribution and treatment" "cyber securit" "SCADA systems"
C	"ICS Security" "water treatment"

Table 2.1: Syntactic keywords used in the relevance searching

Each of the above keyword sets were searched on the aforementioned search engines using relevance searching. Table 2.2 outlines the number of results generated from the searches. If a search engine returned more than 15 results, the first 15 results were used and the others discarded. This process of elimination was used to grab the most relevant

Set	*Google*	*IEEE*	*ACM*	*Elsevier*	*Springer*
A	184	6	11425	7	2
B	8	134	951	1	1
C	273	2	36	9	4

Table 2.2: Syntactic keyword search results

papers in regard to the individual searches. This resulted in 130 search results. Finally, we removed the duplicates and non-paper results leaving 101 papers for further consideration.

2.2.2 Semantic Filter

The papers then had their abstract sections analyzed to determine the semantic context of the individual papers. The papers were then categorized as shown below in figure 2.1 The final filter is applied by the combination of the following 2 groups: "Cybersecurity

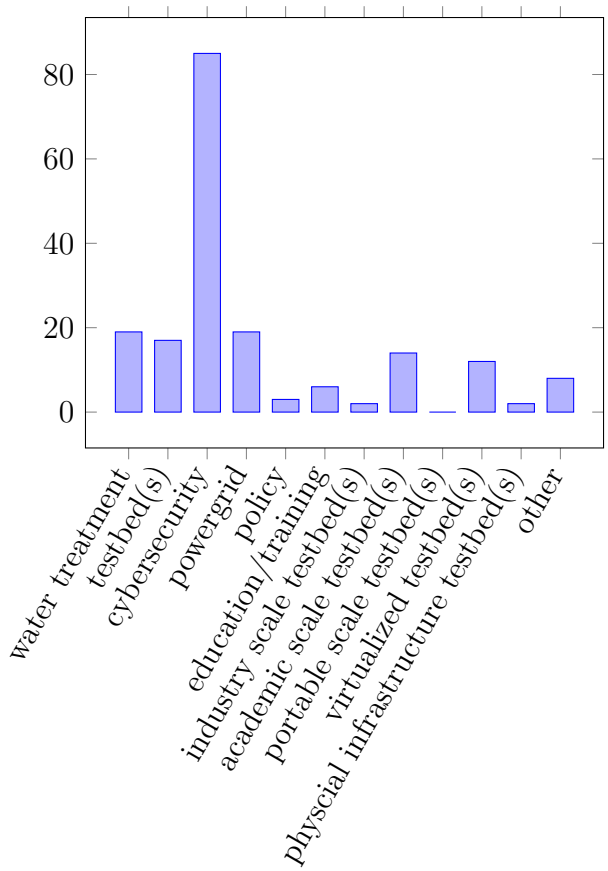


Figure 2.1: Semantic context of papers' abstracts

AND Water Treatment" OR "Testbeds AND Water Treatment". This resulted in 11 papers used as the basis for our detailed review.

2.2.3 Supplemental Papers

In addition to the 11 papers gathered in the process mentioned above. Dr. Haney made the author aware that the Environmental Protection Agency (EPA) had a water security testbed at Idaho National Lab. As a result, this survey includes a briefing document discussing the EPA testbed. This researcher could find no such published works on the EPA's testbed and this is likely the reason it was not uncovered in the efforts outlined in previous sections.

2.3 Current Water Testbeds

2.3.1 The SWaT Testbed

The Secure Water Treatment testbed (SWaT) is a testbed used to run an array of cybersecurity research at the University of Singapore. Mathur *et al.* [4] created this testbed to model a 6-stage water processing facility using both virtualized and real components. The main objective is to provide the important testing environments that will allow for the increased resiliency of ICS. [4]

The testbed produces 5 US gallons/hour and is meant to closely model a real water treatment plant in a form-factor of 90 square meters. Each of the 6 stages of water treatment are monitored and controlled by 2 programmable logic controllers (PLC) with one serving as a backup. All actuators can be controlled from the human machine interface (HMI) or supervisory control and data acquisition (SCADA) system itself. Each field device communicates with it's PLC in an Allen-Bradley Device Level Ring (DLR) protocol over Ethernet. The PLCs communicate over a separate star topology ethernet network. Each of the processes, the HMI, a historian and the SCADA connect to a network switch. All the field devices and the PLCs use EtherNet/IP (ENIP) and Common Industrial

Protocol (CIP) for communication. This communication may occur over wired or wireless connection. [4]

The 6 stages of SWaT are as follows:

- **P1**: controls the inlet pipe into the raw water tank and pumps water to P2
- **P2**: performs chemical dosing of P1 water
- **P3**: ultra filtration occurs and water is pumped to P4
- **P4**: water is stored in a tank and then pumped to Ultraviolet Dechlorinator (sodium bisulphate NaHSO3) may be added
- **P5**: de-chlorinated water is passed through a reverse osmosis filter. The reject is stored in a UF backwash tank.
- **P6**: controls the ultra filtration backwash cycles (keeps the UF filter clean)

Mathur *et al.* 's [4] initial research using SWaT involved the testing of 3 different attacker profiles: An attacker with

1. Access to the SCADA layer network
2. Nearby but NOT physical access
3. Physical access

Several simulated cyber-attacks were performed using the above attacker models on the SWaT testbed. These attacks focused mainly on reconnaissance and man-in-the-middle compromises. These scenarios gave some insight as to how to secure SWaT for the future and aided in the characterization of cyber-physical systems (CPS) response to attacks. Future works for SWaT include adding more diverse sensors to each of the stages in SWaT, hardening SWaT to handle significantly more impure raw water, adding online testbed access, and the connection of SWaT to another testbeds to observe the cascading effects between sectors or other industrial processes. [4]

2.3.2 University of Louisville Hybrid System

A masters student, by the name of Kyle Moss, at the University of Louisville created a small water-processing testbed. This testbed was used mainly to test the impacts of a hardened remote terminal unit (RTU) for use in water purification ICS environments in Moss's Masters thesis [5].

The testbed itself is partially physical and virtual. This testbed consists of a simulated HMI and Master Terminal Unit (MTU) for water processing and distribution. The testbed also includes a "security hardened RTU" physical component that lies between these other systems. Testing this component is the true purpose of this testing environment. In addition, it should be noted that this system design is one-to-one, since it has only one RTU and one MTU. [5]

Moss's design did not incorporate the use of a real-world communication protocol(s) between subsystems or components. Instead, for simplicity's sake, a stripped down UDP protocol was implemented. This communication scheme included 6 different operations: read, write, operate, challenge, challenge-response, and read response. [5]

The testbed simulated water treatment and distribution. The distribution system was comprised of a handful of pump stations. Each station has a holding tank that is filled with a nearby pump. The system is then gravity fed for distribution to customers. Before that occurs, the water is processed. The purification process was described as simply being a series of linked reservoirs. While the water is between reservoirs, chlorine among other chemicals, are added. The reservoirs work to filter out other foreign material as it settles to the bottom of the holding tanks. No specific numbers were given on the number of tanks or pumps used in the simulation [5].

For the implementation, the HMI/MTU was created in C#. This interface is used to set the IP and port of the physical system (RTU) that will be communicating with the simulation. For the sake of realism, the polling data of the pumps and tank levels

can be modified during the simulation. The secure design being tested in the apparatus requires that each node has its own user ID and key. These are used for validation of confidentiality and integrity. The HMI further gives break downs of the chemical dosages, flow percentages on pipes, and set points on equipment. Further, the different systems communicate using a National Instruments NI USB-6009 because of it's number of I/O ports and its ease of use with virtualized systems. The virtualized water treatment and distribution portions were created with a LabVIEW 2010 [5].

In the SCADA cybersecurity testing conducted using Moss's testbed, 4 attack scenarios were carried out with a malicious actor creating malicious MODBUS packets:

1. Write coil attack (w/o security device)
2. Read request attack (w/o security device)
3. Write coil attack (w/ security device)
4. Read request attack (w/ security device)

The results showed that the security device was able to authenticate the appropriate MODBUS communications and block the unauthenticated attack communications. Moss describes future work for the testbed including adding of an Allen Bradley PLC to allow more automation to take place in the testbed, thereby increasing the realism of the apparatus. [5]

2.3.3 University of Florida Training Module

The only water ICS testbed aimed solely at education and training this author has found was created by Doering *et al.* [6] at the University of Florida in 1985. This testbed focuses on the training of water treatment operators. Operator trainees are normally required to undergo certification that requires working experience to become a licensed plant operator. This is a very time consuming process, which could potentially be shortened using an educational testbed.

Doering *et al.* 's [6] testbed design involves 4 components: the controller board, 12bit TECMAR analog to digital and digital to analog converters, a Radioshack TRS-80 Model computer, and a cassette recorder. The controller board would act as the interface between the testbed and the student and instructor. The controller board has a live-updating flow diagram for the student and instructor to be able to view the current state of the testbed. Additionally, a drawing of the water treatment facility being modeled would be printed next to the control board. Control valves would be modifiable with potentiometers for the student to adjust actuation with the process. Section 2.4 discusses research and future applications of Doering *et al.* 's [6] testbed.

2.3.4 Water Security Testbed (WSTB)

The EPA is the lead federal government body charged with protecting the quality of the water supply and it's distribution in the United States. In direct response to the Homeland Security Presidential Directive 9 and Presidential Policy Directive 21, the EPA has constructed a testbed, known as the WSTB, to aid in the protection of water from intentional and unintentional contamination. WSTB allows for research in decontamination methodologies that are best to be used by municipalities. [7]

WSTB comprises of 450 feet of "mortar lined, ductile iron pipes" with two fire hydrants at each end. Each end of the L-shaped piping contains sensors and injection points used for introduction and detection of water contaminates. Additionally, the testbed was constructed above ground for ease of access, leak detection, and prevention of groundwater contamination. The piping is connected to a small premise plumbing that models residential water appliances (water heater, sink, dishwasher, washing machine, and refrigerator). These appliances are connected with a mix of 6-inch diameter PVC, copper, and PEX plumbing to represent the common forms of plumbing found in residence. About 200 feet of 1-inch copper pipe connects the premise plumbing to the testbed. This can be used to detect contamination at a simulated residence. The facility also contains a

28,000-gallon lined lagoon for managing contaminated water used in testing. [7]

The EPA's testbed allows for a variety of research: complex configuration cybersecurity experimentation, distribution network modeling, water quality detection instruments and mitigations, first responder training, and more. The EPA outlines future work for this testbed as finishing biological decontamination, evaluating decontamination of chemicals and radioactivity, evaluation and commercialization of water treatment units, and evaluating cyber-attacks on instrumentation and automation technology. [7]

2.3.5 SecWater Framework

While not actually a testbed, SecWater is a security framework for use in designing secure water treatment facilities developed by Aditya Mathur. It is generic in that it is applicable to other industrial processes, but is targeted with water treatment in mind [8]. There are 7 layers in Mathur's SecWater framework. It is recommended each layer should be implemented as much as cost allows. [8]

The first layer (SL0) is the first line of defense: prevention. It is essentially the implementation of firewalls in the ICS network. SL1 focuses on prevention of attacks and performing some, potentially automated tasks, based on alerts from an Intrusion Detection System(IDS). SL2 similarly, is about detection, but it assumes the attacker has penetrated the ICS network at a much more sophisticated level. SL2 implements what is known as Physical Process Monitoring (PPM). PPM performs deep packet analysis and checks against a rule-set (defined by physical and chemical characteristics) to determine if an alert should be generated. [8]

Multi-point attacks on field devices, however, can prevent the detection schemes outlined in SL2. This is addressed in SL3, where process-specific invariants are placed inside the PLCs, HMI, and/or SCADA. An invariant is "a mathematical relationship among "physical" and/or "chemical" properties of the process controlled by the PLCs in a CPS". [9] These invariants, combined with additional statistical methods offer this new schema

that is capable of detecting the compromise of multiple PLCs. [8]

A further deeply rooted compromise of a water CPS can be detected in SL4 by implementation of Orthogonal Defense dynamics. Orthogonal defense is the implementation of an independent monitoring system. SL5 assumes compromise down to the field device layer. It offers defense in the form of command validation to each actuator. The purpose is to prevent commands that would harm the actuator or other parts of the industrial process. Finally, SL6 is the post-detection control mechanism; how to protect the most critical parts of the process. Mathur describes the SecWater framework as essentially the implementation of the NIST framework for water treatment systems. [8]

2.4 Research Applications

2.4.1 Threat Detection

Process-Related Threats

SCADA systems face a variety of cybersecurity threats. These threats take many forms, including the more traditional attack where system vulnerabilities are leveraged to perform a malicious action. However, an attacker may also use a valid SCADA application to perform a malicious action using permitted actions. This attack scenario is referred to as a process-related threat and cannot be mitigated or detected by traditional cybersecurity mechanisms. [10]

However, Hadžiosmanovic *et al.* propose that the viewing of SCADA and system logs may give evidence of process-related threats. The initial problem is to find a way to analyze and identify process related threats using semantics. This is difficult given the large number of logs generated in a given period of time. [10]

Hadžiosmanovic *et al.* propose an analysis-tool capable of detecting process-related threats. The tool is called "MELISSA" and it has 2 essential components: the data preparator (DP) and the pattern engine (PE). The DP purpose is simply to aggregate

data into a useful format for the PE. [10]

The PE then proceeds to use an algorithm to mine frequent patterns and outputs the patterns in a list based on their frequency. Discussions with shareholders revealed that it may be best to utilize maximal pattern matching. Essentially, the algorithm will only output patterns that match the maximum number of attributes. [10]

Hadžiosmanovic *et al.* tested MELISSA against 14 days worth of SCADA logs from a real water treatment plant. It was capable of finding 486 unique patterns, it averaged about 12-79 patterns daily. The stakeholders claim that manual inspection would be feasible for up to 50 logs a day. [10]

Distributed Attack Detection (DaD)

In an ICS, there are multiple stages to a process. Each stage is typically controlled by a PLC and the PLC then controls the field devices responsible for that phase of the process. This research takes a look at attacks that target a phase of the industrial process by leveraging an attack on multiple field devices or controllers in that phase. This type of attack is referred to as Single-Stage Multiple Point (SSMP). [9]

Adepu *et al.* state that the detection of SSMP relies on the usage of invariants. An invariant is a relationship between chemical and physical properties of the industrial process in question. These invariants can be used to check for SSMP attacks. Their research utilizes 2 types of invariants: State Dependent (SD) and State Agnostic (AD). SD invariants consider the current state of the process and AD invariants do not. [9]

The SWaT testbed was used to test the detection of SSMP attacks using both SD and AD invariants programmed into each PLC's logic. This configuration on the SWaT testbed was tested against 7 different attack scenarios where the following were compromised:

- Input sensors
- Input and output sensors
- Input actuators

- Input and output actuators
- Full stage compromise
- Input and output sensor and input actuators

The results showed that when both SD and AD invariants were utilized all of the above attacks could be successfully detected. It is also important to note that this research only utilized manually crafted, physics based invariants. [9]

Sequence Attacks

ICS are increasingly being attacked by more advanced adversaries. These attacks may have advanced knowledge of the control processes that they are targeting. This type of attack on ICS is known as a "semantic attack". Some semantic attacks can be detected with traditional IDS if they are single malicious events. However, a semantic attack can occur over a series of events that alone are benign but are malicious when combined. This special attack is known as a sequence attack [11].

Caselli *et al.* [11] discuss the development of a layered IDS capable of detecting these attacks called the S-IDS. The first layer consists of the reader. The reader is responsible for interpreting packets, logs, files, data streams, etc. to a single format. The next phase is the Sequencer. Now the reader output is organized based on order, event type, and timestamp. The third phase is the Modeler. This phase creates models representing overall system behavior. It works to compress and emphasize features that would otherwise be hidden. Finally, the detection layer comprises of the algorithms used to detect malicious deviations from other "good" models. These good models must be obtained in a prerequisite learning phase of the S-IDS. [11]

Caselli *et al.* tested the S-IDS on a real water treatment and purification facility's data. It was able to detect and identify sequence attacks while keeping the number of false positives low [11].

2.4.2 Control System Hardening

Orthogonal Defense Mechanism (ODM)

The increasing complexity of attacks against ICS requires developing more advanced defensive structures to defend them. Shrivastava *et al.* propose the development of an Orthogonal Defense Mechanism to achieve this goal. ODM is a collection of software and hardware that are external to the ICS and designed to defend a given ICS [12].

The proposed ODM, by Shrivastava *et al.* [12] , consisted of what is known as intelligence checkers (ICs). Intelligence checkers use process based invariants to detect behavior that is opposed to the health of the process. The ODM contained 2 types of ICs: local and global. The local IC is planted in each sub-processes logic. The field devices have their I/O duplicated to the local IC. The local IC communicates with a global IC, which checks on the health of the process, utilizing the data from each local IC. The ICs are implemented with Raspberry Pi 3's running Ubuntu 16.04 LTS. [12]

Shrivastava *et al.* [12] tested their ODM on the SWaT using a Python script they developed called "WTreat Assault". This script automated 8 of the 9 attacks used to test the ODM developed. These attacks consisted of Man-in-the-middle (MITM), overflow injection, spoofing, replay, and multi-point attacks. The final attack was a distributed denial-of-service (DDoS) against SWaT using hping with 6 Linux machines. The ODM was successful in detecting all of the mentioned attacks with zero false-positives. It also detected each attack nearly directly after they were launched. [12]

Integration of Design and Data Centric Invariants

Much of the work outlined in this paper has to do with using process variants to detect attacks. Azmi Umer *et al.* [13] discuss how the combined implementation of design-centric (DeC) and data-centric (DaC) invariants can improve the defenses of an operational water plant. DeC invariants are developed utilizing the design of the process or system. In contrast, DaC invariants are developed using data derived from a process and tend to be

more state-based. Azmir Umer *et al.* [13] use SWaT to test the generation of invariants between the two processes: DeC and DaC. The general process was as follows for DeC:

1. Invariants are manually developed for SWaT using control algorithms and the physical specification
2. Invariants are codified into structured text
3. Invariants are placed inside of each PLC in SWaT based on either a global or local invariant classification (53 invariants total)

The process used to generate invariants for DaC was as follows: [13]

1. State information from SWaT is gathered from network packets via the historian. SWaT is run for 24 hours for 7 days to generate this traffic.
2. Data set reduction based on features determined to be most relevant to invariant generation
3. Frequent data sets are developed from the reduced dataset
4. The frequency sets are given to an engine that develops the association rules. This is done using what is known as Association Rule Mining (ARM) in machine learning.
5. Plant and component specification are used to verify invariants generated. The invariants are then codified and placed inside the PLCs exactly as seen with DeC development.

Azmir Umer et al [13] conclude that the invariants generated in the DeC requires many man-hours from experts in the process to develop. In contrast to the more blind approach of DaC, it was able to develop invariants that may have been overlooked by experts. However, DaC did not generate invariants that would be considered fundamental

by process experts. Thus, the combination of approaches is likely the best to generate a holistic set of invariants to be used in distributed attack detection schemes. [13]

Model-Based Security Analysis

As malicious actors find increasingly more creative ways to penetrate CPS defenses, it becomes more difficult to defend infrastructure. Many of the methods involved in finding possible attack scenarios for a CPS are labor intensive and may overlook possibilities that should be considered. Kang *et al.* [14] suggest an approximate modeling method that can be used to find previously unknown attacks to a CPS.

Kang *et al.* [14] begin their process by building a model of the system in question. This research was conducted on the SWaT. The Alloy modeling language was used to build the model of SWaT based on components and behavior. The back-end of Alloy, the Alloy Analyzer, is then used to model the attacker based on their capabilities. The model of SWaT will include definitions of an unsafe state for the process. This will allow the model to determine if attacks are successful. This approach uses analysis techniques similar to what is used in Fault-tree analysis. The safety of varying stages is used to determine the safety of the overall process.

In conclusion, Kang *et al.* [14] found that of the 4 attack scenarios generated 3 were validated as possible attack scenarios. Of the 3, 2 of the attacks were unknown to the engineers prior to the start of this research. This method modeling to generate attack scenarios shows promise.

Cybersecurity Cost-Benefit Analysis

Critical infrastructure have been plagued with accidents resulting from equipment failure and natural disaster since their inception. Among incidents, malicious cyber-attacks are relatively new and a much lower frequency. Papa *et al.* [15] look at a cost-benefit analysis of defensive measures that work to detect and prevent both, more common accidents and cyber-attacks.

In their research, Papa *et al.* [15] look into the wastewater sector and the problem of

sewer overflows (SO) at over 10,000 gallons. Regardless of the cause, SOs can be detected using a computational approach of a Behavioral Intrusion Detection System (BIDS) and trust anchors. A trust anchor is a device that protects the integrity of sensor values, when connected upstream to the given sensor. A BIDS simply works by taking readings from the trust anchors and PLCs to determine if the behavior of the system is anomalous. The feedback is given to the operator to decide what actions or overrides should be taken. The effectiveness of this combined approach was confirmed using Matlab simulations. [15]

Papa *et al.* [8] estimate cost of overflows by multiplying the estimated number of overflows per year by the sum of the cost of EPA fine, cleanup, and property damage. Numbers from the California Water Board revealed that 96 SOs occurred in FY 2011 that exceeded 10,000 gallons. [6] Of these, it is estimated that 46 could be detected with the approach outlined. The estimation of the number of SOs annually came from these numbers averaged over an estimate of 1300 miles of sewer lines in populations of over 100,000. [6]The costs associated with each incident came from a variety of sources including the EPA and California Sanitation Risk Management Association. The final cost of defenses were pooled from the cost of training, installation of a minimum of 3 trust anchors and the BIDS, and hardware and software. [15]

Papa *et al.* [15] estimated that cities with over 100,000 in population can expect to average $200,000 in damages resulting from SOs over 10,000 gallons annually. This makes it very cost-effective for the implementation of the defenses outlined in their research. [8]

2.4.3 Education & Training

As mentioned previously, the only testbed to perform any work in the area of education and training, is the training module created by Doering *et al.* [6] at the University of Florida. This testbed was purely for training new wastewater treatment operators.

The training consisted of a trainee sitting in front of the module with their instructor. The instructor would then set a testbed input parameter to such that the effluent rate

would exceed operating limits. The trainee would then be tasked with regaining control of the process and regaining effluent parameters. [6]

Eventually, more complex training scenarios would be provided to the trainee. In addition to adjusting an input parameter, a simulated hardware malfunction would be included. These events would be timed and the trainee would have to diagnose and rectify the problem with the process. Other problems could be programmed in at a timed interval to add to the difficulty of the simulations. The training would continue in this fashion until the trainee was ready to test for their operator's license. [6]

Doering *et al.* [6] found that the training is effective and much less expensive than traditional "on-the-job" training for new operator trainees. The University of Florida testbed also shows that the reproduction of the biological and hydraulic processes of wastewater treatment can be effectively simulated in software.

2.5 Chapter Conclusion

This chapter has described the architecture and purpose of the University of Singapore's SWaT, the University of Louisville's hybrid testbed, the University of Florida's training module, the EPA's WSTB at Idaho National Lab, and Mathur's SecWater framework for securing water treatment facilities.

Additionally, current research in threat detection such as process-related threats, distributed attacks, and sequence attacks. These attacks are partially or completely mitigated with the integration of log-based detection schemes, state dependent and state agnostic invariants, and new smart intrusion detection systems.

Research in control system hardening of water treatment infrastructure was also discussed. The implementation of an orthogonal defense mechanism and a combination of design and data based invariants for detecting multi-point and distributed cyber-attacks. Control system hardening was also discussed in the form of discovering previously unknown CPS vulnerabilities using modeling and methods similar to what is seen in fault-

tolerance analysis. A cost-benefit analysis is performed on cybersecurity defenses that are capable of preventing more common sewer-overflows in addition to cybersecurity attacks.

In summary, testbeds are one of the most effective tools for developing defensive capability in the ICS realm. This chapter was intended to be a cursory glance at the current state of such technology and related research efforts in the critical water treatment sector.

Chapter 3: Education and Training with Virtualized Water Control Systems

3.1 Chapter Introduction

3.1.1 Problem

There is now a very well-documented need for training and education of cybersecurity professionals [16, 17, 18, 19, 20, 21, 22, 23, 24, 25]. This need is even greater for cybersecurity professionals working with Industrial Control Systems (ICS) cybersecurity. Furthermore, malicious threats to Industrial Control Systems (ICS) and critical infrastructures are growing rapidly [26, 27, 28, 29].

SANS and ICS-CERT offer ICS Security Training courses [30, 31]. These include hands-on experience and are taught by leading industry professionals. Colleges and universities are ramping up cybersecurity-focused course and degree offerings [32, 33, 34]. Despite these increased efforts, current training and educational opportunities struggle to reach the scale needed to train and educate our workforce in the area of ICS cybersecurity.

Effective, efficient, cost-effective, and widely available educational resources for ICS cybersecurity education and training are much needed. This is especially more so for the case of hands-on tutorials and open testbed designs.

3.1.2 Proposed Solution, Objectives, and Contribution

In this chapter, we introduce vWaterLab: a fully virtualized ICS cybersecurity testbed for education. vWaterLab objectives are as follows:

- Enable hands-on learning in ICS cybersecurity at scale,
- Cost-effective to replicate, deploy, and configure.

- Open, shareable, and freely available upon request.

The contributions described in this chapter are:

1. We describe vWaterLab's design and implementation (Section 3.7);
2. We introduce two tutorials that use vWaterLab (Sections 3.8 and 3.9);
3. We present a short literature review of ICS cybersecurity testbeds (Sections 3.2, 3.3, 3.4, and 3.5).

3.1.3 Outline of this Chapter

The rest of this chapter is organized as follows: In sections 3.2, 3.3, 3.4, and 3.5 we describe what we observed from of existing scholarly work in the realm of educational ICS cybersecurity testbeds. Section 3.6 briefly describes types of opportunities for ICS cybersecurity learning. Section 3.7 describes the design and implementation of vWaterLab. Sections 3.8 and 3.9 introduce two ICS cybersecurity tutorials for vWaterLab. Section 3.10 concludes our presentation. List of Abbreviations, Acknowledgments, and References follow.

3.2 Literature Review on Educational ICS Cybersecurity Testbeds

Using Google Scholar, we searched for and reviewed in detail ICS cybersecurity education literature from the last decade. For this search, performed in April 2019, we used the following search string: ["methods" "control system cyber security" "education" "training" "curriculum"].

The search above resulted in 17 items. Each of these 17 items was reviewed for semantic context. After eliminating items that were not related to education in ICS cybersecurity, we were left with 7 literature items. We were also made aware of 2 additional items that

are related to education in ICS cybersecurity. As a result, we categorize and report on our findings 9 literature items.

In the following three sections, we report about what we observed in these 9 articles with respect to: (1) Needed Skills (Section 3.3), (2) Instructional Methods (Section 3.4), and (3) Expected Testbed Characteristics (Section 3.5).

3.3 Needed Skills for ICS Cybersecurity

Of the papers reviewed, there was a considerable degree of overlap in ICS skills and knowledge deemed necessary. Almost all of the reviewed papers mentioned a introduction to security fundamentals and ICS implementation techniques. Morris *et al.* [35], Sitnikova *et al.* [36], and Foo *et al.* [37] make direct mention of introductory ICS cybersecurity training. This includes basic security concepts that are commonly taught in introductory cybersecurity courses. In addition, introduction to ICS and common protocols and equipment were included.

Plumley [38] proposed a framework for categorizing cyber training environments targeting ICS with a focus on incident response training [38]. Plumley presented a determination of the skills needed for each of the four NIST Incident Response Lifecycle phases. Except for a small overlap, skills listed by Plumley do not appear in other works. Excluding specializations as described in Plumley's thesis [38], skills discovered in our review are listed in Table 3.1.

3.4 Instructional Methods

3.4.1 Bloom's Taxonomy

Plumley's thesis [38] focused on the categorization of ICS training environments. The author provided a mapping of ICS cybersecurity education to Bloom's taxonomy. Plumley argued that Bloom's phases of `Remember` and `Understand` may be accomplished in a typi-

Confidentiality, integrity, availability, authentication, authorization, accountability, basic cryptography, certificates, hashes, and digital signatures.
Review of relevant events: Threats, vulnerabilities, and attacks.
Network traffic analysis and monitoring tools.
Firewalls, virus protection, and intrusion detection (IDS).
Network defense and segmentation.
Familiarization with ICS components: SCADA, PLC, HMI.
PLC Programming
Review of base protocols: DNP3 and MODBUS.
Injection attacks and countermeasures (false measurements).
Review of relevant standards and policies.
Red Team vs. Blue Team exercises.
Exposure to various vendors and equipment.
ICS-specific vulnerability analysis.
Mitigation design and implementation.

Table 3.1: Skills and knowledge needed for ICS cybersecurity

cal lecture-oriented teaching environment. The phases of `Apply`, `Analyze`, and `Evaluate`, may be better accomplished using educational ICS testbeds. The last phase, `Create`, may be accomplished with testbeds at smaller scale than seen in industry [38]. Yardley *et al.* [39] also describe Bloom's taxonomy as one of the base pedagogical pillars in their ICS educational plan.

3.4.2 Hands-On

There is a very large focus, in the reviewed texts, on hands-on exercises. Either the focus is implied or explicitly stated in many of the reviewed works [35, 38, 37, 39, 36]. Morris *et al.* and Sitnikova *et al.* suggest that Red-Blue Team exercises are a great way to implement the hands-on approach [35, 36]. Red-Blue team exercises involve two teams that compete against each other. The Red team attempts to subvert computing systems that the Blue team is actively defending. This allows both teams to practice using realistic attack-defense scenarios.

3.4.3 Lecture, Practical, and Debriefing

Foo *et al.* and Sitnikova *et al.* propose a three-pronged strategy for developing hands-on learning exercises [37], [36]. Their strategy is as follows:

1. Lecture on background;
2. Hands-on exercises;
3. Debriefing.

The first two items of this three-pronged strategy are a common format used by educational implementations discussed thus far. However, the third item is an addition proposed by Foo *et al.* [37]. Foo *et al.* state that the debriefing session, "... allows students to share their insights with each other." [37]. Foo *et al.* suggest providing a break between hands-on exercises and the debrief session, to allow students to think about their experience [37].

3.4.4 Project-based Learning

Yardley *et al.* implemented a Project-based learning approach for their 2013 summer school, which was focused on training students in Smart Grid cybersecurity. Project-based learning was described by Blumenfeld *et al.* [40]. Yardley *et al.* describe this approach with "... fostering student engagement and longer lasting learning are achieved by combining student interest with a variety of challenging, authentic and real-world problem-solving tasks [39]".

3.4.5 Blended Training

Harris proposes a blended approach that they have successfully implemented at I-daho National Laboratory's Homeland Security Division [41]. The blended approach is called the ADDIE model and involves 5 phases: Analysis, Design, Development, Imple-

mentation, and Evaluation. The first 4 sections are sequential in nature. However, the evaluation phase can occur many times at different parts of the ADDIE cycle.

The analysis phase focuses on the development of knowledge. The design phase involves giving direction to this knowledge in the form of objectives and skills. The development phase focuses on creating materials that will be used in an educational format. This is followed by the implementation phase where resources are allocated for the instructional program. The Evaluation phase is a feedback loop for the instructors to improve their instructional program.

3.5 Educational ICS Testbed Characteristics

In this section, we describe common and unique elements of the testbeds we reviewed. Many of the testbeds share common general designs but their implementations are varied and unique.

3.5.1 Control System Zones and Network

Usually, an ICS network is broken down into zones. All reviewed works mentioned at least an ICS zone and a Corporate zone. Generally, the ICS zone contains all of the control technology and the I/O devices. This is also called the Operational Technology network. The Corporate network is a traditional IT network that has some connection points into the OT network. All the ICS infrastructure reviewed had this configuration or a similar variant [42, 35, 37, 39, 39, 43, 44].

Gao *et al.* described an ICS testbed with detail [43]. The network configuration follows the ANSI/ISA-99 reference model. This model divides the ICS network into four levels:

- Level 3: Corporate Network
- Level 2: Supervisory Control LAN
- Level 1: Control Network

- Level 0: I/O Network

The major difference is that the ICS zone is subdivided into (Levels 2-0) smaller levels. The "Supervisory Control LAN" contains the high-level automation devices, servers, and HMI (Human Machine Interface). The "Control Network" contains the field level control devices. Examples of such control devices are: PLCs (Programmable Logic Controllers), RTUs (Real Time Units), and Smart Relays. The field devices (actuators and sensors) are located in the "I/O Network". Gao *et al.* implement this network design using a virtualized emulation based network for level 3 and level 2. A layer of physical devices represent level 1 and Matlab and Simulink is simulating the level 0 devices.

Green *et al.* offer another variation on their Lancaster's Testbed [44]. Lancaster's Testbed is designed specifically for security research. This testbed follows the Purdue Enterprise Architecture (PERA) [45]. Starting from the top of the stack, the "Enterprise Zone" is comprised of the "Enterprise Network" and the "Site Business Planning and Logistics Network". A "Demilitarized Zone" separates the "Manufacturing Zone" from the "Enterprise Zone". The "Manufacturing Zone" comprises of the "Site Manufacturing Operations and Control" level and the "Cell/Area Zone". The "Cell/Area Zone" can be broken down into the "Area Supervisory Control", "Basic Control", and "Process" layers respectively. The final zone, "Safety Zone" contains the "Safety-Critical" controls and equipment.

The Mississippi State University SCADA Security Laboratory contained both a serially connected ICS network and an Ethernet ICS network [35]. However, as more ICS are designed for Ethernet or are being retrofitted with Ethernet capability, Ethernet seems to a more popular option. Most other testbeds implement Ethernet-based networks, rather than serial-based.

3.5.2 Programmable Logic Controllers (PLC)

In hybrid ICS testbeds, even if mostly virtualized, it is common to have real field au-

tomation layer devices. PLCs are commonly used to implement MTU and RTU. Observed units include: Control Microsystems, Inc. SCADAPack LP PLC, Allen Bradley Compact Logix L35E, Siemens S7-300, Siemens S7-1200, and a variety of trainer PLCs from DirectLOGIC, Allen Bradley, and Mistubishi. Mot commonly, PLCs were programmed in Ladder Logic or in some cases: ANSI C [35, 37, 43, 44].

3.5.3 Human Machine Interface (HMI)

As noted earlier, it is critical that students be able to monitor the changes in the ICS during exercises. The HMI is the way this is accomplished in a real ICS and makes it all the more a necessity in an educational ICS testbed. An HMI was implemented in all of the testbeds mentioned in this chapter. An HMI is definitely as necessary as a PLC, if not more, in an educational ICS testbed. The only physical HMI observed in the reviewed paper's testbeds was the GE/Fanuc iFix and Factory Talk View 5.0 used in the Mississippi State University Labs [35]. The other implementations were either not mentioned or they used a custom-made HMI [42, 37, 39, 43, 44]. For example, the Queensland University of Technology in Australia implemented their HMI in Labview [37].

3.5.4 ICS Protocols

A variety of ICS protocols exist and may be implemented in an educational ICS testbed. However, by far the most commonly used protocols currently appear to be MODBUS [46] and DNP3 (Distributed Network Protocol) [47]. Both DNP3 and MODBUS are used for communication between PLCs, HMI, SCADA, and other ICS equipment. Both DNP3 and MODBUS suffer from a lack of security in their design. This brings many challenges to secure networks that use these protocols. There are other process specific protocols such as GOOSE.

3.5.5 Virtualization

A popular option was the use of virtualization to add a low-fidelity, scalable network.

This also allows for an easier integration and participation of multiple students. This is excellent for architecture that is aimed at Red Team / Blue Team exercises or other tasks that require flexible configuration. Ibukun, Foo, Yardley, Gao, and Green *et al.* all included some implementation of the virtualization platform in their respective testbeds [42, 37, 39, 43, 44]. Most commonly, a the implementation of a single ESXi Server is used to provide a platform for students to connect with a laptop to the ICS Infrastructure. The vSphere suite is also used to easily segment the control system network as needed for hands-on activities [48].

3.6 ICS Security Training and Education

In this section, we provide a review of the current ICS Cybersecurity training opportunities.

3.6.1 SANS ICS Training

SANS [30] offers a variety of courses relating to cybersecurity. As of June 2019, SANS offers 6 courses in the field of ICS cybersecurity. The delivery mechanism for most SANS courses is live in-person teaching of about a week duration. These courses offer several hands-on exercises.

3.6.2 ICS-CERT

ICS-CERT is a U.S. government organization that offers online and in-house training [31]. Online-based courses are knowledge-focused and freely available on the web. In-house and hands-on training is offered in a few locations and limited to a small number of applicants per year.

3.6.3 Certifications

Global Information Assurance Certification (GIAC) is a cybersecurity entity that offers several ICS security certifications [49]. Some of these are: (1) Global Industrial Cyber

Security Professional (GICSP); (2) Global Response and Industrial Defense (GRID); and (3) Global Critical Infrastructure Protection (GCIP). Obtaining one of these certifications requires the passing of a written test.

3.6.4 Educational Institutions

Several educational institutions offer ICS-focused cybersecurity courses. On an average, the course delivery mechanism of these educational institutions is traditional enrollment. Also, most the courses are lecture-based rather than hands-on. Some educational institutions are starting to implement hands-on training in their ICS and cybersecurity-focused courses [33, 34, 42].

3.6.5 Competitions

The objective of ICS-focused cybersecurity competitions is to offer participants an opportunity to practice ICS cybersecurity skills. Noteworthy competitions include: (1) the U.S. Department of Energy's CyberForce competition [50]; (2) the IIC Productions' ICS/SCADA event [51]; and (3) NSHC Security's SCADA competition [52]; . SCADA stands for Supervisory Control and Data Acquisition. Other competitions are being developed in this area.

3.7 Description of vWaterLab

Based on the educational parameters discovered, we propose vWaterLab, a virtualized platform for education in ICS cybersecurity. vWaterLab's design will be a hybridization that will possess the ability to cover the core ICS security material determined in the literature review, Section 3.2. We describe vWaterLab and the assembly of the virtualized water testbed.

3.7.1 Overview

vWaterLab is a low cost educational platform that is capable of running a variety of ICS

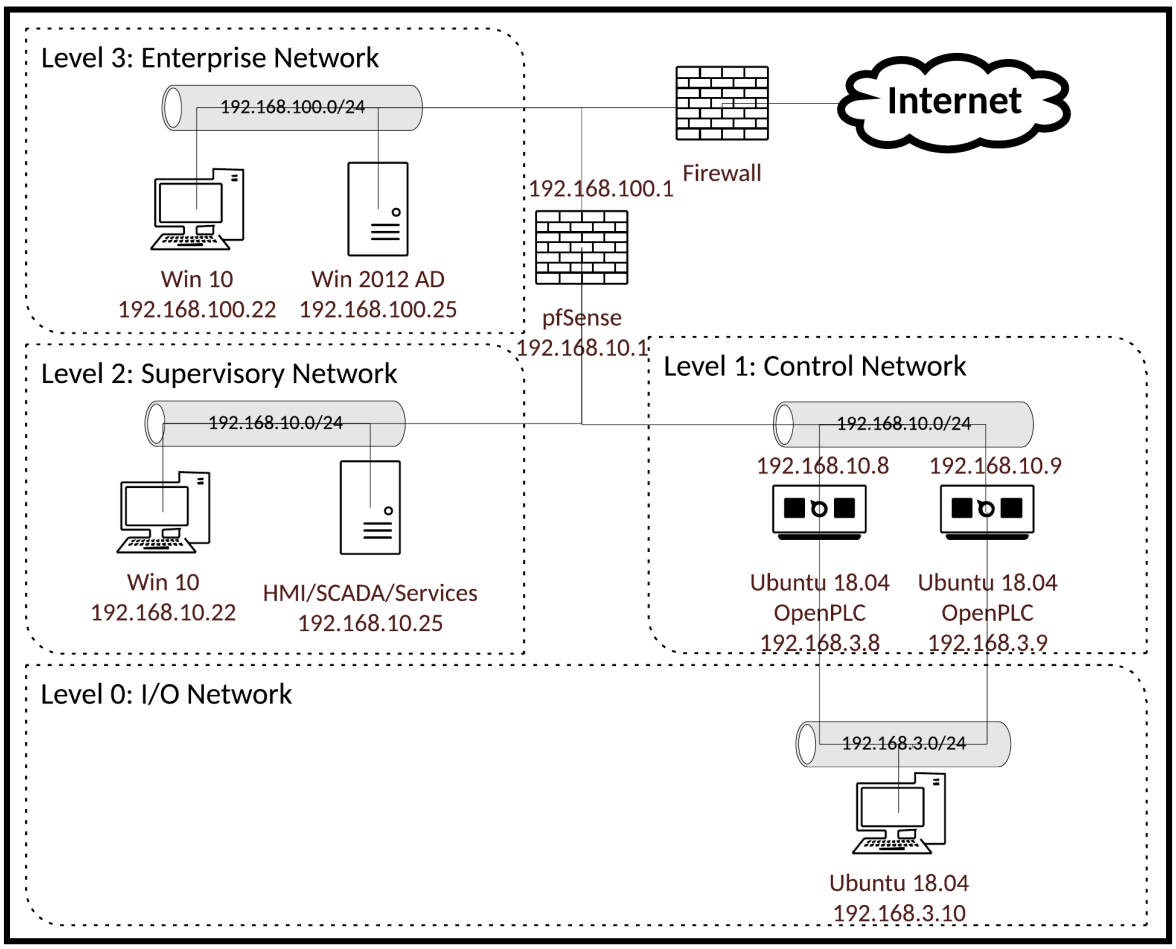


Figure 3.1: vWaterLab's design diagram

Security training scenarios. vWaterLab is based on a simplified water treatment process, enabling easier learning process. Furthermore, the testbed will be semi-configurable in the "Corporate" and "Supervisory" networks. Such customization will allow new components to be added or removed based on desired configuration for a variety of applications.

3.7.2 Design

vWaterLab design implementation can be seen in Figure 3.1 derived from previous works, as discussed in Sub-Section 3.5.1. The design is separated into 4 distinct zones based on the ANSI/ISA-99 reference model seen in Gao *et al.* [43]. Each of these network

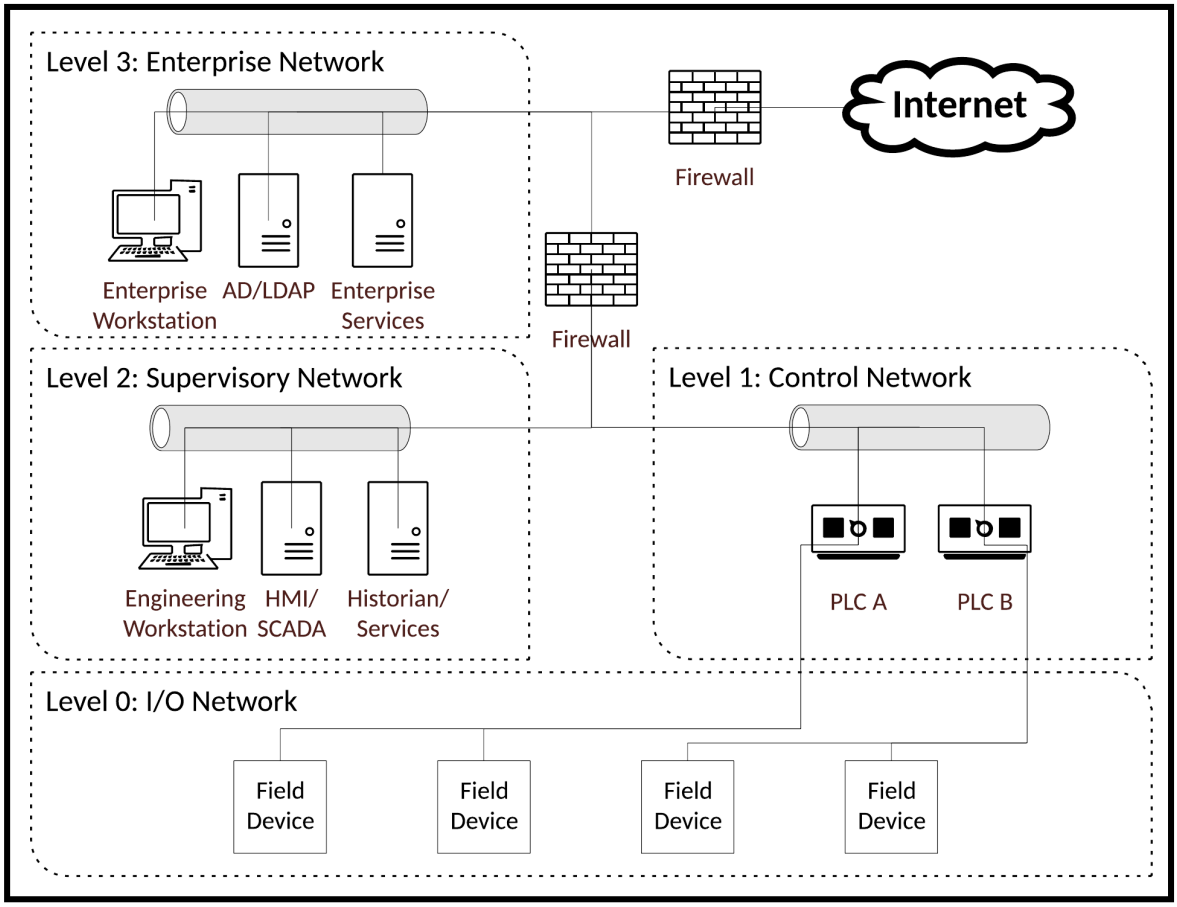


Figure 3.2: vWaterLab's network implementation diagram

connections are Ethernet. This is because all but Morris *et al.* used a completely Ethernet-based conduit architecture [35]. These design decisions and others are based on the literature review found in section 3.2 of this chapter. Figure 3.2 presents vWaterLab's network map, as currently deployed.

3.7.3 Compute Infrastructure

vWaterLab is a virtualized educational testbed, yet even so, it will need physical hardware to run on. This implementation will be carried out on the University of Idaho's own cyber-lab, RADICL. RADICL is the "Reconfigurable Attack-Defend Instructional Computing Laboratory". The goal of this special purpose laboratory is to enable hands-on

research and instruction in the areas of cybersecurity, cyber-defense, and modern computing platforms and networks. Since RADICL's inception, its hardware and software infrastructure has received several enhancements. The latest improvements, implemented in 2019, were funded by the State of Idaho and the M.J. Murdock Charitable Trust. RADICL makes full use of virtualization features built into modern cloud and enterprise computing cluster environments. RADICL enables teams of students and researchers to create and deploy multiple independent experiments that are quick to set-up and modify. Within the context of these isolated experiments, students and researchers design, implement, examine, explore, and develop a detail-oriented and hands-on view of modern computing infrastructures, along with their associated applications and protocols, and their strengths, weaknesses, and vulnerabilities. In addition, in RADICL, students and researchers develop a clear, detail-oriented, and hands-on understanding of the approaches, techniques, and tools used to protect today's computing systems and applications. RADICL also provides a dedicated and isolated platform that enables students to prepare and practice for cyber defense competitions.

3.7.4 VM: pfSense Firewall

pfSense was selected as the firewall implementation not because of it's wide spread use in ICS; pfSense was selected because it is very popular, Open Source option. It can easily be configured using a graphical web interface, making it an easier platform for beginners. pfSense also supports a number of options that would be very useful for future in-lab exercise: VPN Server, DNS/DHCP, IDS/IPS, Routing, Stateful Packet Inspection, IPsec, and VLAN support to name a few [53].

This dual network interface machine works as a router and firewall between the ICS and Enterprise Network. The firewall is configured to allow for any communication between the two networks. While not realistic, this configuration gives students an opportunity to practice configuring firewalls on the Enterprise and ICS network perimeter.

3.7.5 VM: Engineering and Enterprise Windows 10 Workstations

Currently, Windows 10 is an extremely popular choice in enterprise environments. Windows 10 was also selected for usage as the Engineering Workstation for simplicity's sake. Windows 10 is a platform that most students will be familiar.

The Engineering and Enterprise Workstations are both Windows 10 VMs. These machines are connected to the Windows AD Server's domain.

3.7.6 VM: Windows DC/LDAP Server

Windows Active Directory (AD) and Group Policy (GP) are an exceedingly common way to implement authentication of users and implementation of security policy in enterprise networks [54], [55]. The commonality of this setup is why it is implemented in the enterprise network of vWaterLab.

This Windows server VM has Active Directory and Domain Name services running on it. This is the only domain controller in vWaterLab. There are two AD accounts (Enterprise user & Engineer user) made on the network, one for each Workstation. It is used for authentication of the Windows workstations.

3.7.7 VM: SCADA/HMI with ScadaBR

ScadaBR is a free, open source Supervisory Control and Data Acquisiton (SCADA) software [56]. There are few SCADA software that exist; On top of being free, ScadaBR has a rich feature set. ScadaBR provides a Human Machine Interface (HMI), event and alert generation, and some Historian application for storing data at set intervals. ScadaBR is provided under GPLv3.

The Supervisory Control and Data Acquisition (SCADA) is implemented with an Open Source software called "ScadaBR". The VM can be directly downloaded from the OpenPLC Project's website https//www.openplcproject.com/reference-installing-scadabr [57]

. Additional configuration of this machine includes: adding both PLCs as data sources and HMI construction.

3.7.8 VM: Ubuntu with OpenPLC

Similar to ScadaBR, OpenPLC is open source and freely available [57]. OpenPLC features 3 pieces of software: the runtime, editor, and HMI builder. The runtime is the actual PLC software that can be run as a soft-PLC (Windows & Linux) or on an embedded platform (ex. Raspberry Pi). The runtime software can execute PLC programs in the form of Ladder Diagram (LD), Instruction List (IL), Function Block Diagram (FBD), Sequential Function Chart (SFC), and Structured Text (ST). The editor is software that allows for the creation of aforementioned PLC programming. Finally, the HMI editor allows for the PLC to display information and is implemented with the ScadaBR project. So compatibility, open source design, and the documentation of the code make this a perfect choice for PLC implemenation.

Obtained from the OpenPLC Project, the PLCs are implemented with the Open-Source OpenPLC Runtime and Editor [57]. These tools are installed on two Ubuntu 18.04 machines. The PLCs are not loaded with any ladder logic or other programs for vWaterLab. These machines are dual network interface. The reason being is that the input and output behavior is simulated on the I/O Simulation machine. One network interface connects to the I/O simulation and the other connects to the ICS network.

3.7.9 VM: Ubuntu Server with Python: I/O modeling

Hardware-in-the-Loop (HiL) usually refers to hardware and firmware-implemented mathematical models to represent physical processes accurately. HIL is used for a large quantity of the testbeds reviewed.

Since the objectives of vWaterLab are to enable easy and inexpensive replication, instead of expensive hardware/firmware HiL implementations, Python scripts with the PyModbus library are used to simulate the physical components of an ICS system and

their respective I/O.

This VM is implemented with the Ubuntu 18.04 operating system. It operates using a Python script (utilizing PyModbus) to communicate with the PLCs and update their memory with values that simulate physical I/O. Additional software installed includes: Python and PyModbus.

3.8 Tutorial I: PLC Pentesting using MODBUS

As a proof of concept, two introductory ICS Security tutorials were developed. These tutorials are two hours long and are a combination of lecture with a focus on hands-on application. These tutorials attempt to map to some of the key skill and knowledge areas identified in the literature review in Section 3.2.

3.8.1 Overview

This tutorial gives an introduction to ICS and PLCs. Students learn PLC programming, logic, and communication. Students focus on learning about MODBUS protocol and leverage its inherent security vulnerabilities to perform a variety of attacks. Methods to mitigate the attacks are also delivered in the tutorial.

3.8.2 Educational Markers

This tutorial maps to the following educational markers:

1. Familiarization with PLC and HMI components;
2. Introduction to PLC programming using Ladder Logic;
3. Examination of ICS traffic using Wireshark;
4. Review of MODBUS protocol;
5. Perform injection attacks.

3.9 Tutorial II: Water Testbed Vulnerability Assessment

3.9.1 Overview

This tutorial focuses on securing ICS using Vulnerability Assessment. An introduction to vulnerability assessment is given through a small, mock assessment. The students are introduced to the high-level methods of an assessment. The last part of the tutorial involves mitigating the injection attacks seen in the previous tutorial using defense-in-depth via firewall configuration. This tutorial is intended to serve as an extension of the `OpenPLC Pentesting` tutorial.

3.9.2 Educational Markers

This tutorial maps to the following educational markers:

- Familiarization with PLC, HMI, and SCADA components;
- Review of MODBUS protocol;
- Perform injection attacks;
- Perform an ICS Vulnerability Assessment;
- Develop mitigations to the injection (and similar attacks);
- Examination of ICS traffic using Wireshark;
- Exposure to network defense using Firewalls.

3.10 Chapter Conclusion

This chapter introduced vWaterLab: a new, freely accessible, virtual Industrial Control Systems-focused cybersecurity educational testbed. We presented the design and

implementation of vWaterLab. All the virtual components and network connections were described. In addition, two tutorials were presented that use vWaterLab. The tutorials and testbed itself follow methods observed in our literature review of educational ICS testbeds. We reviewed not only similar platforms, but also observed their architectural methods and configuration. vWaterLab's educational methods and metrics were also developed based on the works surveyed in our literature review. We hope that our endeavor, vWaterlab, will help provide an easily reproducible and cost-efficient educational resource.

Chapter 4: Risk Informed Design: Small-Scale, Educational Water Treatment Cyber Testbed

4.1 Chapter Introduction

4.1.1 Problem

For many, hands-on exercises are the best way to learn about a new topic or acquire a skill. In the field of cybersecurity, this is no different. Most cybersecurity education follows a similar path of lecture followed by application with practical skill building exercises. The explosion of events like: capture the flag, red team vs. blue team, and online tutorials through virtual machines has made it easier for aspiring cybersecurity students to obtain their much needed practice. However, the area of ICS Cybersecurity has not seen the same growth in hands-on training exercises. One reason for this is the lack of ICS environments available for students to practice their skills. Usually, ICS training environments (or testbeds) are expensive, large-scale, or too complex for a beginner to understand, let alone setup.

4.1.2 Current Solution

There are a number of virtualizated platforms in existence. However, they are not catered for educational purposes; They are complex systems that require advanced knowledge of the modeled processes. They are generally not made for laypersons to understand. On top of that, these virtualized platforms make use of physical servers that can be very costly. Additionally, virtualization of the I/O layer of the ICS often lacks realism. For this reason, it is generally recommended to use real-hardware at this layer. Lastly, they

are not open source designs. This is not a problem for their primary application, research; However, it makes them more difficult to be used in a wide-spread manner.

4.1.3 CacTiE: Cost-effective, Compact, Cybersecurity Testbed for ICS Education

This chapter proposes the development of a small-scale, physical, economical, educational teaching platform for ICS Security education called CacTiE. The hybrid design of CacTiE features real ICS components and physical I/O. This implementation should directly address the problems addressed in the previous section. The goal of CacTiE is to provide a cheap alternative for colleges, high schools, and organizations that have limited resources.

4.1.4 Approach & Objectives

There are multiple methods used to provide a more compact, economical, realistic, and accessible educational platform. First, there are a small number of small-scale, low-cost, real ICS automation components (i.e. PLCs). Examples of these can be seen in table 4.1. Second, cost savings can be realized by purchasing non-industrial grade equipment, where it is unnecessary for educational purposes: switch, router, workstations, and I/O devices (actuators/sensors).

The overarching design goals for CacTiE are as follows:

- Cost: $< \$2000$
- Size: Desk-sized
- Simplified process
- Real, Physical I/O
- Facsimile process
- Open Source design

4.1.5 Contributions

The contributions of this chapter are as follows: (1) The design and construction of a low-cost, small-scale, educational ICS Cybersecurity testbed. (2) An approach for assessing risk in ICS testbeds is proposed and utilized to generate a risk-informed design for the testbed. (3) We apply this approach to the design previously mentioned.

4.1.6 Outline of this Chapter

The rest of this chapter will introduce the proposed testbed design. The proceeding two sections are dedicated to the risk analysis and mitigation of the proposed testbed design. The testbed's construction will then be laid out for others to be able to model and create their own testbed. The literature review performed on current, educational ICS testbeds is outlined and discussed. Finally, the chapter will conclude with a summation of the chapter and future works.

4.2 CacTiE: Design

This section will review the reasons for building CacTiE. The overarching design decision, goals, and implementation are outlined. The design is further influenced by a variety of relevant standards and risk analysis in section 4.3 and section 4.4. The final design and it's implementation are revealed in section 4.5.

4.2.1 Lessons Learned from vWaterLab

Previously, this author built a virtualized platform for ICS cybersecurity education. vWaterLab is a completely virtualized, educational ICS Security testbed. As with any design, there were some flaws and lessons learned from building vWaterLab. Here are the issues encountered with the implementation of vWaterLab:

- Moderately-sized, virtualized ICS testbed environments require expensive hardware to operate (i.e. servers)

Vendor	*Product line*	*Start Cost (USD)*
Velocio.net [58]	Ace	$49.00
Allen Bradley [59]	Micro800	$49.50
AutomationDirect [60]	CLICK	$69.00
Velocio	Branch	$69.00
Divelbiss [61]	Solves-It!	$91.00
Divelbiss	Micro Bear	$99.00
AutomationDirect	DirectLogic	$127.00
Siemens [62]	LOGO	$145
AutomationDirect	Productivity	$171.00
Divelbiss	Enhanced Baby Bear	$178.00
Divelbiss	PCS	$193.00
AutomationDirect	Do-More	$197.00
Divelbiss	Harsh Environment (standard)	$340.00
Divelbiss	Versatile Base	$392.00
Schneider Electric [63]	Modicon	$449.00
Divelbiss	Harsh Environment (Enhanced)	$760.00
Allen Bradley	MicroLogix	$872.00
Siemens	S7	$1,355.00
Allen Bradley	SmartGuard 600	$1,805.00
Allen Bradley	SLC 500	$2,390.00
Schneider Electric	SCADAPack	$2,530.24
Allen Bradley	CompactLogix	$3,790.00

Table 4.1: Economic progammable logic controllers

- Beginner students cannot physically see the process (lack of the full-picture)
- Emulation and simulation may not accurately portray ICS components
- I/O layer simulation software is expensive and struggles to be as representative of the process
- Virtualization hardware and expertise in it's deployment are a barrier to wide-spread adoption

4.2.2 Overview

CacTiE is essentially a hardware implementation of vWaterLab with improvements based upon the flaws mentioned thus far. The goal of CacTiE is to provide a cheaper, hardware implementation by avoiding the cost and expertise that virtualization requires. The target audience is smaller, cost-restricted consumers (i.e. colleges, high schools, and independent prosumers). CacTiE leverages further cost and scale savings by using I/O and robust limited technology where it is not required for pedagogical application. Further, the processes that ICS testbeds model can be overly complicated for educational application. CacTiE avoids this problem by modeling a facsimile that is loosely based on a water treatment plant (See figure 4.1)

4.2.3 CacTiE's Water Processing

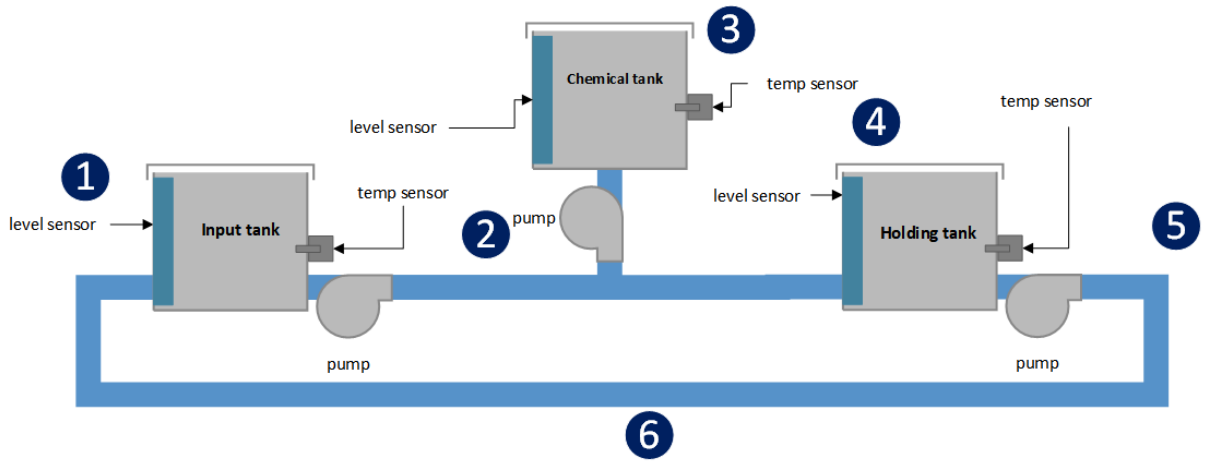


Figure 4.1: Facsimilie process modeled by CacTiE

CacTiE's water facsimile process is based on the disinfection stage in water processing. According to the Center for Disease Control and Prevention[64], the disinfection is the stage where a disinfectant (typically chlorine) is added to previously filtered water to remove parasites, bacteria, and viruses. CacTiE's process works by circulating water through three tanks filled with water. Water temperature sensors are used in place of

chlorine dioxide sensors for safety reasons. The process is cyclic and does NOT produce an effluent that is disinfected. The below section describes the flow of water in CacTiE (see figure 4.1):

1. Water flows into the 'Input tank'.
2. The pump drains the 'Input tank'.
3. Another pump moves the water to the 'Holding tank'.
4. As needed, 'chemical' is pumped into the 'Holding tank'.
5. The mixture is stored in the 'Holding tank' and aimed at holding a set concentration.
6. The 'disinfected' water is pumped out of the 'Holding tank'. In implementation this effluent makes its way back to the 'Input tank'.

4.2.4 Network Map

The network diagram is based on the previously mentioned vWaterLab testbed (see Figure 4.2). In this implementation, there is one PLC automating (1) the raw water tank levels (2) adjusting the amount of chemical to add to the mixture (3) The holding tank and the output pump.

4.2.5 Architecture

This section outlines the core physical components selected for CacTiE, how they were selected, and how they are implemented in the design.

Programmable Logic Controllers (PLC)

In order to simplify the design, only one type of PLC was used. The AutomationDirect CLICK PLC [60] was selected for 3 reasons: (1) It has one of the lowest price points (2) There is plethora of information on it's application and programming (3) It comes with free programming software. See figure 4.3a

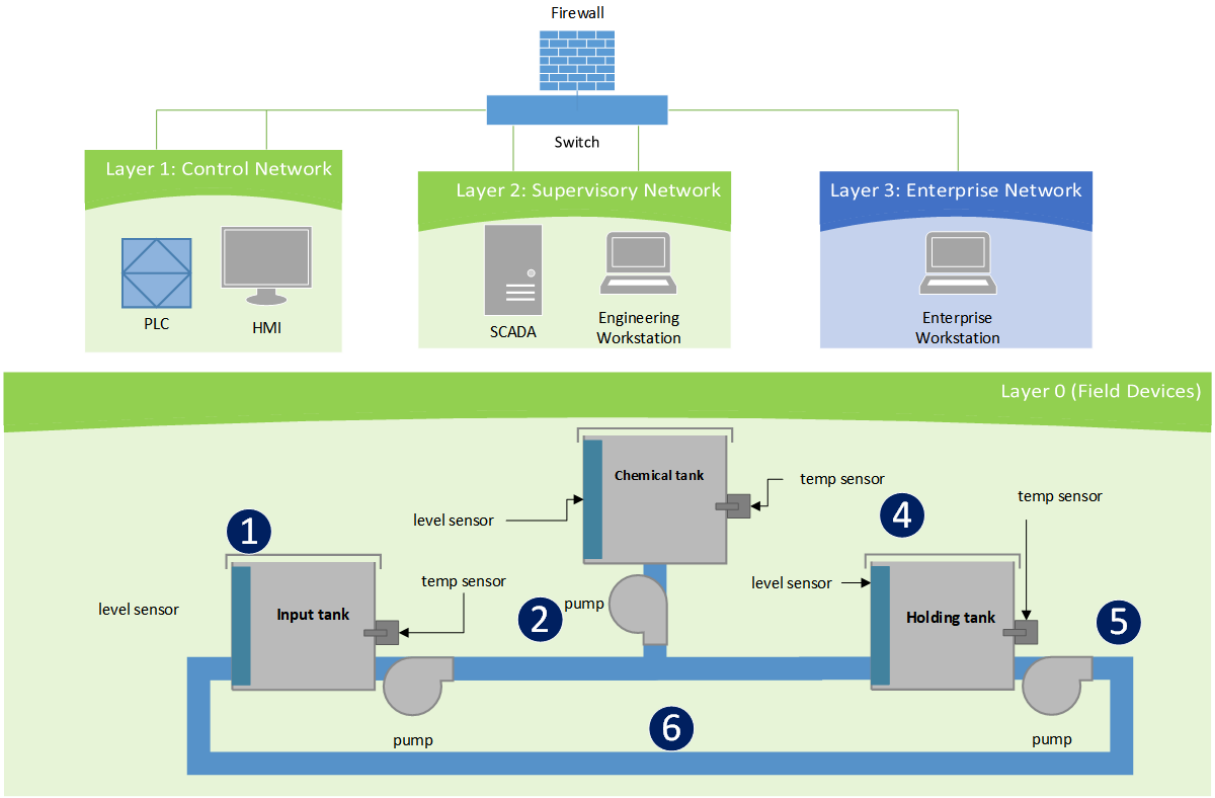


Figure 4.2: Network diagram of CacTiE

Human Machine Interface (HMI)

The HMI is used to report process information in the field and make adjustments on the fly. The choice of HMI was simple after selecting the CLICK PLC. AutomationDirect has a relatively cheap ($98) unit called the C-more Micro EA3[65]. The C-More is a small, 3 inch interface that works off of direct serial connection to the CLICK PLC. See figure 4.3b

Networking Appliances

Industrial application managed switches can be very expensive. AutomationDirect offers the relatively cheap STRIDE 5-port switch for $422. However, most managed switches tend to be even more expensive. In order to save on cost, a standard router and 2 unmanaged switches are purchased to implement the network management in CacTiE. The Linksys WRT AC1900 router [66] and TP-Link 5-Port switch meet this criteria at a

(a) CLICK PLC (b) C-More Micro HMI (c) Raspberry Pi

Figure 4.3: PLC, HMI, and SCADA/Workstation device implementations

(a) TP-Link 5-Port switch (b) Linksys WRT AC1900 router

Figure 4.4: The selected network appliances

combined $200.

Physical I/O (Sensors & Actuators)

A number of actuators and sensors are selected for CacTiE. Each of the three tanks in the system are equipped with 2 FLS-HS-100 [67] water level sensors and a single temperature sensor (see figure 4.5c). The water level sensors (see figure 4.5a) are implemented

(a) Float sensor (b) Water pump (c) Thermocouple

Figure 4.5: The selected I/O devices

with as float valves; The state of the float is used to determine if the tank is low or full. The type-k thermocouple (see figure 4.5c) is the robust THMK-A01L04-01[68] from AutomationDirect. These probes are armored and have a much shorter insertion length of 1/4" compared to 6", which is standard and too large for this application. The system includes 3 small, submersible, 5VDC pumps [69] (see figure 4.5b) capable of moving 150 liter of water per hour; These pumps are the only outputs and actuation utilized in the testbed.

Workstations and Servers

Expense and size restrictions, limited the choices for computing devices to be used for the the workstations and the SCADA server. The Raspberry Pi [70] was the choice for implementation for these reasons and in addition it is a well-documented and modular "system-on-chip" (see figure 4.3c).

4.3 CacTiE: Risk Assessment

4.3.1 Motivation

Any platform that is used in education should consider the inherent risk associated with the usage of the platform. Further, the physical nature of CacTiE, and similar ICS testbed platforms, makes the concern for safety and security obvious. For example, CacTiE will be wired with 120v AC power, making it a possibility for users or bystanders to be shocked. It is the duty of the designers and operators to ensure the system adequately considers risk.

4.3.2 State of Risk in ICS Testbed Design

There is a distinct lack of risk consideration in ICS testbeds. As seen in the literature review (see section 4.6), the majority of peer-reviewed papers discussing risk in ICS testbeds is astoundingly small. When it is mentioned, it is certainly not discussed at any

length.

Despite this, there are a number of standards that discuss risk assessments on full ICS environments. These methods and tools are considered in developing and conducting the risk assessment utilized on CacTiE. It is important to note that these standards have a large focus on cybersecurity-based, adversarial risk.

There is less mention of safety based risk, which we believe should be the main focus of CacTiE's risk assessment. This is due to the fact that the point of CacTiE is that Cybersecurity vulnerabilities should be present for students to learn to defend against. Risk, from malicious cyber-attacks, will only be considers as it has to do with intentional misuse and harm.

4.3.3 Risk Methodologies Reviewed

OCTAVE Allegro

A method of risk assessment developed by Carnegie Mellon for use in assessing organizational risk. OCTAVE Allegro [71] consists of 3 phases: (1) Assets are identified and current safeguarding strategies are reviewed. The criticality of assets is evaluated, security requirements are documented, and threats to those requirements are identified. (2) Key technical components and vulnerabilities are identified. (3) A list of relevant risks are developed. The most critical have mitigations developed. The target audience are individuals in charge of an organization's (of over 300 personnel) operational risk.

European Network and Information Security Agency (ENISA)

ENISA has a document[72] on risk assessment and risk management targeted at security and IT professionals. The document is meant to act as a general specification on how to perform risk assessment and management. The ENISA document bases much of it's structure on the OCTAVE and ISO 13335-2 standards. The document surveys a variety of other standards on risk assessment and management, tools, and their respective feature sets. The ENISA document further describes the generalized process of risk assessment

as: (1) Risk identification (2) Analysis of relevant risks (3) Risk evaluation. The continuation, or risk treatment, follows the steps of: (1) Identify all options (2) Develop an action plan (3) Action plan approval (4) Action plan implementation (5) Identify residual risk (6) Risk acceptance (optional).

BSI-Standard 100-3

This methodology, provided by BSI[73], provides a simple, risk analysis technique that may be applied to information processing. Once again, the target audience is security professionals. The first step is to perform a background investigation; The appropriate parties should perform preliminary analysis to ensure they understand the state of security and the system they plan to analyze. The second step involves summarizing the threats to the system in question. Relevant target objects are identified as well as threats to their confidentiality, integrity, and availability. Thirdly, additional analysis is conducted to determine if any threats were missed in the first sweep. Fourth, the threat assessment is performed. This step analyses current security measures in terms of completeness, mechanism strength, and reliability. The remaining risks (residual) are determined in step five. Step 6 looks at proposed security measure and determines their effect when combined with current measures. Step 7 involves evaluating the new implementation.

ISO 31000/31010

The ISO 31000/31010 method of risk analysis and mitigation, as described by Cross [74], is a methodology targeted for open system designs. Open Systems are complex systems that rely on external interactions. Typically, no one person understands the entirety of the process. The standard approaches risk assessment of these systems using a combination of traditional risk assessment and decision making processes. The overarching model breaks down risk management into scope definition, risk assessment, risk treatment; Risk assessment is broken down further into risk identification, risk analysis, and risk evaluation. Additionally, IS 31010 considers risk communication, recording & reporting, and monitoring & review. Relevant risk assessment tools mentioned include:

Failure Modes and Effects Analysis (FMEA), Fault Tree analysis, Hazard and Operability analysis, fish-bone method, decisions trees, etc.

4.3.4 Regulations & Standards

NEMA Standards Publication ICS 6-1993

This standard [75] has to do the practices that relate to the design and construction of ICS enclosures. This standard does not directly apply to this work, as this is meant mostly to inform users about the regulations and standards issued onto National Electrical Manufacturers Association (NEMA) classifications of enclosures. This is pertinent in that it informs the decision of what kind of enclosure protection is necessary based on moisture, sensitivity of equipment, ventilation requirements, component spacing, electrical grounding, and wire bending space.

NEMA Standards Publication ICS 1-2000

This standard [76] discusses general requirements for ICS (ex: Shock, temperature, vibration, signal noise, etc). Much of this document is more relevant to full-scale, real control systems. Key components in this document that are of relevance to CacTiE are: wire gauge recommendation, distance between components, Climatic Macro-Environmental considerations (outside of the enclosure), and micro-environmental considerations (inside the enclosure).

NEMA Standards Publication ICS 1.3-1986

This standard[77] details the preventative measures and maintenance that ICS require. ICS 1.3-1986 deals primarily with risk considerations on the human operations side of the testbed. This is mainly safety precautions like work practices, Deenergization, Work on Energized systems, and Control System equipment hazards. This standard also includes best practices for: cleaning, states of operation, servicing equipment, and equipment hazards.

4.3.5 Input Data

CacTiE's design is fairly simplified compared to a real ICS. In analyzing risk, we look at the major components of the testbed. These components are:

- PLC
- HMI
- Water Tanks (3)
- Water piping
- Monitor
- Switch and Router
- Raspberry Pi (Server and Workstations)
- The Pumps (3) and other sensors

4.3.6 Proposed Method

We propose the usage of a combination of traditional risk assessment methodologies: HAZOP, HRA, and Fault Trees. Many of the aforementioned methods currently used on real control systems are generalized approaches. We believe that the implementation of: HAZOP, HRA, and Fault Trees will provide sufficient risk analysis. Due to the simplicity of the design, HAZOP is a viable option and will provide a great deal of detail. HAZOP is traditionally used in chemical treatment processing and it will help in finding hazards before constructing the testbed. HRA was also selected because of the great degree of involvement of human interaction and decision making involved in running CacTiE. Finally, the most critical risk can be further analyzed with a fault tree to aid in determining root causes. This will lend itself directly to risk mitigations and management.

4.3.7 HAZOP

Overview

A HAZOP is a bottom-up, systematic method of identifying hazards in a system before they occur. Traditionally used in chemical processing. It is an iterative process of reviewing components and their parameters. In HAZOP, deviations are defined as parameters + guidewords. This is usually defined in a Deviations table.[1]

According to CandelaLearning [1], the process involved in performing a HAZOP is as follows:

1. Select component
2. Take note of the Design Intention
3. Select a process parameter
4. Combine with a guideword to form possible deviation to normal operation
5. For each deviation, identify possible causes and consequences
6. Evaluate existing protections for adequacy
7. Suggest suitable action to improve operability
8. Continue with other guidewords and repeat
9. Select new process parameter and repeat
10. Select next parameter and repeat

Guidewords

The process of a HAZOP can be greatly affected based on a poor choice of paramters and guidewords. In this work, we use the a table provided by the CandelaLearning[1] (see

Parameter	*More*	*Less*	*None*	*Reverse*	*As well as*	*Part of*	*Other than*
Flow	high flow	low flow	no flow	reverse flow	deviating concentration	contamination	deviating
Pressure	high pressure	low pressure	vacuum		delta p		explosion
Temperature	high temperature	low temperature					
Level	high level	low level	no level		wrong level		
Time	too long/too late	too short/too soon	sequence step skipped	backwards	missing actions	extra actions	wrong time
Agitation	fast mixing	slow mixing	no mixing				
Reaction	fast reaction/runaway	slow reaction	no reaction				unwanted reaction
Startup or Shutdown	too fast	too slow			actions missed		wrong recipe
Draining or Venting	too long	too short	none		deviating pressure	wrong timing	
Inertiing	high pressure	low pressure	none			contamination	wrong material
Utility failure (instrument air, power)			failure				
Maintenance			none				
Vibrations	too high	too low	none				wrong frequency

Table 4.2: List of guidewords [1]

table 4.2). This table provides the mappings that will be used in the HAZOP for CacTiE.

HAZOP Table

After following the steps outlined, we generated a HAZOP table containing pertinent information (see table 4.3). We systematically reviewed each of the key components and their parameters. We then developed the appropriate deviations based on guidewords established previously.

4.3.8 HRA

Overview

As mentioned previously, an HRA has to do with systemizing and analyzing human factors as it relates to risk. We will analyze the Setup operations for starting up CacTiE for normal operation. The bound of our system is CacTiE and specifically the interactions involved in the setting up process. [3]

Task Analysis

In all 4 steps, there are elements of human involvement. We have analyzed all 4 steps in this HRA for that reason. The process of setting CacTiE up for normal operations:

1. Make sure that the enclosure is closed and no equipment is missing
2. Fill tanks with water. Hot water should go in the "chemical tank" and cold water in the "raw water tank".
3. Turn the Raspberry Pis and the monitor ON
4. Log in to the SCADA. Check to ensure that there are no alerts that could signify an equipment failure.

HRA Modeling

Following each of the tasks, smaller sub-steps and associated failure and recovery actions are developed (see table 4.4).

Component	*Deviation*	*Cause*	*Consequence*	*Recommended Action*
Raspberry Pi's	High Temp	Inadequate spacing	Premature failure	Ventilate equipment
	Power failure	Surge protector flipped	Raspberry Pi won't boot	Use a robust surge protector
	High humidity	Tank or pipe leaking	Equipment failure	Use a robust surge protector
PLC	High Temp	Inadequate ventilation	Premature failure	Space out equipment or ventilate the enclosure
	Power failure	Surge protector flipped	PLCs won't boot	Use a robust surge protector
HMI	High Temp	Inadequate ventilation or spacing	Premature failure	Space out the equipment or ventilate housing
	Power failure	Surge protector flipped	The HMI won't boot	Use a robust surge protector
Tanks	High pressure	Water build up	The tank fails	Install an overflow relief valve
Piping	High pressure	Water build up	The pipe fails	Program SCADA warning and add inspect pipes
Monitor	Power failure	Surge protector flipped	The Display turn ON	Use a robust surge protector
	High humidity	Cause	Consequence	Recommendation
Switch/Router	High Temp	Inadequate spacing	Premature failure	Space out the equipment
	Power failure	Surge protector flipped	Network devices won't boot	Use a robust surge protector
	High humidity	Tank or pipe leaking	Equipment failure	Use a robust surge protector, add a 2nd enclosure
Pump	High pressure	Water build up	premature wear/failure	Use one-way valve
	Power failure	Power supply failure	Equipment stops	Use a robust surge protector
Float Sensors	Power failure	Power supply failure	Equipment stops	Use a robust surge protector
	Power failure	Power supply failure	Equipment stops	Use a robust surge protector
	Power failure	Power supply failure	Equipment stops	Use a robust surge protector

Table 4.3: HAZOP for CacTiE

Step	*Failure Action*	*Recovery Action*
1	Operator fails to identify all of the equipment	Operator successfully identifies all equipment
1	Operator fails to completely close the enclosure	Operator successfully closes the enclosure
2	Operator fails to identify the unique tanks	Operator successfully identifies all of the unique tanks
2	Operator fails to place hot water in "chemical tank"	Operator successfully places hot water in "chemical tank"
2	Operator fails to place cold water in "chemical tank" or "holding tank"	Operator successfully places cold water in "chemical tank" and "holding tank"
3	Operator fails to identify the Monitor or the Raspberry Pi's	Operator successfully identifies the monitor and the Raspberry Pi's
3	Operator fails to identify the monitor power button	Operator successfully identifies the monitor power button
3	Operator unplugs the Raspberry Pi's without a graceful shutdown	Operator successfully powers down the Raspberry Pi's
4	Operator fails to identify the workstation	Operator successfully identifies the workstation
4	Operator fails to log into the workstation	Operator successfully logs onto the workstation
4	Operator fails to connect to the SCADA	Operator successfully connects to the SCADA
4	Operator fails to authenticate to the SCADA web-interface	Operator successfully authenticates to the SCADA web-interface
4	Operator fails to identify web-interface's "alerts"	Operator successfully identifies web-interface's "alerts"

Table 4.4: HRA for CacTiE startup procedure

4.3.9 Fault Tree Analysis

Overview

Fault Trees are conducted to determine potential causes of high risk events. It is particularly effective for creating mitigation to high-valued risks. This form of analysis is performed after the HAZOP and HRA because high risk events detected in these analysis will feed into the fault trees. Based on the preliminary results, the fault tree analysis will be performed on the following events: (1) PLC failure (2) Pump 2 failure [3]

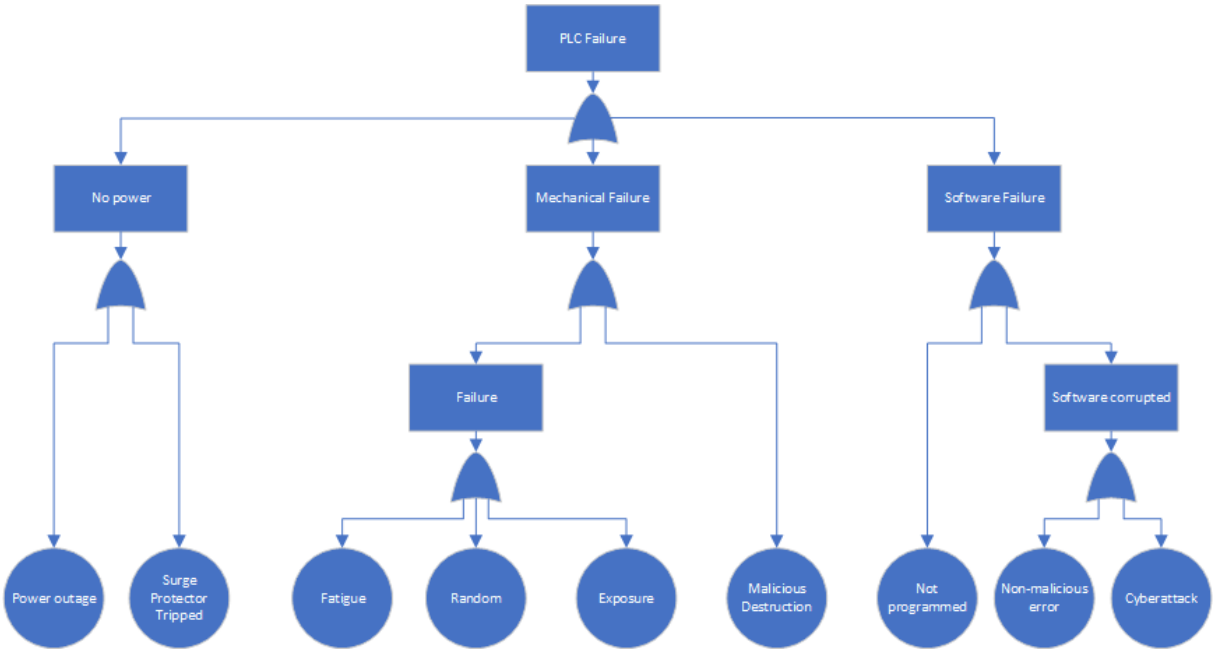


Figure 4.6: Fault tree for PLC failure

Fault Tree: PLC Failure

The PLCs control the I/O but event more so, they are one of the more expensive components in the testbed. A PLC can fail (in terms of operation) for a variety of reasons. If there is no power, there could have been a surge or the power might be out. Mechanical failures can result from traditional failures or intentional attacks. The class of traditional failures may take on the form of fatigue, random, or exposure related failures. In the case of CacTiE, exposure would be excess noise, heat, or moisture. Software related failures can result from a lack of programs being loaded or problems with said programs. The programs could be suffering from an unintentional bug or a malicious cyber-attack. (see Figure 4.6)

Fault Tree: Pump Failure

Pump failure was one of the high risk events. If the pump cannot circulate water, the system no longer functions. A pump can fail from electrical . An electrical fault can be a short or a result from a power surge. Additionally, the pump can fail if the water pressure

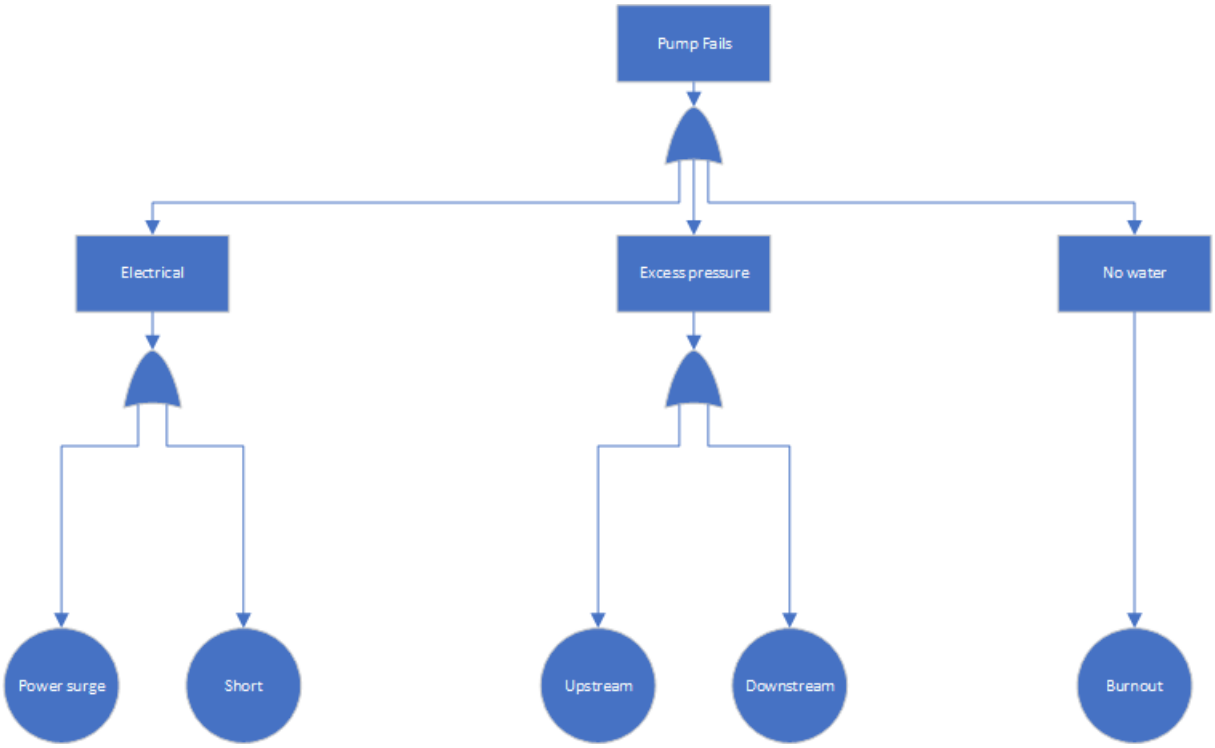


Figure 4.7: Fault tree for a pump failure

exceeds it's rating. The excess pressure can occur either upstream or downstream of the pump. Finally, the lack of water can result in pump burnout. (see Figure 4.7). Note that due to the fact that cheaper pumps, such as the one purchased, have little to no documentation, a qualitative analysis is utilized.

4.3.10 Results

Common Trends

Many of the risks involved adequate and controlled power distribution to each of the components. Additionally, protection of the equipment from over heating and exposure to water appear to be not thoroughly covered.

4.3.11 Discussion

Major takeaways

Much of the testbed relies on human operations. It is just as critical to consider the

policies and implementations of the testbed as it is to craft it's design so as to promote greater safety and risk reduction.

Limitations

More work could be done to further quantify probabilities to gain a clearer picture of the overall risk profile for the testbed.

4.4 CacTiE: Risk Management

4.4.1 Identified Risks

In the risk assessment, there were a variety of observed risks to the system. (1) In order to save on cost, less robust equipment was used for the servers and networking components. However, these components are not adequately protected against the humid and potentially wet environment. (2) Measures were taken to ensure that the enclosure selected would withstand the wet environment of CacTiE. However, the enclosure has no ventilation and has a rather small foot print, making it easy for equipment to overheat. (3) There are currently no documents explaining how to operate the CacTiE safely. There is considerable area for students to danger themselves based on lack of details in the operation of the testbed. Specifying lower temperatures for the water and adding hot water warnings to the procedure. (4) Exposed electrical equipment could be a source of shock when water is added to the tanks. The network components, servers, and the surge protector are all uncovered.

4.4.2 Mitigation Goals

(1) Improve the safety of the system so that way users are not in any undo danger when using CacTiE. (2) Decrease the chance for normal use of the testbed to result in premature failure of equipment.

4.4.3 Assumptions

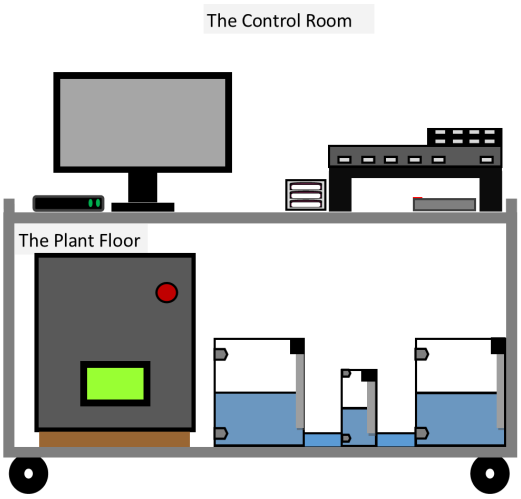
(1) This system will be generating a relatively low amount of noise. This level of noise will likely not be enough to consider in terms of potential degradation of equipment of attenuation of signals. (2) This system will not be have a hazardous potential of energy release, as such lock-out tag procedure will not be necessary. (3) CacTiE will not be dealing with high voltage (over 120VAC).

4.4.4 Resulting Recommendations

(1) Addition of a second enclosure or partition to separate the networking equipment, thin clients, surge protector and the Raspberry Pi from water exposure. These enclosures will be NEMA 12 standard which is more than sufficient at dealing with the level of water exposure that we should be worried about. (2) Introduction of fans with mesh covers to help cool the equipment and manage thermals more carefully. (3) Development of a more concise policy and procedure for setup, operation, and closing of the CacTiE testbed. (4) If costs allow, the addition of inline fuses to the PLCs, HMI, and SCADA. These are the most expensive and important pieces of equipment. This will help mitigate against their loss in the event of an electrical surge.

4.4.5 Societal Impacts

Based on our findings in the literature review, there is s distinct lack of coverage of risk in educational ICS testbeds. There is already a huge safety culture in ICS. However, this work will provide methods for concrete analysis of testbed risk (especially safety). This will hopefully influence the safety culture as it currently stands in educational ICS environments.



(a) Mockup of CacTiE Physical layout

(b) CacTiE Assembled

Figure 4.9: CacTiE phusical infrastructure

4.4.6 Discussion

There are many field that can serve from improved automation and much of ICS risk profile can potentially be improved by reducing human reliability. However, in an educational platform made for humans, it is not prudent to attempt this sort of approach to risk management in ICS educational testbeds. It has proven to be a distinct challenge to improve the risk profile in regards to human reliability. A takeaway from this work is that more clear, concise language in operational manuals can be effective in dealing with this challenge.

4.5 CacTiE: Construction

4.5.1 Physical Assembly

CacTiE's construction began by first obtaining suitable containers to serve as tanks. We utilized 8 and 1.8 quart pitchers for the tanks. These tanks were fitted with bulkhead adapters with threaded adapters to fit (1) 1/4" ID tubing barbs (2) 1/2" threaded inserts

for the float valves (3) 1/4" threaded inserts near the lid for the thermocouples. Rubber washers and adhesive silicone were used to create water-tight fittings were leaks presented themselves.

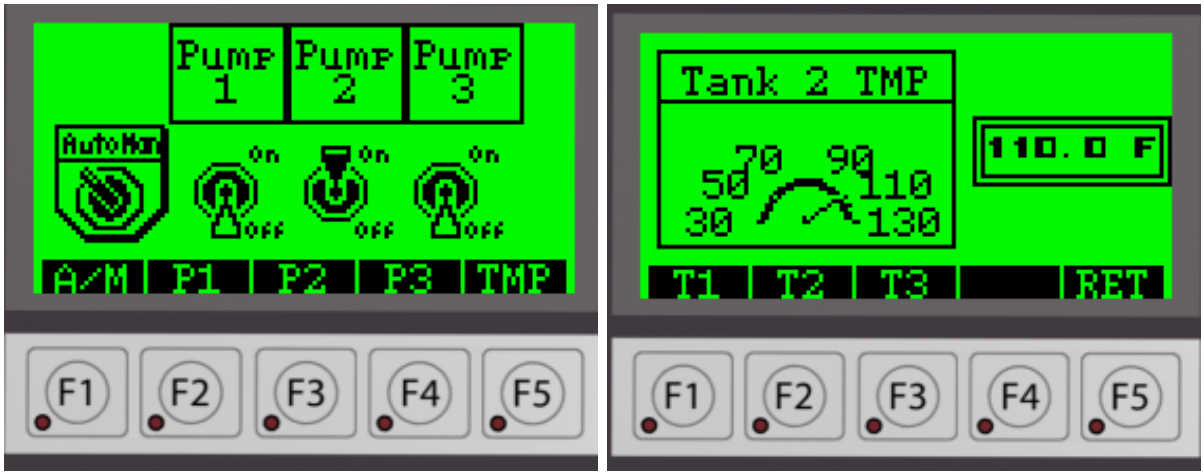
After ensuring that each tank was completely water-tight, the ICS cabinet began to be assembled. This consisted of cutting out 2 holes for both power, data communications, and I/O to be cabled. This connection was sealed with both a 1/4" and 5/16"rubber grommet. Further, the HMI and Emergency Stop had to have holes cut out for them. The final modification came when 2 rows of 35mm DIN railing (used for mounting hardware) and 2 rows and 2 columns of 1" wide wire duct where cut and installed. It should be noted that the DIN railing and wire duct were installed to a removable back panel inside of the cabinet. This is standard practice for equipment modification and installation at later steps.

The control components were then installed to the din railing after the ducting was screwed in. Starting from the top left, the following components were installed: 1.3A 24VDC CLICK PLC power supply, the CLICK Ethernet PLC, and an analog module made specifically for thermocouples (AutomationDirect C0-04THM). The bottom DIN rail was utilized entirely for terminal blocks, or specialized fittings that provide secure connection for wiring to and from I/O.

Finally, the Raspberry Pis, Monitor, KVM Switch, Bluetooth keyboard, router, and switches are all installed on the top of the cart. These are all connected to a surge suppressor.

4.5.2 CLICK Ethernet PLC and C-More Micro HMI

With the CLICK PLC physically installed in the ICS cabinet, that only leaves a few things left to do before programming the CLICK. First, we route a standard 110VAC power cable through the 1/4" bushing in the side of the cabinet. The 3 wires: line, neutral, and ground are wired to the CLICK power-supply. The CLICK has 3 terminals



(a) Pump activation (b) Temperature values

Figure 4.10: The programmed HMI

on the bottom of it that a the 24VDC Positive, Negative, and the ground are connected. After doing this, the CLICK PLC Programming software is installed in order to program the ladder logic. This software is free and is found on AutomationDirect's website.[60]

The ladder logic itself is not particularly different on the CLICK software. The software uses a simplified set of instructions, but it is still highly capable. The Ethernet interface on the CLICK can be used to connect the PLC to the CLICK Programming Software for programming. The C-More can only be programmed with a special USB to Serial communication cable but also comes with it's own free software to program on. See figure 4.10

4.5.3 Wiring

The water tanks then have the float and temperature sensors installed. At this point the 1/4" and 3/8" tubing is installed. Now the Ethernet, thermocouple, pump and float sensor wires can be fed through the 5/16" rubber grommet that has been installed on the ICS cabinet. The sensors and pumps are then wired to the CLICK PLC I/O in a sinking configuration. The Emergency stop will also be wired from the door in a similar manner. The thermocouple can then be wired to the special thermocouple analog input module.

Finally, plug in the HMI with the included serial cable to the CLICK PLC's available port. The cables are routed through the ducting and when complete the covers are placed over them.

4.5.4 Network Configuration

The Linksys WRT AC1900 router was selected for it's cost and due to the fact that it support the OpenWRT firmware. OpenWRT [78] is a Open Source Linux kernel that can be flashed to some routers and switches. This allows for the extension of functionality and will effectively result in feature not natively supported in the factory firmware (vlan tagging and granular custom firewall rules). The firmware was found on the OpenWRT website https://openwrt.org/ by navigating a table. After dropping the new firmware onto the router we enabled DHCP and configured vlans for Enterprise, Control, Supervisory, and Management. The other switches are used to allow for multiple machines to connect to the Enterprise and Supervisory networks.

4.5.5 SCADA and Workstations

CacTiE utilizes a SCADA server to give students an opportunity to understand an important piece of control infrastructure. An Open Source Software is used called ScadaBR. ScadaBR was configured to provide alerts, watch set points, host a web-based HMI, and provide log aggregation. The SCADA software was installed onto one of the Raspberry Pi 3s. This Pi ran the Ubuntu 16.04 server OS. It was found that the default Raspbian OS was not compatible with the ScadaBR project [56]. The remaining 2 Pi's had a very straightforward Raspbian Desktop OS which can be found on the Raspberry Pi webpage https://www.raspberrypi.org/downloads/.

4.5.6 BoM

This subsection outlines the total bill of material for the CacTiE testbed. This details each part and product used to develop the testbed. This list does not include a list

of tools used to assemble the components. Not all of these tools are required. The tools include: circular saw, dremel, drill, impact driver, multimeter, soldering iron, wire strippers, hacksaw, file, and a flat head screwdriver. It should also be noted that a 24" HDMI Computer Monitor and 5 Ethernet Patch cables were donated by the University of Idaho's Secure and Dependable Computing Systems office. The value of these items is NOT included in the Bill of Materials.

4.6 Literature Review

This section provides a literature review of risk assessments of ICS testbeds. Additionally, some papers are included on other physical testbeds for ICS that focus on education.

4.6.1 Procedure

Other Small-scale, ICS Cybersecurity Testbeds

We performed a review of other physical-based, ICS related testbeds. Relevance keyword searching using the keyword set:

"Industrial Control System testbed" "SCADA testbed" "cybersecurity" "ICS Testbed"

The searching only included papers over the past 10 years using Google Scholar. There were 5 hits.

Risk Assessment: Books and Standards

This researcher was already in possession of a few texts that proved useful to risk assessment in control systems. First, the Ostrom *et al.* text [3] covers risk assessment techniques in a general scope. It has some industrial related risk assessment examples. Second, Knapp *et al.* have a text[79] on Industrial Networks. This literature review included their chapter on Risk and Vulnerability Assessments. They also included a fine list of relevant standards. The standards reviewed are relative to ICS risk assessment

in general. Standards referenced reviewed are: BSI 100-3 [73], CERT OCTAVE [71], ENISA: Principles and Inventories for Risk Management/Risk Assessment Methods and Tools [72]. Additionally, ISO/IEC 31010 and 31000 are included through Cross' paper [74] covering the standards. These standards are covered in Section 4.3.2.

4.7 Chapter Conclusion

In summary, CacTiE is an approximately $1600 solution for Cybersecurity education in the realm of control systems. This platform obtains small scale and cost by simplifying the industrial process modeled and only using equipment that is necessary to lend to education. However, this chapter also points out the lack of ICS testbeds using cost-effective physical infrastructure and the state of risk assessment methodologies for testbeds. We examined standards pertaining to control systems (ex: NEMA, BSI, ENISA) in general and applied risk assessment tools used in industry. The performance of the combined HAZOP and HRA pointed out flaws in the preliminary design and lack of documentation on safe system usage. Fault Trees revealed the ways that sensitive equipment in CacTiE can fail and revealed hardening techniques that can be used to improve the overall safety of the system. Finally, this chapter describes the build process and gives a breakdown of the parts used to develop the CacTiE system.

Product	*Unit Cost (USD)*	*Quantity*
5VDC Water Pump	$10.59	3
24VDC NC/NO Float Sensor	$13.00	6
Type-K Thermocouple	$18.50	3
Emergency Stop Button & Label	$15.75	1
CLICK Standard Ethernet PLC	$163.00	1
Thermocouple Analog Input Module	$155.00	1
C-More Micro HMI	$98.00	1
HMI Programming Cable	$44.50	1
1.3A 24VDC Power-supply	$39.00	1
1.5A 5VDC Power-supply	$29.00	1
Raspberry Pi 3 w/Power-supply	$44.98	3
Raspberry Pi 3 Tower Case	$40.49	1
32GB SD Card	$7.49	3
Linksys WRT AC1900 Router	$149.99	1
5 Port Dumb Switch	$17.99	2
KVM HDMI/USB Switch	$65.99	1
Bluetooth Keyboard	$18.78	1
Monitor Riser	$16.99	1
Surge Suppressor	$25.99	1
8-quart Container	$24.00	2
1.8-quart Container	$17.49	1
Cart	$70.36	1
NEMA 12 Fiberglass Enclosure (12”x10”x5”)	$95.25	1
35mm DIN rail (1 meter)	$12.50	1
1” Wide Wire Ducting (2 meters)	$17.50	1
1/4” ID Vinyl Tubing (3’)	$1.74	1
3/8” ID Vinyl Tubing (1.5’)	$1.47	1
Clear Epoxy	$5.76	1
3’ 16AWG Copper Stranded Wire	$5.00	1
Nylon 1/4” Barb to 3/8” Bushing	$0.69	6
Nylon 1/4” Barb to 3/8” Barb	$0.69	3
Nylon 1/4” Hex Nipple	$0.69	3
Nylon 3/4” to 1/2” Bushing	$1.79	6
Nylon 3/4” to 1/4” Bushing	$1.79	3
Nylon 3/8” to 1/4” Bushing	$1.79	3
Nylon 3/8” to 1/2” Bushing	$1.79	6
1/2” ID Rubber Washer	$1.50	6
1/4” One-way Check Valve	$5.99	3
	TOTAL:	*$1591.29*

Table 4.5: BoM for CacTiE

Chapter 5: Summary and Conclusions

In this work, we have performed an in-depth analysis of the state of the water cybersecurity testbed field. Using this knowledge, we explored the possibilities of utilizing virtualization to create cost-effective testbed. To do this, an analysis of current pedagogical methods and frameworks is performed. The topics and capabilities of modern ICS testbeds formed the common criteria used to give the virtualized However, virtualization relies on expensive technology to perform the virtualization and requires further expertise. A solution arose in the form of CacTiE, were the virtual testbed took form in a physical space. After performing a HAZOP, HRA, and Fault Tree analysis it was found that CacTiE's design could stand to be more resilient and documentation should be made for safe operation of the system. This thesis further showed the improved implementation of CacTiE and discussed cost and explained the construction and configuration processes.

It is our firm belief that CacTiE's design can be used to bridge the gap in education for Cybersecurity students interested in critical infrastructure. This system provide a cost-effective and simple solution that is highly modular. This design can be adapted to the needs or wants of higher education institutions and prosumers, depending on the budget and expertise.

Bibliography

[1] R. A. HAZOP, "Epa's water security test bed." https://courses.candelalearning.com/riskassessment/chapter/hazop/, 2015.

[2] S. Caltagirone, P. Ortman, S. Melton, D. Manz, K. King, and P. W. Oman, "Design and implementation of a multi-use attack-defend computer security lab," in *Proceedings of the 39th Hawaii International Conference on System Sciences (HICSS-39)*, vol. 9, (Kauai, Hawaii, U.S.A.), p. 220c, IEEE Computer Society, January 2006.

[3] L. T. Ostrom and C. A. Wilhelmsen, *Risk Assessment: Tools, Techniques, and their Applications*. Wiley, 2012.

[4] A. P. Mathur and N. O. Tippenhauer, "Swat: A water treatment testbed for research and training on ics security," in *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*, pp. 31–36, IEEE, 2016.

[5] K. T. Moss, "Water treatment and distribution simulation for a scada security testbed," Master's thesis, University of Louisville, 2012.

[6] R. D. Doering, C. S. Bauer, and S. G. R., "A low-cost computer-based training simulator for waste water treatment plant operators," in *Proceedings of the 1982 Winter Simulation Conference*, IEEE, 1982.

[7] E. W. I. P. Division, "Epa's water security test bed." Internet, 2017.

[8] A. Mathur, "Secwater:a multi-layer security framework for water treatment plants," in *CySWATER 2017*, pp. 29–32, ACM, 2017.

[9] S. Adepu and A. Mathur, "Distributed detection of single-stage multipoint cyber attacks in a water treatment plant," in *ASIA CCS '16 Proceedings of the 11th ACM*

on Asia Conference on Computer and Communications Security, pp. 449–460, ACM, 2016.

[10] D. HadÅ¾iosmanovic, D. Bolzoni, and P. H. Hartel, "A log mining approach for process monitoring in scada," *International Journal of Information Security*, vol. 11, pp. 231–251, 2012.

[11] M. Caselli, E. Zambon, and F. Skargl, "Sequence-aware intrusion detection in industrial control systems," in *CPSS 2015*, pp. 13–24, ACM, 2015.

[12] S. Shrivastava, S. Adepu, and A. Mathur, "Design and assessment of an orthogonal defense mechanism for a water treatment facility," *Robotics and Autonomous Systems*, vol. 101, pp. 114–125, 2018.

[13] M. A. Umer, A. Mathur, K. N. Junejo, and S. Adepu, "Integrating design and data centric approaches to generate invariants for distributed attack detection," in *Cyber-Physical Systems Security & Privacy (CPS-SPC) 2017*, pp. 131–136, ACM, 2017.

[14] E. Kang, S. Adepu, and D. Jackson, "Model-based security analysis of a water treatment system," in *Software Engineering for Smart Cyber-physical Systems (SEsCPS)'16*, pp. 22–28, ACM, 2016.

[15] S. Papa, W. Casper, and T. Moore, "Securing wastewater facilities from accidental and intentional harm: A cost-benedit analysis," *International Journal of Critical Infrastructure Protection*, vol. 6, pp. 96–106, 2013.

[16] M. Suby and F. Dickson, "The 2015 (isc)2 global information security workforce study," April 2015.

[17] ISACA, "State of cybersecurity implications for 2016." Online, February 2016.

[18] NIST, "Report on securing and growing the digital economy," 2016.

[19] Center for Strategic and International Studies, "Hacking the skills shortage," July 2016.

[20] CyberSeek, "Cybersecurity supply/demand heat map." Online, January 2018.

[21] Verizon, Inc., "2016 data breach investigations report." Online, April 2016.

[22] Verizon, Inc., "2017 data breach investigations report." Online, June 2017.

[23] Verizon, Inc., "2018 data breach investigations report." Online, June 2018.

[24] M. Suby and F. Dickson, "The 2017 global information security workforce study: Benchmarking workforce capacity and response to cyber risk," December 2017.

[25] D. Conte de Leon, C. Goes, M. Haney, and A. Krings, "Adles: Specifying, deploying, and sharing hands-on cyber-exercises," *Computers and Security*, vol. 74, no. May 2018, pp. 12–40, 2018.

[26] Dragos, Inc., "Threat proliferation in ics cybersecurity: Xenotime now targeting electric sector, in addition to oil and gas." Online, June 2019.

[27] B. Perelman, "The rise of ics malware: How industrial security threats are becoming more surgical." https://www.securityweek.com/rise-ics-malware-how-industrial-security-threats-are-becoming-more-surgical, 2018.

[28] A. A. Jillepalli, D. Conte de Leon, and J. Alves-Foss, "Operational characteristics of modern malware: Pco threats," in *Proceedings of the Fifth Cybersecurity Symposium*, CyberSec-2018, pp. 5:1–5:6, ACM, 2018.

[29] A. A. Jillepalli, D. Conte de Leon, Y. Chakhchoukh, M. Ashrafuzzaman, B. K. Johnson, F. T. Sheldon, J. Alves-Foss, P. Tosic, and M. A. Haney, "An Architecture for HESTIA: High-level and Extensible System for Training and Infrastructure risk Assessment," *International Journal of Internet of Things and Cyber-Assurance*, vol. 2, no. 5, pp. 103–121, 2018.

[30] SANS, "Sans website." https://www.sans.org/. Visited: April 2019.

[31] ICS-CERT, "Training available through ics-cert." https://ics-cert.us-cert.gov/Training-Available-Through-ICS-CERT. Visited: April 2019.

[32] Cyberdegrees.org, "Cybersecurity degrees." Online, 2019. Visited: June 2019.

[33] A. A. Jillepalli, D. Conte de Leon, and F. T. Sheldon, "CERES NetSec: Hands-on Network Security Tutorials," *Journal of Computing Sciences in Colleges*, vol. 33, pp. 88–96, May 2018.

[34] D. Conte de Leon, A. A. Jillepalli, V. J. House, J. Alves-Foss, and F. T. Sheldon, "Tutorials and Laboratory for Hands-On OS Cybersecurity Instruction," *Journal of Computing Sciences in Colleges*, vol. 34, pp. 242–254, October 2018.

[35] T. Morris, A. Srivastava, B. Reaves, W. Gao, K. Pavurapu, and R. Reddi, "A control system testbed to validate critical infrastructure protection concepts," *International Journal of Critical Infrastructure Protection*, vol. 4, no. 2, pp. 88–103, 2011.

[36] E. Sitnikova, E. Foo, and R. B. Vaughn, "The power of hands-on exercises in scada cyber security education," *IFIP Advances in Information and Communication Technology*, vol. 406, pp. 83–94, 2013.

[37] E. Foo, M. Branagan, and T. Morris, "A proposed australian industrial control system security curriculum," in *Proceedings of the Hawai'i International Conference on System Sciences 2013*, pp. 1754–1762, IEEE, 2013.

[38] E. G. Plumley, "A framework for categorization of industrial control system cyber training environments," Master's thesis, U.S. Air Force Institute of Technology, March 2017.

[39] T. Yardley, S. Uludag, K. Nahrstedt, and P. Sauer, "Developing a smart grid cybersecurity education platform and a preliminary assessment of its first application," in

Proceedings of the Frontiers in Education Conference 2014, pp. 1–9, IEEE, February 2014.

[40] P. C. Blumenfeld, E. Soloway, R. W. Marx, J. S. Krajcik, M. Guzdial, and A. Palincsar, "Motivating project-based learning: Sustaining the doing, supporting the learning," *Educational Psychologist*, vol. 26, no. 3-4, pp. 369–398, 1991.

[41] E. Harris, "Success in industrial control system cyber security training," tech. rep., Idaho National Laboratory, January 2016.

[42] I. A. Oyewumi, A. A. Jillepalli, P. Richardson, M. Ashrafuzzaman, Y. Chakhchoukh, B. K. Johnson, M. A. Haney, F. T. Sheldon, and D. Conte de Leon, "ISAAC: The idaho CPS smart grid cybersecurity testbed," in *2019 3rd IEEE Texas Power and Energy Conference (TPEC)*, pp. 1–6, Feb 2019.

[43] H. Gao, Y. Peng, K. Jia, Z. Dai, and T. Wang, "The design of ICS testbed based on emulation, physical, and simulation (EPS-ICS Testbed)," *Proceedings - 2013 9th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP 2013*, pp. 420–423, 2013.

[44] B. Green, A. Le, R. Antrobus, U. Roedig, D. Hutchison, and A. Rashid, "Pains, Gains and PLCs: Ten Lessons from Building an Industrial Control Systems Testbed for Security Research," *10th USENIX Workshop on Cyber Security Experimentation and Test (CSET '17)*, pp. 1–8, 2017.

[45] T. J. Williams, "The purdue enterprise reference architecture," *Computers in Industry*, vol. 24, pp. 141–158, sep 1994.

[46] Modbus, "Modbus specifications." http://www.modbus.org/specs.php. Visited: June 2019.

[47] IEEE Standards Association, "Ieee standard for electric power systems communications - distributed network protocol (dnp3)," Tech. Rep. 1815-2012, IEEE, 2012.

[48] VMware, "What is vmware vsphere?." https://www.vmware.com/products/vsphere.html, 2019.

[49] GIAC, "Certifications: Industrial control systems." https://www.giac.org/certifications/ics. Visited: April 2019.

[50] U.S. DoE, "Department of energy's cyberforce competition." https://cyberforcecompetition.com/. Visited: April.

[51] IIC Productions, "Ics/scada capture the flag competition." https://www.infosec-city.com/sg18-ctf-ics-scada, 2018.

[52] Hack in the Box Security Conference, "2018 scada ctf." https://conference.hitb.org/hitbsecconf2018dxb/scada-ctf-village-by-nshc. 2018.

[53] Netgate, "pfSense Firewall Appliance Features." https://www.netgate.com/solutions/pfsense/features.html, 2019.

[54] Microsoft, "pfSense Firewall Appliance Features." https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-2000-server/bb727030(v=technet.10), 2009.

[55] Microsoft, "Group policy for beginners." https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-7/hh147307(v=ws.10), 2012.

[56] Sensorweb, B. R., "SCADABR." http://www.scadabr.com.br/, May 2019.

[57] T. Alves, "OpenPLC." https://www.openplcproject.com/, 2019.

[58] Velocio.net, "Programmable Controllers." http://velocio.net/, 2017.

[59] Allen-Bradley, "Programmable Controllers." https://ab.rockwellautomation.com/Programmable-Controllers, 2019.

[60] AutomationDirect, "Programmable Controllers." https://www.automationdirect.com/adc/shopping/catalog/programmable_controllers, 2019.

[61] Divelbiss, "Programmable Controllers." http://www.divelbiss.com/Products/CatDetails.asp?ProdCatID=1, 2019.

[62] Siemens, "SIMATIC Controllers." https://www.industry.usa.siemens.com/automation/us/en/automation-systems/industrial-automation/, 2019.

[63] Schneider Electric, "PLC Controllers for Industrial Machines." https://www.schneider-electric.com/en/product-category/, 2019.

[64] CDC, "Water Treatment." https://www.cdc.gov/healthywater/drinking/public/water_treatment.html, 2015.

[65] AutomationDirect, "C-more Micro HMI: EA3-S3ML-RN." https://www.automationdirect.com/adc/shopping/catalog/hmi_(human_machine_interface)/c-more_micro_panels/3_inch_panels_-a-_accessories/ea3-s3ml-rn, 2019.

[66] Linksys, "Linksys wrt1900acs dual-band wi-fi router, 1.6 ghz cpu." https://www.linksys.com/us/p/P-WRT1900ACS/. 2018.

[67] AutomationDirect, "Float level switch: Fls-hs-100." https://www.automationdirect.com/adc/shopping/catalog/process_control_-a_measurement/level_sensors_-a_controllers/float_level_switches/fls-hs-100#btn-bar-a. 2019.

[68] AutomationDirect, "Temperature sensor: Thmk-a01l04-01." https://www.automationdirect.com/adc/shopping/catalog/sensors_-z_encoders/temperature_sensors_-a_transmitters/thermocouple_sensors/sensors_(adjustable_immersion)/thmk-a01l04-01. 2019.

[69] Amazon, "Dc 4v-6v 150l/h mini brushless submersible water pumps motor micro water cooling pump with usb connector." https://www.amazon.com/4V-6V-Brushless-Submersible-Cooling-Connector/dp/B01LWXV7DE/ref=sr_1_6?ie=UTF8&qid=1537967141&sr=8-6&keywords=water+pump+5v. 2019.

[70] Raspberry Pi Foundation, "Buy a raspberry pi 4 model b - raspberry p." https://www.raspberrypi.org/products/raspberry-pi-4-model-b/. 2019.

[71] R. A. Caralli, J. F. Stevens, L. R. Young, and W. R. Wilson, "Introducing octave allegro: Improving the information security risk assessment process," tech. rep., Carnegie Mellon University, 2007.

[72] ENISA, "Risk management : Implementation principles and inventories for risk management / risk assessment methods and tools," tech. rep., European Union Agency for Network and Information Security, June 2006.

[73] BSI, "BSI Standard 100-3 - Risk analysis based on IT-Grundschutz," *Bundesamt für Sicherheit in der Informationstechnik (BSI)*, 2008.

[74] J. Cross, "ISO 31010 Risk Assessment Techniques and Open Systems," in *Open Systems Dependability*, 2017.

[75] NEMA, "Nema ics 6-1993: Industrial control and systems: Enclosures," tech. rep., National Electrical Manufacturers Association, 2016.

[76] NEMA, "Nema standards publication no. ics 1-2000: Industrial control and systems: General requirements," tech. rep., National Electrical Manufacturers Association, 2008.

[77] NEMA, "Nema standards publication ics 1.3-1986: Preventive maintenance of industrial control and drive system equipment," tech. rep., National Electrical Manufacturers Association, 2015.

[78] The OpenWrt project, "OpenWrt Project: Welcome to the OpenWrt Project." https://openwrt.org/start. Accessed: 12th November 2019.

[79] E. D. Knapp and J. Langill, "Chapter 8 - risk and vulnerability assessments," in *Industrial Network Security*, Syngress, 2015.