5-2023

LAYING THE FOUNDATION FOR A MINIATUAIRZED SCADA TESTBED TO BE BUILT AT CSUSB

Ryan Perera

LAYING THE FOUNDATION FOR A MINIATUAIRZED SCADA TESTBED TO BE BUILT AT CSUSB

A Project

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Information Systems and Technology

by

Ryan Michael Perera

May 2023

LAYING THE FOUNDATION FOR A MINIATUAIRZED SCADA TESTBED TO BE BUILT AT CSUSB

A Project

Presented to the

Faculty of

California State University,

San Bernardino

by

Ryan Michael Perera

May 2023

Approved by:

Vincent Nestler PhD, Committee Member, Project Chair

Conrad Shayo PhD, Committee Member, Reader

Conrad Shayo PhD, Committee Member, IDS Department Chair

ABSTRACT

This culminating experience sought to lay the foundation for a miniaturized physical SCADA testbed to be built at California State University San Bernardino to enable students to apply the cybersecurity knowledge, skills and abilities in a fun and engaging environment while learning about what SCADA is, how it works, and how to improve the security of it. This project was conducted in response to a growing trend of cybersecurity attacks that have targeted our critical infrastructure systems through SCADA systems which are legacy systems that manage critical infrastructure systems within the past 10 years. Since SCADA systems require constant availability, it makes it hard to test the security of these devices which is why testbeds have been designed to analyze how a cyber-attack affects these systems in a safe environment. To build a SCADA testbed at CSUSB this project designed a requirements documentation based on the following questions so that the next person that wants to accomplish this task can take the requirements outlined and build a miniaturized physical SCADA testbed. To craft the appropriate requirements documentation this project aimed to answer the following questions: Q1. How can a miniaturized SCADA testbed be built for a school environment using open-source architecture? Q2. What critical infrastructure sectors can be easily implemented into a physical SCADA testbed? Q3. Which cyber-attacks can be easily replicable in a SCADA scenario-based environment? Q4. How should SCADA scenarios be modeled for an implementation into this testbed? To answer these questions, research was conducted utilizing scholarly articles on currently available SCADA testbeds, conducted interviews with individuals that have built SCADA testbeds, and distributed a survey to different SCADA professionals to build a requirement documentation for the miniaturized SCADA testbed, which included functional and nonfunctional requirements, use case diagrams and detailed use cases. After gathering the data from 3 different interviews with SCADA professionals and aggregating responses of the surveys we crafted a requirements documentation which includes a requirements documentation, detailed use cases, use case diagrams, and a classes and relationship chart so that the next individual who works on this project can use these ideas and begin construction of a miniaturized SCADA testbed at CSUSB.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

This culminating experience project seeks to lay the foundation for developing a physical SCADA testbed to be designed and implemented for the Cyber Intelligence and Security Organization Club at California State University, San Bernardino (CSUSB) to provide students an opportunity to learn how SCADA systems work, how the security of it can be improved and to develop the cybersecurity skills students learn in their classes which include computer pen testing, computer programming, and digital forensics. To accomplish this, this introduction will define the problem with SCADA systems, by going over what SCADA is, why bad guys are motivated to attack it, what is SCADA vulnerable to, why we need to have more people defending it, designing a test bed with open-source architecture, and the issues setting up the environment. After defining the problems of SCADA in this chapter, Chapter II will be a literature review describing the history of SCADA, significant attacks that occurred on SCADA systems and their impact, and then summarize other SCADA testbed projects. Following the literature review Chapter III will be the methodology chapter that describes how the data was gathered and aggregated using surveys to be distributed out to several types of SCADA professionals and interviews with builders of SCADA testbeds, to build, to build a requirements documentation to lay the foundation for a SCADA testbed to be built at CSUSB. Lastly, Chapter IV will aggregate the information gathered from the surveys and interviews to build a requirements documentation that includes functional and non-functional requirements, flow charts, and use cases to help lay a foundation for the testbed to be built and implemented at CSUSB.

In a speech delivered at Capitol Hill on Feb 10, 2011, Defense secretary under President Obama, Leon Panetta, stated, “The potential for the next Pearl Harbor could very well be a cyber-attack”, (Ryan 2011). Unfortunately, ten years after that speech, he is correct. As technology continues to advance and integrate, so does the possibility of a cyber-attack hitting our nation’s Supervisory Controls and Data Acquisition (SCADA) systems that could cause cascading effects on the country. The National Institute of Standards and Technology, which is used as a national standard to help public and private organizations protect themselves from cybersecurity threats defines SCADA in NIST standard 800-82 Revision 2 as “SCADA is a generic name for a computerized system that can gather and processing data and applying operational controls over long distances. Typical uses include power transmission and distribution and pipeline systems.” (pg. B-16). Before SCADA systems were developed, critical infrastructure was managed manually through individuals who used buttons and levers to ensure equipment was operating at appropriate levels. As these industrial organizations began to grow exponentially, it became cumbersome for individuals to manually manage these systems over long physical distances. This resulted in individuals having to go and interact with devices in remote locations. As a result, SCADA systems were developed to increase the efficiency of plant floors and cover a wide range of communication using automation. When SCADA systems were developed, they were made with the objectives to be constantly available, have long life cycles for technology, and be physically secured to prevent individuals seeking to cause harm to our nation's critical infrastructure. As technology continued to advance, SCADA systems went through multiple generations, starting with the Monolithic generation advancing to the Internet of Things SCADA System currently in use, which has improved the efficiency of critical infrastructure despite being unable to keep up with modern threats. As SCADA systems evolved, they became integrated with the internet by connecting to corporate networks in the Networked SCADA systems generations, which left SCADA systems vulnerable to various types of cyber-attacks for attackers to take advantage of.

The integration of SCADA systems with corporate networks has given an opportunity for a multitude of threat actors to disrupt the United States critical infrastructure. Based on research conducted by Michael Robinson the current trend of SCADA threat actors indicates that nation states, terrorist organization, insider threats, and highly skilled hackers pose the greatest risk to SCADA system, based on the number of resources available to them, the skill level of their hackers, and the ability to achieve higher attack levels which can be depicted in figure 1 below.

Table 1. Actor Attack Level Table
The SCADA Threat Landscape (Robinson, 2013)

Highest Attack Level	Political Motivation	Personal Motivation	War /Peackeeping Motivation
Sweeping and Sustained (5)	Nation States		Nation States
Extensive and Sustained (4)	Terrorist Groups	Insiders Terrorist Groups	
Wide Ranging (3)		High Skill Hackers	
Limited (2)	Hacktivists	Hacktivists Med. Skill Hackers	
Minimal		Low Skill Hackers	

The levels of impact from a cyber-security attack on SCADA systems range from level one, where the attacks have a minimal impact to level five where the effects of the attacks are sustaining and sweeping with the attack crippling multiple parts of the nation’s critical infrastructure. Figure 1 shows that the threat actors that pose the greatest threat to SCADA systems based on highest attack levels would be nation state actors, terrorist organizations, insider threats, and highly skilled actors. Moreover, despite Figure 1 depicting different threat actors such as hacktivists, and medium and low skills hackers at lower levels of impact, it doesn’t mean they can’t get lucky and be able to achieve a higher attack level with wide-ranging effects and everlasting effects. It just means that there is a lower probability of that happening compared to the ones ranked higher on the list (Robinson, 2013).

After discussing the different types of attackers interested in targeting SCADA Systems (Appendix A), how SCADA systems work (Appendix B), What SCADA systems are vulnerable to (Appendix C), how to bridge the cultural gap between IT individuals and SCADA engineers (Appendix D) and the barriers to testing on live SCADA Systems (Appendix E), this paper will continue by describing the reason for using a miniaturized SCADA testbed and the architecture that will be used to build it.. This can be seen in the CyberCity project, which was developed by Ed Skoudis for the military who said “If you tell them [military], that one of your professionals was able to hack into a power grid and turn the lights back on, certain people in the military would say, you just showed me my people can play a videogame” (Skoudis 1). With how Ed Skoudis outlined the philosophy of the military desire for a miniaturized SCADA testbed, development of a requirements documentation to help design a miniaturized physical SCADA testbed would be beneficial to students at CSUSB because it would enable students to see the effects of cybersecurity attacks on SCADA systems in real-time, and apply the cybersecurity knowledge, skills and abilities learned in class to make suggestions on how to improve the security of SCADA systems in a fun, engaging and educational way. To build this miniaturized SCADA environment, this project will use an open-source framework called

SCADABR which was used in a physical SCADA testbed to control water between a two-layer dam.

Before discussing SCADABR architecture, it is important to note that SCADA includes five essential components which are the physical system, cyber-physical link, distributed control systems, network connections, remote monitoring, and control systems to replicate a SCADA system in a testbed environment. As was discussed previously, the physical system will be made up of sensors attached to the SCADA systems to measure appropriate levels of the system and notify the user if something is wrong using alarms. The cyber-link is made up of wires used to transmit signals between the physical system and the PLC. Network connections are connections that enable communication between the RTU, and the PLC. Distributed control systems are devices that provide users with an interface to manipulate the physical devices of the SCADA testbed, such as the actuators and sensors at the cyber-link layer. Lastly, remote monitoring systems are devices found at the control station that enable users to interact with the SCADA system directly such as a historian.

With the essential components of a SCADA testbed described, we can discuss the open-source architecture that will be used for the design and implementation of the SCADA testbed for CSUSB, which will be SCADABR. According to Mazurkiewicz (2016), “SCADABR is a web application written in Java and released under GNU General Public License version 3. It is a typical SCADA system offering logging, reporting, and control functionality. It can communicate with devices using many popular protocols such as Modbus RTU, Modbus TCP, Bacnet, OneWire, Http, and Zwave" (Mazurkiewicz 2). With this, users can develop a SCADA testbed and run it on Apache Tomcat Server, which can be connected locally or on a network server. An example of where SCADABR was modeled was in a testbed using a two-layer dam system designed by Mohammad Alim et.al. where SCADABR controlled the multiple sensors and actuators connected to the dam to get readings on the water levels and determined if water levels of the dams needed to be filled or dispensed to the proper dam. Alim et.al described how it was able to control the sensors and actuators related to the dam through a "Raspberry Pi which hosts the OpenPLC Runtime environment and connects to an UniPi expansion board. The UniPi's relay controls the alarms, valves, and pumps while reading the level sensor data over its analog input ports" (pg. 3). In addition, in the testbed SCADABR was integrated with Python 3 for development and execution to automate cyber-attacks on the testbed, which have the capabilities to manipulate the control, valves, and pumps by using PyModbusTCP library. Lastly, we have the logging of SCADABR. According to Alim et.al, "Data logs include contents of network traffic packets and telemetry metrics. The collected metrics consisting of packet sizes in bytes, timestamps of packet transmission, inter-packet arrival time, packet processing time, protocol overhead, and efficiency, throughput, and client flow for the respective IP" (pg. 5). With this, metrics are aggregated together to give us a representation of the acceptable rate of traffic at which the SCADA testbed operates on.

With the open-source architecture of SCADABR being outlined above, there are some issues when putting the environment together with a physical SCADA testbed. For example, this SCADA testbed focused on building around a two-layer dam system which limits its ability to include multiple different critical infrastructures in a physical tabletop, which this project aims to accomplish. Despite this setback, the logic of how this testbed was designed provides a generalized outline that can be implemented with different critical infrastructure sectors such as the alarm systems, setting up sensors to detect when SCADA systems are working properly and not, and the ability to design different attacks using PyModbusTCP library. In addition, another issue with setting up the environment will be~~,~~ being able to allow students to pen test the environment from their machines as the cyber-attacks depicted in this testbed were built using automated scripts, which will require some modification of the testing environment. Lastly, another concern for setting up this environment will be the size it takes to build this SCADA testbed and be able to apply all the components mentioned earlier in this paper to be able to support multiple critical infrastructure industries to give students an overview of how cyber-attacks on SCADA systems impact different critical infrastructure sectors.

Despite the issues, this project will face with being built for the CISO club at CSUSB, utilizing SCADABR offers a great option to model our miniaturized physical SCADA testbed since it is easy to understand and was utilized in a physical SCADA testbed which was given to the military for studying SCADA systems, and would serve as an excellent addition to the CISO club at CSUSB. In addition, SCADABR incorporates a lot of fun and exciting things students can learn about to develop their cybersecurity skills to make them marketable to the job market such as programming with Python, utilization of Raspberry Pi's, and applying their pen testing skills to conduct different types of cyber-attacks against the miniaturized SCADA testbed. The development of miniaturized SCADA testbed at CSUSB will help develop students' critical thinking skills, which can be used to help improve the future of SCADA security in the years to come as cyber-attacks on these systems continue to grow.

Problem Statement

CyberCity, developed by Ed Skoudis, is a physical SCADA testbed built for the United States Military where they can analyze the impact of a cyber-attack on these systems in real time, which enables the military to improve the response to cyber-attacks on United States critical infrastructure sectors by analyzing the impact different types of cyber-attacks can have on critical infrastructure (Skoudis 1). Projects like CyberCity are crucial for analyzing and improving the security of SCADA systems in a safe environment, since common cybersecurity techniques cannot be used on real SCADA systems since they require constant availability. Since CyberCity is propriety property of the United States Military, it would be a great learning opportunity for schools and organizations to establish their own miniaturized SCADA testbeds like CyberCity to help train the next generation of cyber warriors. Currently at CSUSB, the Cyber Intelligence and Security Organization Club (CISO) is working to develop their own physical miniaturized SCADA testbed to simulate and analyze the damages caused by common cyber-attacks on SCADA systems, which will enable students to apply the cybersecurity skills they learn in classes in a fun and engaging environment, while learning about how SCADA systems work, how they are vulnerable, how they are secured, and how to detect and respond to harm.

Objective

The objective of this project is to lay the foundation for a miniaturized SCADA testbed~~-~~to be built and implemented at CSUSB~~,~~ by creating a requirements document that will leed to the development of a physical SCADA testbed on campus to provide students an opportunity to apply cybersecurity knowledge, skills, and abilities in a fun and engaging environment. Students will be immersed in an integrated miniaturized SCADA testbed for testing and analyzing how SCADA systems work, what they are vulnerable to, and the impact of SCADA vulnerabilities being exploited in real time.

Research Questions

1. How can a miniaturized SCADA testbed be built for a school environment using open-source architecture?
2. What critical infrastructure sectors can be easily implemented into a physical SCADA testbed?
3. Which cyber-attacks can be easily replicable in a SCADA scenario-based environment?
4. How should SCADA scenarios be modeled for an implementation into this testbed?

CHAPTER TWO

LITERATURE REVIEW

SCADA systems have become an essential part of modern society performing key functions such as ensuring essential goods and services from critical infrastructure sectors are efficiently delivered throughout the United States. They started in the 1960s due to manufacturing organizations beginning to scale out of size causing strain on workers manually tasked with controlling industrial equipment. The first generation of SCADA systems introduced was the Monolithic SCADA System, which was built on mainframe computers but did not have access to computer networks. The second generation of SCADA systems was known as distributed SCADA Systems, where SCADA systems took advantage of Local Area Network (LAN) technology, PC-based human-machine interface, and cloud technology which granted vendors the ability to optimize data transfers through vendor-specific communication protocols but prevented these devices from communicating with other vendor-specific SCADA machines. The third generation of SCADA systems was Networked SCADA Systems, where SCADA systems adopted an open system architecture and communication protocols that were not vendor specific by taking advantage of modern technologies such as Ethernet which enabled systems from other vendors to communicate with each other through the internet. This led to the latest model of SCADA systems which are Internet of Things SCADA systems. According to Jeffries (2021), “Combining SCADA systems with the cloud, IoT provides SCADA systems with an alternative to PLCs and the use of data modeling and complex algorithms” (pg.1). As SCADA systems evolved throughout each generation their accessibility to the internet grew, which left them exposed to cybersecurity attacks which have been exploited in notable cyber-attacks causing cascading effects down the supply chain.

With SCADA systems becoming integrated into the internet in the Networked SCADA Systems it opened for the potential cyber-attacks which have been exploited by adversaries to cause notable damage to countries’ infrastructures. One of those attacks was known as Stuxnet. According to Fruhlinger (2022), “Stuxnet was a powerful computer worm designed by the United States and Israeli Intelligence that targeted the centrifuges used for enriching uranium to delay or derail the Iranian Nuclear Program” (pg.1). Stuxnet worked by installing a rootkit on the targeted system at the user or root kernel level that checked to see if the computers were connected to a PLC which managed the spin rate of centrifuges used to separate the different isotopes from each other to build a nuclear bomb. If a computer was connected to a PLC related to the nuclear centrifuges, Stuxnet would begin altering the programming of the PLC which communicated to the centrifuges to spin irregularly and hide this irregularity through communicating to the computer managing the PLC that everything was fine, which made it difficult to detect and diagnose the problem before it was too late. For Stuxnet to work Fruhlinger (2022) explains that “Stuxnet exploited no less than four zero-day exploits which included a Windows

Shortcut flaw, a bug in the print spooler, two escalations of privileges, a zero-day in the Siemens PLC, and an old hole in the Cornflicker attack" (pg. 1). As a result of this attack Stuxnet set the Iranian Nuclear Program back approximately two years and was the first cyber-weapon of cyber-warfare which set the precedent for nations to target their adversary's critical infrastructure as a new form of modern warfare.

Another SCADA attack that targeted a country's critical infrastructure was the Colonial Pipeline Ransomware attack of May 2021. According to Kerner, "Colonial Pipeline is one of the largest and most vital oil pipelines in the United States that comprises of more than 5,500 miles of pipeline starting in Texas and ending in New Jersey which supplies nearly half of the fuel for the East Coast" (pg.1). During May 2021, a hacker organization known as DarkSide accessed Colonial Pipeline network through an exposed password for a virtual private network account which was discovered in a previous cybersecurity breach and stole 100GB of data within two hours, and then encrypted Colonial Pipeline computers using ransomware which affected multiple systems such as billing and accounting. As a result, Colonial Pipeline was voluntarily shut down to prevent the spread of ransomware in the organization. With Colonial Pipeline shut down gas prices skyrocketed across the East Coast due to panic buying, and airline industries were affected by the shortage of Jet Fuel, which resulted in Colonial Pipeline paying approximately $4.4M to DarkSide to get their systems unencrypted, where through the help of the United States Government were able to recover $2.4M of the ransom paid.

Lastly, another significant SCADA attack that occurred was the Ukrainian Power Grid Attack in 2015. This attack was conducted by a group of highly skilled hackers by overwriting firmware on critical devices at 16 of the substations leaving them unresponsive to any remote commands from operators. To gain access to the system the attackers started a social engineering campaign in the Spring of 2014 using spear-phishing to target IT staff and system administrators working to distribute electricity throughout Ukraine which allowed the attackers to gain access to the corporate networks but needed to pivot to the SCADA networks that controlled the grid since the network was segmented. They pivoted by conducting extensive reconnaissance mapping on the SCADA networks to gain access to the Windows Domain Controllers by harvesting user credentials before modifying the uninterruptible power supply with malicious firmware. According to Zetter, “This firmware replaced legitimate software on serial-to-ethernet converters at a dozen substations. Taking out the converters would prevent operators from sending remote commands to close the breakers once a blackout occurred” (pg.1). This cyber-attack on the Ukrainian grid resulted in power being disabled for 6 hours initially, damaging the control centers for the grid which would take months to become fully operational again, and set the precedent for the safety and security of power grids in different nations since it was the first successful cyber-attack on a country’s power grid.

With significant consequences outlined in the examples above many organizations have developed SCADA testbeds to analyze the impact of a cyber-attack on critical infrastructure and research ways to improve its security of it. The rest of this section will focus on summarizing SCADA testbeds and their scenarios, SCADA vulnerabilities, and the SCADA threat landscape to help with the development of a miniaturized SCADALAND for use and integration into CSUSB CISO Club.

Robinson (2013) describes the SCADA threat landscape by outlining the motivation of distinct types of cybersecurity threat actors such as hostile nations, foreign intelligence services, terrorist groups, and criminal groups seeking to disrupt or damage SCADA systems. He then ranks them by constructing a table ranking the level of impact of cyber-attack on a sale of 1 to 5 they can cause on a system and maps them against three different motivation types which include political, personal or war/peacekeeping, to show which type of attackers can cause the greatest potential damage to SCADA systems. “Level 1 was described as having minimal impact on the SCADA systems whereas level 5 would indicate a cyber-attack on a SCADA system that was sweeping and sustained which could result in the crippling of a nation’s critical infrastructure which brings everyday life to a halt” (pg. 3). After ranking the threat actors targeting SCADA, Robinson describes the different attack vectors present in SCADA systems and gives a blue and red team perspective on the attack vector to show how to secure it or take advantage of the vulnerability. He then ends the article by talking about how cyber-attacks can be chained together to carry out an attack to gain access to the SCADA system.

Smith (2006) takes a risk-based approach to analyze the current challenges SCADA devices face by giving a brief history of SCADA devices. Smtih discusses technical, cultural, and political challenges, and threats to SCADA before concluding on mitigation strategies for improving SCADA security. For discussing the limitation of SCADA devices, Smith outlined the limitation of SCADA over what can be installed and used on a SCADA device. Considering the fragility of these network devices and brought up some ideas on how to improve it using firewalls but emphasized it was not an end-all for improving security for SCADA devices. He then goes over the cultural issues that exist between SCADA engineers and IT personnel and then makes brief suggestions on how to bridge the gap between the two groups to improve the overall security of SCADA without compromising the availability of these critical networks. He explains how private organizations do not prioritize the security of SCADA since they focus on issues related to generating money. Leaving SCADA security up to the IT and SCADA engineers to solve the problem and make suggestions on how to improve it. Regarding the threat of SCADA, Smith outlined how SCADA devices connected to the internet were a threat and described previous SCADA attacks. “Seven key threats (attackers) to the United States critical infrastructure as outlined by the FBI, and how the attack on September 11th, 2001, showed the world the United States was weak which motivated attackers to come after our critical infrastructure" (pg. 12). He finishes the article by describing some mitigation strategies for SCADA which include cooperation between the public and private sectors and regulatory bodies setting standards relating to certain industries to better secure the nation's critical infrastructure.

Alim et.al (2021), built an open-source reproducible physical two-layer canal testbed based on SCADABR architecture. It incorporates physical sensors, and an alarm system to help maintain appropriate levels of water in the upper and lower reservoir using a predefined water setpoint level. Alim describes SCADABR architecture as, "An open-source web-based user interface utility hosted on Apache Tomcat Server, to monitor and control each of the spillway's sensors and actuators" (pg. 351). Alim then describes the five essential components of SCADA that made up his testbed. The components of the testbed include the physical layer, cyber-physical link layer, distributed control systems, SCADA network connections, and remote monitoring and control systems to show how to the testbed maintains proper water levels in the two reservoirs. For setting up attack scenarios to be used in this testbed Alim utilized Python 3 with the PyModbusTCP library to implement certain scriptable cyber-attacks such as injection attacks, reconnaissance attacks, man-in-the-middle, and DoS attacks into the physical testbed to automate and test the security of the devices in the testbed. To see how these different types of attacks affected the system, network traffic data was collected before, during, and after a cyber-attack using a data logger. The data logger collected information relating to packet size in bytes, timestamp of packet transmission, protocol overhead and efficiency, throughput, and client flow for respective client IP and aggregated this information to determine an acceptable rate at which the system operates and uses that as a baseline for comparing the traffic to when an attack occurs. The article concludes by conducting a vulnerability scan on the testbed using Nessus or OpenVAS and classifying the vulnerabilities based on criticality which ranges from critical to high, medium, and low.

Silverman et.al (2020), discusses the vulnerabilities present within SCADA systems. He explains this by providing a historical perspective on why SCADA systems are vulnerable, recent government efforts to help defend SCADA systems, analyzing current vulnerability trends in SCADA systems, and outlining vulnerabilities present within different SCADA components. To start things off Silverman explains why SCADA systems are vulnerable from a historical perspective by briefly describing executive orders President George W. Bush and Barack Obama took to improve the security of SCADA. After this, Silverman talks about the current vulnerability trends plaguing SCADA devices by referencing documentation from TrendMicro and the Global ICS and IIOT Risk Report to highlight some of the most prevalent vulnerabilities in many SCADA systems which include unpatched legacy software, lack of antivirus, weak authentication, and rogue devices Following his discussion on vulnerability trends plaguing SCADA Silverman then discusses vulnerabilities present within SCADA components such as buffer overflows in SCADA hardware, viruses and malware in SCADA software, and denial of service attacks in communication devices. He asserts the dangers of vulnerabilities present withing different SCADA devices by presenting a use case of how a cyber-attack could have been prevented on a device called Triconex 3008. He wraps up the paper by briefly listing recommendations on how to improve SCADA security in SCADA hardware, software, communication devices, and standard operating procedures.

Mazurkiewicz (2016) evaluated several types of open-source SCADA software to be implanted into a low-cost heating system that monitors the temperature, water, and power consumption of the system connected that would run on a Raspberry Pi 2. To do this, Mazurkiewicz outlined specific requirements for picking an open software application that met the requirements of the project she was working on. The requirements of the project included the ability to operate on multiple operating systems, easy deployment of the system, and the ability to integrate with the internet of Things device using an appropriate open-source SCADA software architecture which was SCADABR. Mazurkiewicz defines SCADABR as, “SCADABR is an open-source web application built in Java that offers typical SCADA capabilities such as logging, reporting, and control functionality for SCADA protocols to communicate with devices through a graphical user interface on an Apache Tomcat Server and servlets for users to define components of a SCADA testbed such as buttons, images, charts, and alarms” (pg. 2). After this Mazurkiewicz briefly discusses how the different components of SCADABR work by elaborating on the plant configurations of

SCADABR which include areas such as graphical views, logic and processing data, and software configurations.

Quieroz et.al (2009), built a SCADA testbed using OMNET++ architecture to analyze the impact of a distributed denial of service attack on a virtualized SCADA testbed containing a simplified two-tank water system. Omnet++ architecture is an “Object oriented discrete event simulation framework written in C++. It is based on concepts of modules that communicate with each other using message passing. Communication between modules occurs either through input/output gates or direct messages” (pg. 359). The two-tank water system worked by having a sensor on each of the water tanks to measure water levels. If the water level in one tank were too low, they would release water from the other tank to help balance out the other tank through a pump. When the Distributed denial of service attack was administered against the system, it flooded the RTU with TCP syn packets attempting to disrupt normal operations of the system. The result was a spike of dropped connections from users after reaching its maximum number of connections, which was 100, until the network began slowly returning to normal after the attack had happened. The impact of DDoS flooding the connections in the RTU prevented or reduced the ability of the operators to manage the appropriate water levels of the 2 tanks.

Tesfahun (2016), built a scalable virtual SCADA testbed using Common Open Research Emulator in a Linux environment to analyze the impact of a distributed denial of service attack and a false injection attack on a water-storage tank system. According to Tesfahun, “CORE architecture was developed by Boeing in their Research and Technology division and uses lightweight virtualization on Linux systems where virtual nodes are built on the CORE framework and connected to virtual networks using Linux ethernet bridging (pg. 57). Unfortunately, CORE architecture has some limitations to it such as some protocols not being built into the CORE framework so Tesfahun et.al designed a python script to integrate different SCADA protocols such as MODBUS in the emulator. When the DDoS attack which was built in Python, was used on the testbed, it flooded traffic on the RTU which disrupted communication between the MTU and RTU which resulted in the bandwidth being consumed causing a communication disruption between both devices. The authors then tested a man-in-the-middle attack using a program called free tool called Ettercap which intercepted and modified command-and-control messages between the RTU and MTU which resulted a water tank level being drained.

Farooqui (2014), discusses the basics of SCADA architecture before describing the testbed of a SCADA network based on TrueTime 2.0 beta architecture to analyze the impact of two different Denial-of-Service attacks on a SCADA testbed designed to control distinct types of processes on DC servo motors based on a given signal. TrueTime architecture can be defined as, “TrueTime beta simulator is a MATLAB/Simulink real-time simulator developed by the Department of Automatic Controls which enables the co-simulation of controller tasks, network transmissions, and continuous plant dynamics” (pg.

99). In addition, TrueTime 2.0 is compatible with different programming languages by itself except for C++ which needs a compiler to run with it and includes essential functions for the creation of simulations such as task creation code, timers, and events using the kernel block in TrueTime. This architecture was implemented into the SCADA testbed to control processes related to the DC servo based on a given signal in a round-robin scheduling scheme which represents the location of physical devices and their controllers. The testbed consisted of 8 kernel blocks representing the sensors and actuators of different DC servos, in addition to having an extra node that is known as the attacker node which is used for attacking the network with a denial-of-service attack that remains passive until initiated by the attacker. With this Farooqui tested a DoS attack on the network by sending false input signals to all the PID controllers where the controllers take this data and transfer it to the actuators connected to the servos to fix it which leads to a destabilization of the processes in the testbed. The level of impact on the servos is based on what percentage of bandwidth the DoS is using which includes .25%, 1%, and 5% with .25% representing a minor traffic disruption to the testbed and a 5% bandwidth where legitimate traffic is significantly affected, and processes lose its stability significantly beyond the 1% bandwidth threshold. The article concludes by describing the DoS bandwidth's impact at the levels mentioned above to showcase the impact on the SCADA testbed's rise and settling time and percentage overshoot to depict the destabilization of the network.

Ahmed (2016) built a multi-portable physical SCADA testbed for the University of New Orleans that models three different critical infrastructures in one testbed which includes gas, electrical power, and water waste treatment where individuals can learn about SCADA systems at different locations such as classrooms and conferences. The testbed uses a control center made up of an HMI and data historian, in addition to a remote site where the SCADA sensors, actuators, and PLCs are installed to monitor and control the physical processes of the critical infrastructures in this testbed. The authors then describe the physical components of each of the critical infrastructure sectors chosen in this testbed. before discussing the benefits of this testbed which include a demonstration of physical processes in SCADA devices, multiple different programming software support, and varied SCADA protocol support. The article ends by discussing the current limitations related to this testbed which include no fieldbus I/O support, no connectivity with the cloud, and no IoT devices implemented in the testbed before discussing other SCADA testbeds that have been built.

In conclusion, this literature review provided multiple articles that detailed the threat landscape of SCADA systems, current SCADA vulnerabilities, and different SCADA testbeds that have been built. In the articles the researchers answered my research questions by discussing what types of critical infrastructure sectors were analyzed such as the dam sector within the article “A Laboratory-Scale Canal SCADA system testbed for Cybersecurity Research”, “A

SCADA testbed for Cybersecurity and Forensic Research and Pedagogy” where they utilized a gas pipeline, power distribution and wastewater treatment plant, to answer the question of what critical infrastructure were implemented into a physical SCADA testbed. The previous research then provided me an answer to which cyber-attacks can be easily replicable in a SCADA scenario-based environment by describing the types of attacks that were used in these research papers such as man in the middle attacks, denial of service attacks, distributed denial of service, and false injection attacks and briefly describing how they affected the system, in articles like “A Laboratory-Scale Canal SCADA system for Cybersecurity Research”, “Building a SCADA Testbed, and a SCADA Testbed for investigating Cyber Security Vulnerabilities in Critical Infrastructure.”. Next, they answered my research question of how a miniaturized SCADA testbed can be built for a school environment using open-source technology by describing the open-source architecture they used in their testbed such as Omnet++ which was found in “Building a SCADA Testbed”, SCADABR and OpenPLC which was used in “A Laboratory-Scale Canal SCADA system for Cybersecurity Research'', TrueTime 2.0 in “Cyber Security Backdrop: A SCADA testbed”, and Common Open Research Emulator in “A SCADA Testbed for Investigating Cyber Security Vulnerabilities in Critical Infrastructure”. Lastly, for answering my research question of how should SCADA scenarios be modeled for implementation into the testbed the scholarly articles “A Laboratory-Scale Canal SCADA system for Cybersecurity Research” described how an automatic attack application was built to launch cyber-attacks on the testbed automatically, “Building a SCADA Testbed” briefly described how the DDoS attack would affect the SCADA system, A Scada Testbed for investigating Cyber Security Vulnerabilities in Critical Infrastructure” then describes normal operations of the testbed before detailing the effects a denial of service attack had on the PID controller of the testbed. The answers provided to me through the previous research helped me gain a generalized understanding of what goes into building a SCADA testbed. To accomplish the goal of building a requirements documentation for building a miniaturized SCADA testbed, surveys were built and distributed for SCADA professionals to interviewing three people who built SCADA testbed which will be described in the next chapter.

CHAPTER THREE

RESEARCH METHODOLOGIES

This project aims to lay the foundation for a miniaturized physical SCADA testbed to be built at CSUSB using open-source architecture to enable students to apply cybersecurity concepts they learned in class in a practical environment, in addition to being able to learn how to improve SCADA security in the future. To lay the foundation for a miniaturized physical SCADA testbed to be built at CSUSB this project will design functional, non-functional requirements, and use cases to be used as a baseline to model the physical miniaturized SCADA testbed. For constructing non-functional and functional requirements for the project, surveys were distributed to SCADA professionals related to their job roles in SCADA in addition to conducting interviews with individuals who have worked on building a SCADA testbed to answer my research questions to related to building a requirements documentation for a miniaturized SCADA testbed at CSUSB The surveys focused on testbed architecture, SCADA scenarios, and the effectiveness of the SCADA testbed for training individuals that work in SCADA. These surveys were sent to three different SCADA professionals which include SCADA individuals that built testbeds, SCADA individuals that use SCADA Testbeds, and for individuals that work on SCADA. For the interviews we conducted three different interviews with individuals who have built a physical SCADA testbed and someone who is building a virtual lab testing environment for students to learn about SCADA systems. Below will detail which methods were utilized to gather the data to answer these four questions.

1. How can a miniaturized SCADA testbed be built for a school environment using open-source architecture?

For answering this question, the research was supported by researching scholarly articles related to SCADA testbeds and conducted interviews with three different builders of SCADA testbeds. For researching the open-source architecture for SCADA testbeds, the articles utilized were “A laboratory scale Canal SCADA System testbed for cybersecurity research” where they built a physical dam-based SCADA testbed using SCADABR architecture and OpenPLC architecture, and “An open-source SCADA application in a small automation system” to gain a better understanding of the architecture outlined in the previous architecture mentioned. For conducting the interviews, the questions that were asked questions such as what type of open-source architecture they used to build their testbed? Can any parts be replaced with open-source and what devices did you use to configure your SCADA testbed?

2. What Critical Infrastructure systems can be easily implemented into the physical SCADA testbed?

For answering this question, the research utilized scholarly articles, interviews and surveys that were distributed out to builders of SCADA testbeds. For researching the critical infrastructures to implement into the SCADA testbed articles such as “Building a SCADA Security testbed” where they utilized a water pump system to be hacked into, “A laboratory scale Canal SCADA System testbed for cybersecurity research'', where they utilized a two-layer dam system to see which critical infrastructures were used in their testbeds, and “A SCADA testbed for Cybersecurity and Forensic Research and Pedagogy” where they utilized a gas pipeline, power distribution and wastewater treatment plant representing the power were. electrical and water critical infrastructure were utilized. For conducting the interviews, the questions asked in the interviews such as which critical infrastructures are included in your SCADA testbed, what is the difficulty in setting up multiple different critical infrastructure sectors in your testbed. Regarding the survey asked for, can multiple critical infrastructure sectors be included in a single testbed? What is the difficulty in integrating multiple critical infrastructure sectors, and why were these sectors chosen over other critical infrastructures?

3. Which cyber-attacks can be easily replicable in a SCADA scenario-based environment?

For answering this question, the research utilized scholarly articles, interviews with SCADA builders and survey questions that were distributed out to builders of SCADA testbeds. For researching the scholarly articles, the research utilized articles such as “A study on the vulnerabilities and threats to SCADA devices” which detailed the different types of cyber-attacks on different SCADA systems. The next one utilized was “A laboratory scale Canal SCADA System testbed for cybersecurity research” where they utilized denial of service attacks, reconnaissance attacks, man in the middle attacks, and injection attacks in their SCADA testbed. Next article was “Building a SCADA security testbed where they utilized a distributed denial of service attack to attack the water pump system. After that an article called “A SCADA testbed for investigating Cyber Security Vulnerabilities in Critical Infrastructures” was read where the authors utilized DDoS attack and a false injection attack on their SCADA testbed. When conducting the interviews, questions were added such as which cyber-attacks were used in this testbed, how are cyber-attacks configured in this testbed, and what is the difficulty with configuring different SCADA attacks? Lastly, some of the questions asked on the survey were how do these systems in the testbed react when they are manipulated by a cyber-attack, and which cyber-attacks were used in this testbed, why were these attacks preferred over other cyber-attacks, how are cyber-attacks configured in this testbed and what were the issues with setting them up, and what functional requirements are required to setup these types of attacks.

4. How should SCADA scenarios be modeled for implementation into the testbed?

For answering this question, the research utilized scholarly articles, interviews with SCADA builders and surveys that were distributed out to builders of SCADA testbeds. The research articles that were utilized to answer this question include articles such as “A laboratory scale Canal SCADA System testbed for cybersecurity research” where they configured their cyber-attacks automatically using an attack script. The next article utilized was “A SCADA Testbed for Investigating Cyber Security Vulnerabilities in critical infrastructure “which described how a DDoS and false injection attack were utilized in the testbed. The last article utilized for answering this question was “Building a SCADA Security Testbed” where they described how the cyber-attack scenario would impact the SCADA system. When conducting the interviews, the questions that were asked were how did you go about building attack and defend scenario for your SCADA testbed, and how are cyber-attacks initiated in your testbed? Lastly, some of the questions asked on the survey were what the impact of a cyber-attack was, what are the objectives of the scenario, and who was the attacker. what type of attack is used in this scenario, how are SCADA scenarios modeled and integrated into your testbed, and how do these systems in the testbed react when they are manipulated by a cyber-attack, what does a normal state of these SCADA systems look like for the sectors chosen.

With the methodology outlined on how the research decided to answer my research question, the rest of this section will detail why a qualitative approach to gathering the required data, and the benefits and disadvantages to using a survey and how interview were utilized to gather the required data. A qualitative approach was utilized to gather information about SCADA systems through a distributed survey, to gain an understanding of the thought process that went into building a SCADA testbed such as, what critical infrastructure are designed in the testbed, how cyber-attack are designed and configured, and how hard is it for a new user to learn SCADA systems. Different versions of the surveys were crafted based on their roles in SCADA. These surveys are organized into three sections including a demographic section, questions related to SCADA job role, and ends with questions relevant to everyone who works in SCADA regardless of the job role. The goal when making these surveys is to respect the participants privacy and reduce the amount of personal identifiable information to be collected. The surveys started by asking the participants' name and asking them how long they have been working in the SCADA field. for. The next set of questions was based on the type of job role the participants play in SCADA and their opinions on how SCADA testbeds should bet setup, how well do participants interact with it, the difficulty level of a newcomer learning SCADA systems, and what are the most common cyber-attacks you see on SCADA systems. The survey ends with a set of questions every SCADA individual will have regardless of role, asking about the effectiveness of SCADA testbeds, what the current trends in SCADA look like and what the future looks like for securing SCADA. For conducting the interviews, the survey questions were modified to gain a detailed understanding of what goes into crafting a SCADA testbed. The reason the survey and interview questions were organized like this is that these questions will help develop the functional and non-functional requirements, use case diagrams with actors, main flows of sample use cases, and develop the classes and relationships with for the project and gain an understanding of what the trends are in securing SCADA. Within the time constraints on this project, my faculty advisor Dr. Nestler distributed my surveys to various SCADA professionals, where they had one week to complete the survey, in addition to getting me in contact with individuals who were available to be interviewed to help me understand what goes into building a SCADA testbed.

With the methods used to conduct the research, there were some advantages and disadvantages that came with using surveys and interviews for gathering data. Some of the advantages of using an online survey to gather information are the participants can take their time with writing well-thought-out answers, are easy to distribute, and don’t require setting up individual interviews with multiple people. Some of the disadvantages of using an online survey to gather information are there are no immediate ways to follow up with an individual that took the survey, low response due to survey fatigue, and not being encouraged to provide accurate and honest answers for the survey. With the advantages and disadvantages outlined using a survey method, building the survey with appropriate questions, and getting enough responses on the surveys was a difficult task. As a result, the surveys were refined with my faculty advisor and my chair reader, to ensure that the research extracted the appropriate information needed to build the requirements documentation and limit the disadvantages of surveys. When the research was not receiving enough responses Dr. Nestler was informed, who reached out to individuals that worked on building SCADA testbeds to be interviewed so the requirements documentation could be built appropriately. The questions utilized in the interviewed were modified from the questions on the survey to extract data relevant to designing a requirements documentation for an open-sourced miniaturized SCADA testbed to be built at CSUSB.

After the data was gathered from surveys and interviews were completed, the data was aggregated by looking at the different responses and determined the frequency and similarities of results within the several types of SCADA individuals that were interviewed to see what important concepts needed to be included in the requirements documentation. Once this was completed, the research was formatted using the book, Arlow*, J., &amp; Neustadt, I. (2001). Uml 2 and the Unified Process Second Edition Practical Object-Oriented Analysis and Design* was provided to me by my chair reader to design appropriate requirement documentation such as information flows, use case diagrams, and main flows of the sample main use cases.

Utilizing surveys and interviews to gather the required information, helped enabled me to lay a foundation for an open-source miniaturized SCADA testbed to be built at CSUSB where students can apply their cybersecurity knowledge, skills, and abilities that they learn in class in a fun and engaging way. The next section will contain the requirements documentation for building a miniaturized SCADA testbed at CSUSB which includes functional and non-functional requirements, use case diagrams, use case details, classes, and main flows for sample use cases.

CHAPTER FOUR

SCADA TESTBED SYSTEM ANALYSIS AND DESIGN

After successfully gathering data from scholarly articles, survey distribution and interviews with professionals who helped build SCADA testbed the data was analyzed to see which elements would need to be incorporated to build a SCADA testbed using an Object-Oriented Analysis and Design approach. An Object-Oriented Analysis and Design approach was utilized in this paper, to construct the building blocks for the miniaturized SCADALAND testbed to be built at CSUSB. To lay the foundation for building this testbed, this paper built a requirements documentation that includes functional and non-functional requirements, detailed use cases, use case diagram, project glossary, context diagram, class diagram and why it was important to include these in the paper. The reason the functional and non-functional requirements were built was to provide the future builders of the system with an explanation of how the SCADA system should function and constraints of the SCADA testbed. As mentioned in Chapter III Methodology, the data gathered through conducting interviews with builders of SCADA testbeds, distribution of surveys, and researching scholarly articles. Next the purpose of designing the detailed use cases was to show a step-by-step sequence of events for important system functions with important users. After the system an actor semantics was developed to define the users in the requirements documentation. The reason for an actor semantics was built was to define the four main type of users for the testbed to give the builders an understanding of what their role in the testbed is. Continuing, the glossary was constructed to help future builders of this testbed understand the technical language in SCADA systems which can be found in Appendix F).

After this a use case diagram was built to showcase the high-level functions of the miniaturized SCADA testbed to be built at CSUSB. In Figure 3., there are four main users of the testbed which are database administrator, system administrator, system user and hacker showcasing an overview of what their roles are in the testbed and how the system operates at a high-level. With the diagram depicted a reader could understand that the System Administrator helps setup users of the system through active directory and permission assignment, whereas system users connect to a VPN to manipulate SCADA devices.

Following the use case diagram was the context diagram depicted in Figure 4. The context diagram was built to provide an explanation of how the overall SCADA system works. In the middle of the context diagram the SCADA and IT system is depicted with multiple external factors interacting with it. For example, the human machine interface interacts with the system by sending user modifications to the system to be applied to the physical SCADA systems and the system returning a visual representation of remote processes for users. With the context diagram future builders of the testbed will understand how the main components of building a SCADA testbed which include important devices such field devices, programmable logic controllers, and human machine interfaces interact with the system testbed to be built.

the relationship between users and the use cases constructed.

Finally, an overview class diagram was constructed to model objects in the testbed and the relationships between them which can be depicted in Figure 4. This diagram provides a visual representation of what the testbed system will look like and the relationships between different classes in the testbed. This diagram acts as a blueprint for building the SCADA system by providing a logical overview of how the system is set up. Additionally, an overview class diagram can be expanded upon to provide different attributes such as types of data, which data is publicly or privately available, and the class operations which details the jobs that each class is responsible for. Once the overview class diagram is expanded upon further, builders of the testbed can take the information presented in the class diagram and begin building the system through the use of computer programming.

Table 2. Functional and Non-Functional Requirements Documentation

ID	Details	Type	Priority
R1	The TB-SCADA System shall include logging to collect network traffic before, during, and after a cyber-attack with tools such as Wireshark	Functional Remote Monitoring and Control Systems	Musthave
R2	The TB-SCADA System shall have Attack Scenarios configured based on Ip's and port numbers	Functional Attack Application	Musthave

R3	The TB-SCADA System shall support multiple different cyber-attacks (ransomware, a man in the middle, denial of service, SQL injection etc.)	Nonfunctional Attack Application	Musthave
R4	The TB-SCADA System shall include sensors and actuators to notify when critical infrastructure sectors are compromised or not operating at appropriate level (like flashing lights)	Functional SCADA Physical System	Musthave
R5	The TB-SCADA System should include Cyber-attack scenarios and be designed and built in Python using PyModbusTCP.	Nonfunctional Attack Application	Shouldhave
R6	The TB-SCADA System could include Cyber-attacks could be configured by simple configuration buttons in a graphical user interface	Nonfunctional Attack Application	Couldhave
R7	The TB-SCADA System should include multiple different types of critical infrastructure sectors apart of it using OpenPLC and SCADABR	Nonfunctional	Shouldhave
R9	The TB-SCADA System shall include electrical wires to transport voltage from sensors and actuators to PLC	Functional Physical System	Musthave
R10	The TB-SCADA System shall have Users connect to testbed IT network through an SSH connection	Functional IT Network	Musthave
R11	The TB-SCADA System shall include at least two different types of users which include regular users (participants) and root users as the ones that deploy and design testbed	Functional Active Directory	Musthave
R12	The TB-SCADA System shall include SCADA Scenarios will need to include an attacker, objective, roles for blue and red	Functional Attack Application	Musthave

	team, and impact of an attack when it is realized		
R13	The TB-SCADA System shall be isolated from the wider internet	Functional Network Design of Testbed	Musthave
R14	The TB-SCADA System shall use SCADABR open-source architecture	Functional SCADA/ OT Network underneath IT network	Musthave
R15	The TB-SCADA System should run on Apache Tomcat Server	Functional Remote Monitoring and Control Systems	Shouldhave
R16	The TB-SCADA System should be portable and take up as minimum space as possible	Nonfunctional Physical System	Shouldhave
R17	The TB-SCADA System shall require reconfigurability of cyber-attacks and a return to normal state for critical infrastructures within the testbed	Functional Attack Application	Musthave
R18	The TB-SCADA System could include a Raspberry Pi to act as a Distributed Control System to monitor sensor data and operate connected devices automatically or manually through human participant	Functional Remote Monitoring and Control Systems	Couldhave
R19	The TB-SCADA System should be able to run on Linux, Windows or Mac computers for users participating in it	Nonfunctional Availability of System	Shouldhave
R20	The TB-SCADA System should incorporate ladder logic as a programming language for programmable logic controls	Nonfunctional Programmable Logic Controllers	Shouldhave
R21	The TB-SCADA System shall include fit in a budget of $3,044.45 to build and have money left over to replace custom parts.	Nonfunctional Financial	Musthave

R22	The TB-SCADA System shall include an IT network on top of SCADA system that will consist of router, firewall, email access and an HMI that has access to the SCADA system at a minimum	Functional Network Design of Testbed	Musthave
R23	The TB-SCADA System shall need to be on its own personal network to not interfere with school’s Wi-Fi	Nonfunctional Network Design of Testbed	Musthave
R24	The TB-SCADA System shall utilize Rasbperri Pi’s to interface with testbed sensors and pumps	Functional Sensors and Actuators in SCADA network	Musthave
R25	The TB-SCADA System shall incorporate NIST 800 controls for password strength in IT Login System	Nonfunctional IT Login System	Musthave
R26	The TB-SCADA System should be able to be scalable to add additional security layers, additional firewalls, VLans and DMZs	Nonfunctional Network Design of Testbed	Shouldhave
R27	The TB-SCADA System shall include Human Machine Interface setup in Windows	Functional Human Machine Interface	Musthave
R28	The TB-SCADA System network below the HMI in IT layer will include legacy software, firewalls, limited connectivity, and firewall	Functional SCADA network/ (OT Network)	Musthave

Detailed Use Cases

Table 3. Creating Users for IT and OT systems

Use Case: Creating Users for IT and OT systems
Use Case ID: 1
Primary Actor: System Administrator
Preconditions: System Administrator must be logged into system
Main Flow: 1. The use case begins when a system administrator sets up Active Directory service inside IT network 2. Active Directory creates users based on specified number of users given by Systems Administrator 3. Active Directory randomly generates a username in the format of last name first name and ID 4. Active Directory randomly generates a one-time password that must be changed on user's first sign on into network 5. System Administrator assigns roles to users on what they will be doing in the testbed 6. System administrator shares the login information with users of the system 7. User's login to the system using where they will change their passwords based on National Institute of Standards and Technology 800-63.
Postconditions: User accounts are successfully set up to work in the IT and OT environment with appropriate permissions to do their job and nothing else.

Table 4. User Modification of SCADA critical infrastructure setting

Use Case: User Modification of SCADA critical infrastructure setting
Use Case ID: 2
Primary actor: System User
Preconditions: User must be logged into IT system and connected to internal VPN
Flow of Events: 1. The use case begins when a user navigates to Human Machine Interface which supervises SCADA system 2. User modifies settings for appropriate critical infrastructure in HMI 3. Message is sent to Historian prepped to send message communicated in Modbus protocol down to Rasbperry Pi's acting as Programmable Logic Controllers on SCADA systems 4. Raspberry Pi acts as a Programmable Logic Controller to interpret message and applies change to SCADA device 5. SCADA device sends message back to HMI to share information on new configuration
Postconditions: 1. New change is applied to SCADA device and stats are communicated to HMI

Table 5. Removing Cameras viewing SCADA system from the public internet.

Use Case: Removing Cameras viewing SCADA system from the public internet
Use Case ID: 3
Primary actor: System User
Preconditions: User must be logged into IT systems and connected to internal VPN
Flow of Events:
1. This use case begins when a user enters HMI and navigates to cameras
2. User turns off cameras temporarily
3. User navigates to firewall
4. User modifies firewall rules to block sensitive data from being seen on public internet
5. User returns to HMI to turn cameras on again
Postconditions:
User navigates to the internet website where cameras were viewing SCADA system to verify information isnt public anymore

Table 6. Logging of Data in Human Machine Interface and Historian

Use Case: Logging of Data in Human Machine Interface and Historian
Use Case ID: 4
Primary actor: System actor
Flow of Events: 1. The use case begins when Data is generated at physical SCADA systems 2. Data is sent to Rasbperry PI's acting as the Programmable Logic Controllers 3. Raspberry Pi interprets analog data from SCADA devices and changes it to digital to be sent to HMI and Historian 4. Data is sent to HMI and Historian using Modbus Protocol 5. Data is stored and aggregated into the Historian 6. Aggregated data is displayed on HMI to provide visual representation of SCADA system status
Postconditions: System User can make appropriate changes to SCADA system based on information displayed at HMI.

Table 7. Hacker uses social engineering attack to gain user credentials

Use Case: Hacker uses social engineering attack to gain user credentials
Use Case ID: 5
Actors: Hacker, Internal User
Preconditions: none
Flow of Events: 1. The use case begins by a hacker discovers and scans network to find Ip address related to email server 2. Hacker creates social engineering email and changes IP address to bypass firewall 3. email bypasses firewall 4. email is received by internal user of IT network 5. Internal user opens link in email 6. Internal User enters credentials 7. Hacker receives credentials entered by user
Postconditions: 1. Hacker uses credentials to gain access to IT system on top of SCADA system

Table 8. What A SCADA system does when it gets hacked by an attacker

Use Case: What A SCADA system does when it gets hacked by attacker
Use Case ID: 6
Primary actor: Hacker Secondary actor: Internal User
Preconditions: An attacker has entered the SCADA network and has increased the heat level of a nuclear power plant potentially leading to a nuclear meltdown
Flow of Events: 1. The use begins when an attacker has entered the SCADA network and has increased the heat level of a nuclear power plant potentially leading to a nuclear meltdown 2. Rasbperry Pi's connected to SCADA system notice change in levels and starts blinking red and sounds an alarm to notify users of change in levels 3. Internal users are notified of the attack and navigates to HMI to see what's going on 4. User modifies settings to bring SCADA system back to normal 5. Message is sent from HMI to the Rasbperry PI to apply the changes to SCADA device 6. SCADA system applies changes sent from HMI 7. SCADA system returns to normal operating state.
Postconditions: 1. SCADA system returns to green indicating systems is operating normally and alarm has been silenced 2. Change password configurations on users that were hacked to make it harder for Hackers to get back into organization

Table 9. Conducting a Denial-of-Service attack on SCADA systems

Use Case: Conducting a Denial-of-Service attack on SCADA systems
Use Case ID: 7
Primary Actor: Hacker
Preconditions: Hackers should have access to the IT network on top of the SCADA system
Flow of Events: 1. This use case begins when hackers make their way to the HMI device 2. Hackers start creating Modbus packets which will be sent out of the Historian to the SCADA systems 3 Hacker sends Modbus packets to the Rasbperry Pi's acting as the Programmable logic controllers 4. Programmable Logic Controllers attempt to interpret all the Modbus messages coming through 5. PLC is unable to interpret all these messages and slows or blocks communication between SCADA device and HMI and Historian 6. SCADA device is unable to communicate with HMI and Historian resulting in increased chance of SCADA device being damaged or harming testbed
Postconditions: SCADA system eventually returns to normal after the Denial-of-service attack has finished.

Table 10. Ransomware attacks on IT and OT network

Use Case: Ransomware attack on IT and OT network
Use Case ID: 8
Primary Actor: Hacker
Main Flow: 1. Hacker discovers and scans network to find Ip address related to email server 2. Hacker creates Ransomware program to be embedded into a phishing email to be sent to internal network 2. Hacker creates social engineering email containing and changes IP address to bypass firewall 3. Internal user receives and opens link present in email 4. Ransomware executes and begins searching for valuable assets in the IT and OT network 5. Ransomware encrypts valuable assets in testbed such as HMI, PLC, SCADA systems, Email server, Active Directory 6. Message displays from Hacker asking organization to pay them off to unencrypt their system

Postconditions:
Organization pays hacker money to unencrypt their assets resulting in massive financial loss for the organization

Figure 1. Use Case Diagram

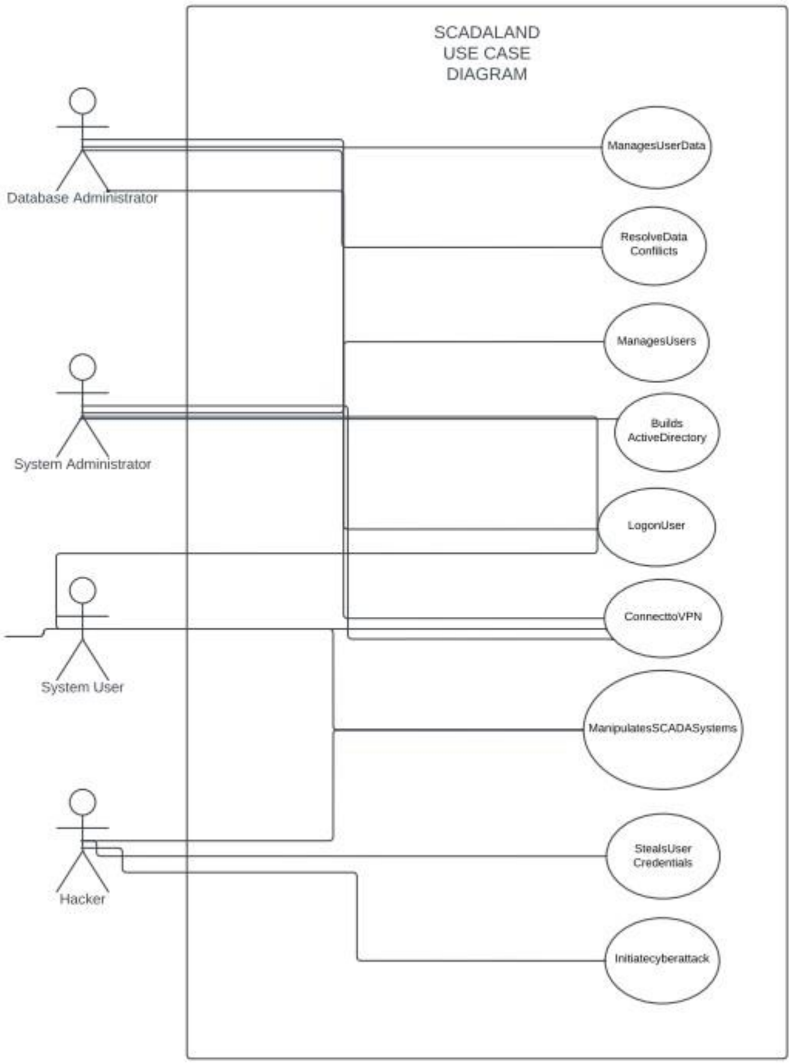


Table 12. Actor Semantics

Actor Semantics	
Actor	Brief Semantics
User	Is a regular user on the testbed that can access IT network and the SCADA network by connecting to the HMI with a VPN
System Administrator	an individual that creates and manages users by assigning them roles to users in the testbed
Database administrator	manages the confidentiality of data for IT network and resolves data conflicts
Hacker	Individual seeking to gain unauthorized access to IT and OT network

Figure 2. Context Diagram

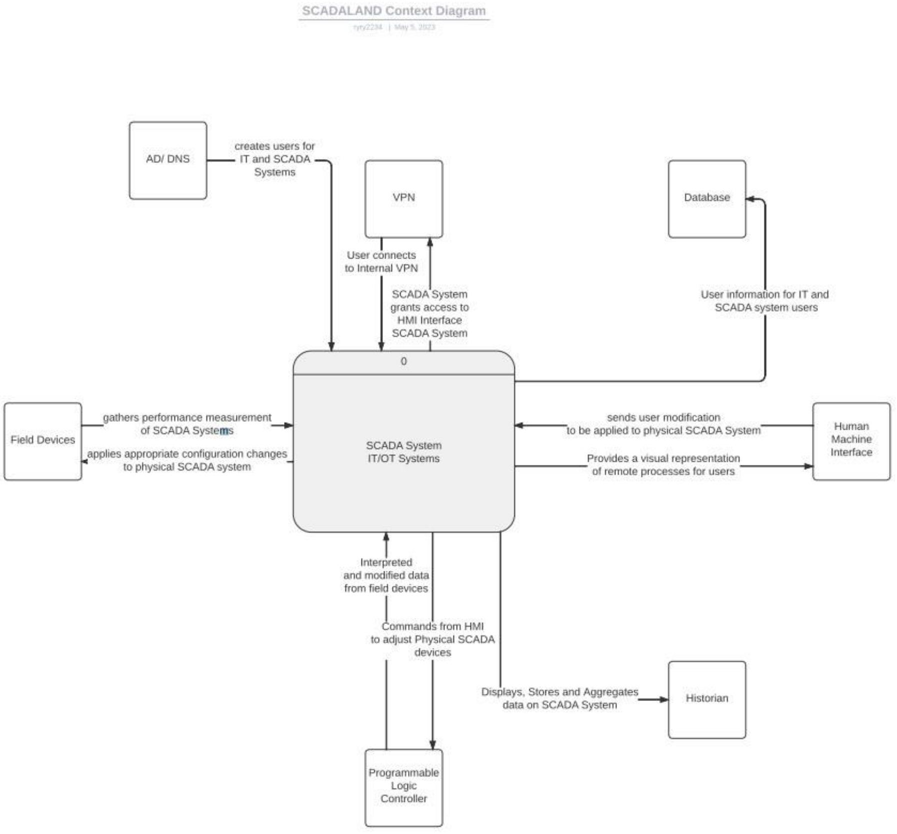
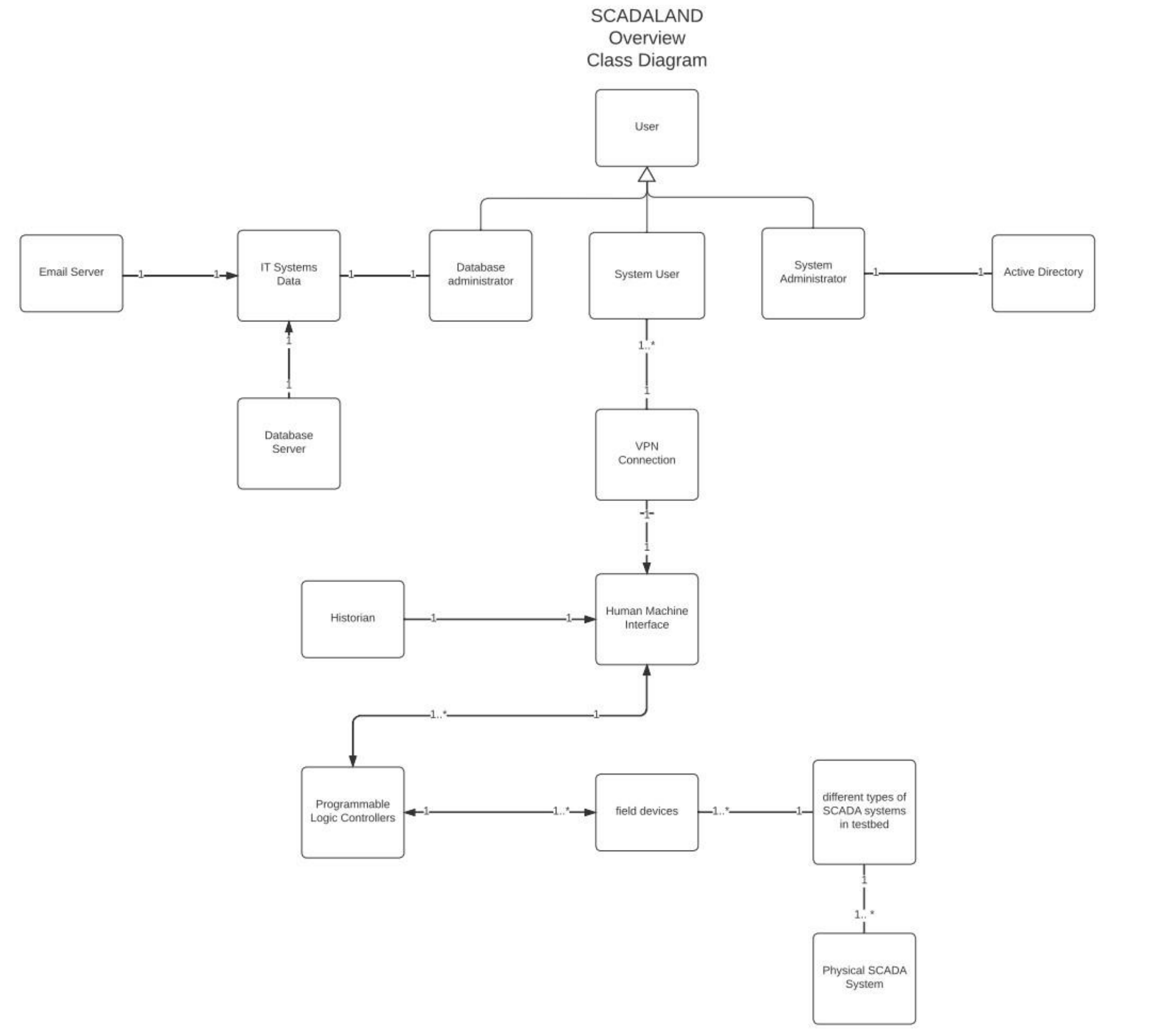

SCADALAND Context Diagram
ryry2234 | May 5, 2023
AD/ DNS
creates users for IT and SCADA Systems
VPN
User connects to Internal VPN
SCADA System grants access to HMI Interface SCADA System
Database
User information for IT and SCADA system users
0
SCADA System IT/OT Systems
Field Devices
gathers performance measurement of SCADA Systems
applies appropriate configuration changes to physical SCADA system
Human Machine Interface
sends user modification to be applied to physical SCADA System
Provides a visual representation of remote processes for users
Interpreted and modified data from field devices
Commands from HMI to adjust Physical SCADA devices
Displays, Stores and Aggregates data on SCADA System
Historian
Programmable Logic Controller

Figure 3. Overview Class Diagram



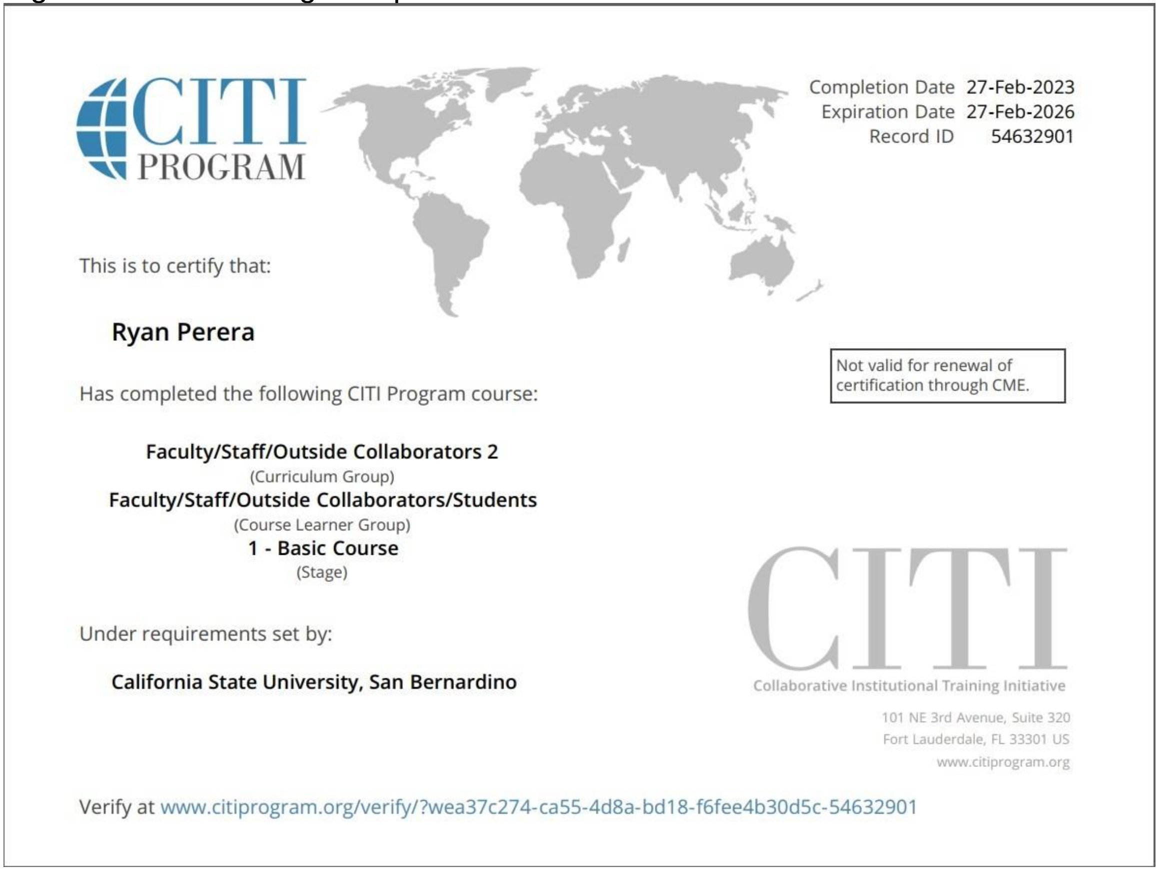
CHAPTER FIVE

CONCLUSIONS AND AREAS FOR FURTHER STUDY

This culminating experience detailed why SCADA is targeted by adversaries, why we need more people learning how to defend them, and how developing a SCADA testbed for training simulations prepares future protectors and defenders for SCADA systems. This culminating experience project answered four questions: Q1. How can a miniaturized SCADA testbed be built for a school environment using open-source architecture? Q2. What critical infrastructure sectors can be easily implemented into a physical SCADA testbed? Q3. Which cyber-attacks can be easily replicable in a SCADA scenario-based environment? Q4. How should SCADA scenarios be modeled for an implementation into this testbed? Further studies include for future studies developing a detailed SCADA System Design, purchasing, and configuring the SCADA testbed Hardware, and developing interactive training simulations for training future protectors and defenders of SCADA systems.

Additional studies will include developing a cloud-based SCADA environment where cybersecurity students in various programs in the country can learn and compete on how to protect, secure, defend and recover from SCADA breaches.

Figure 4. IRB Training Completion Form



CITI PROGRAM

Completion Date 27-Feb-2023
Expiration Date 27-Feb-2026
Record ID 54632901

This is to certify that:

Ryan Perera

Not valid for renewal of certification through CME.

Has completed the following CITI Program course:

Faculty/Staff/Outside Collaborators 2
(Curriculum Group)
Faculty/Staff/Outside Collaborators/Students
(Course Learner Group)
1 - Basic Course
(Stage)

Under requirements set by:

California State University, San Bernardino

CITI
Collaborative Institutional Training Initiative

101 NE 3rd Avenue, Suite 320
Fort Lauderdale, FL 33301 US
www.citiprogram.org

Verify at www.citiprogram.org/verify/?wea37c274-ca55-4d8a-bd18-f6fee4b30d5c-54632901

APPENDIX A

CYBER ATTACKERS MOTIVATED TO ATTACK SCADA SYSTEMS AND TYPICAL SCADA ARCHITECTURE

Hostile nations or foreign intelligence services are cyber groups that have access to a wide range of resources from their own government seeking to gain a strategic advantage in war or peacetime through disruption or destruction of essential services, such as SCADA devices. A common example of this is the Stuxnet virus designed by the United States and Israel to target Iranian centrifuges to disrupt their goal of developing nuclear weapons without starting a physical war with Iran, which set the precedent for foreign adversaries to target a nation’s critical infrastructure and delayed their development of a nuclear warhead.

The next type of attacker that would be motivated to target SCADA systems would be terrorist organizations. Terrorist organizations are groups of individuals with an ideological motivation to instill fear into nations by damaging critical assets, causing loss of life, and may avoid persecution in the country they operate out of due to unofficial support from the government in which they come from. An example of this would be where Iranian hackers hacked into a New York Dam in 2013 seeking to cause damage but were prevented due to the dam being in maintenance mode. Another type of motivated attacker against SCADA would be a criminal group.

Criminal groups are a group of attackers that are made up of amateurish to professionally skilled individuals seeking to gain notoriety or financial gain through disruption to a computer system. A common technique they use is ransomware, which encrypts an organization’s assets, and the organization must pay money to have its computer assets unlocked. This was most notably seen in 2019 with the attack on Colonial Pipeline, where a criminal organization known as DarkSide infiltrated Colonial Pipeline and encrypted their devices, thus causing a massive supply disruption in the Eastern United States.

Another type of attacker motivated against SCADA systems would be industrial spies. Industrial spies can be individuals from competing organizations or foreign adversaries looking to gain a competitive advantage over their competition by stealing company secrets or causing reputational damage through a cyber-attack that can be made up of individuals in the organization or hired by the organization. An example of this would be a Russian spy who gets hired on at a United States Missile Defense contractor and then takes that information and sends it back to his homeland so they can reverse engineer it and build their own version of it.

Next on the list of potential SCADA attackers is insider threats. An insider threat is a low or high-ranking individual that has extensive knowledge of how a company's network and computer systems work and who is motivated by revenge on a company for a perceived injustice against them. An example of this was former NSA (National Security Agency) employee Edward Snowden who leaked highly classified information about the United States, which included information about the United States government spying on its own people back in 2013.

Lastly, script kiddies are highly motivated individuals seeking to gain notoriety through their ability to break into a computer system using pre-made scripts found on the internet or through their extensive knowledge and background to prove how capable their skills are.

APPENDIX B

HOW SCADA SYSTEMS WORK

With multiple threat actors motivated to attack the United States SCADA systems, it is crucial to understand how SCADA systems work to help design a miniaturized physical SCADA testbed at CSUSB to give students the opportunity to learn about SCADA systems and how to defend them in a controlled environment. SCADA systems are made up of three layers which include control stations, communication networks, and field devices as depicted below in Figure 5.

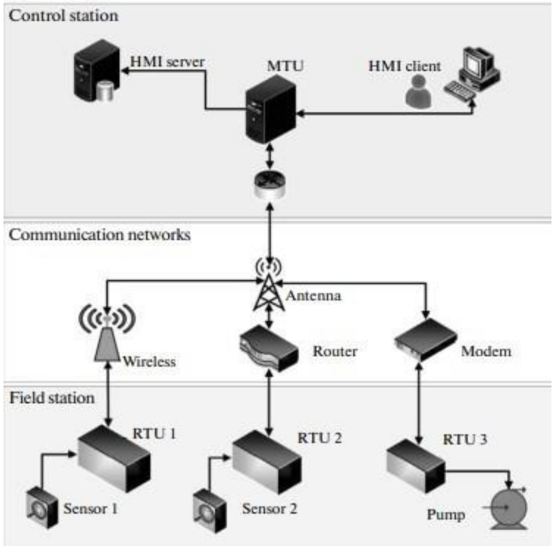


Figure 5. A typical SCADA Architecture
A SCADA Testbed for Investigating Cyber Security Vulnerabilities in Critical Infrastructures (Tesfahun 2015)

The field station layer includes physical devices such as actuators and sensors that are connected to the physical SCADA equipment in a remote site that conduct performance measurement analysis of the devices. These devices then send the information to the control system to implement appropriate actions in the form of electrical signals through the communication layer. The communication layer uses remote terminal units or programmable logic controllers to interpret messages between the field station and the control station by taking electrical signals from the physical layer and converting them to digital signals and vice versa. After this we have the control station. Tesfahun (2015) describes the control station as,

> “The control station includes a master terminal unit and human-machine interface where the MTU is the heart of SCADA systems which supervises and controls the activities of various Remote Terminal Units and sends control commands to physical processes. Then the MTU uses polling techniques to collect data from the RTUs, and based on collected data the HMI provides a visual representation of remote processes for operators to change plant configurations” (pg.2).

The MTU plays a key role in SCADA, since if it was to be damaged or manipulated by an attacker it would incorrectly interpret signals sent from the field devices and send incorrect commands, which could be the difference between someone being able to drink water provided from a water treatment facility or being poisoned from it.

APPENDIX C

VULNERABILITIES IN SCADA SYSTEMS

With a basic understanding of how SCADA systems work (Appendix 1), we start by discussing the vulnerabilities present within SCADA architecture which include hardware, software, communication, and standard operating procedures within the three layers. We will begin by talking about SCADA hardware, which are devices that communicate back to the control station such as RTUs and HMIs. These devices are vulnerable to man-in-the-middle attacks and buffer overflows. A man-in-the-middle attack is an attack where an outside attacker hijacks communication between two networks and makes sure his commands are being sent instead of legitimate traffic. Buffer overflows are attacks where attackers try to write more code onto data than it has the availability thus slowing down traffic. With these types of attacks, SCADA systems could be at risk of having traffic manipulated that gives SCADA field devices false information which can result in an attack like when an attacker tried changing the salination level of a water treatment plant in Florida in 2021.

The next type of SCADA component vulnerable to cyber-attacks is software, such as a computer program part of SCADA systems such as Microsoft Word, PowerPoint, and OneDrive. Considering SCADA devices have a long lifespan and can’t apply updates due to these systems needing to be constantly available, the number of software vulnerabilities for SCADA has increased between the period of 2015 and 2019 and peaking in 2018, as seen in figure 6 depicted below provided below.

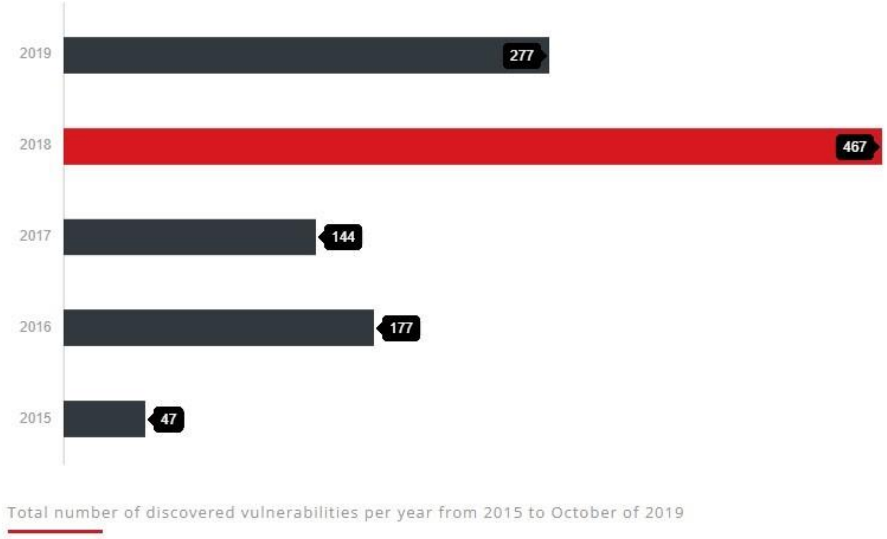


Figure 6.
TrendMicro One Flaw too Many: Vulnerabilities in SCADA Systems

According to TrendMicro, the peaking of vulnerabilities in 2018 was due to "A large portion of this count was from WebAccess and WECON's LeviStudioU, an HMI software. Delta Industrial Automation and Omron were also among the vendors that had newly discovered vulnerabilities in 2018" (pg. 1). With the growth of vulnerabilities in SCADA hardware some of the vulnerabilities present in SCADA hardware include viruses, malware, logical and authentication errors.

Next, we will talk about vulnerabilities present in SCADA communication systems. As it was mentioned before, SCADA communication networks act as an interpreter between the field station and control station to ensure SCADA devices receive information properly using RTUs. Since SCADA devices were originally designed to be in an isolated environment disconnected from the internet at large, legacy SCADA communication devices suffer from being unable to be upgraded, modernized, or patched to the latest software, which leaves them open to cyber-attacks such as denial of service (DoS) and man-in-the-middle attacks which can limit the ability of SCADA devices to communicate properly with each other. A denial-of-service attack is where an adversary floods a network with traffic coming from a single computer consuming all the computer resources resulting in reduction in efficiency or crashing.

Lastly, SCADA standard operating procedures are security mechanisms, decisions, and actions to protect against malicious attackers. According to Silverman (2020), "If SOPs are not regularly re-evaluated, they may not include the most up-to-date information for securing a SCADA environment. They may also fail to address new or deprecated devices, applications, and computer systems associated with the SCADA architecture" (pg. 4). With inadequately developed SOP organizations working with SCADA systems can find themselves at elevated risk of a cyber-attack due to weak password expectation, lack of proper encryption and authentication protocols, and a poorly developed cyber-awareness training program. If these vulnerabilities were to be exploited, they would have tremendous consequences. According to TrendMicro (2019), "The impact of an attack on SCADA systems could range from downtime, production delays, cascading effects down the supply chain, damage to equipment, and critical human safety hazards" (p. 2). Downtime can be described as any amount of time when a computer system is not functioning due to a disruption caused by threat actors, technical difficulties, or scheduled maintenance.

APPENDIX D

BRIDGING THE CULTURAL GAP BETWEEN IT AND SCADA ENGINEERS

With the wide range of SCADA vulnerabilities, it is important that there are more individuals defending SCADA to help improve the security of SCADA systems. One area that can help improve the security of SCADA systems is through bridging the cultural gap between SCADA engineers and IT professionals. According to Steven S. Smith (2006), “IT professionals focus on protecting confidentiality, integrity, and availability whereas SCADA engineers focus on ensuring the availability of systems with confidentiality and integrity at a lower priority.” (pg. 7). This thinking can be attributed to SCADA engineers understanding that SCADA systems are fragile, need to have constant availability because any disruption in a SCADA system will have cascading effects on the country, and continue to believe previously established SCADA security methods were effective enough to secure these devices, however, that idea has been disproven. With the different viewpoint on what is important for IT and SCADA individuals, it is important to bridge the gap between these two groups to improve SCADA security overall. To improve overall SCADA systems security, IT professionals need to understand that SCADA systems cannot employ all modern security control techniques a typical corporate network can employ such as firewalls, packet scanners, and computer pen testing, whereas the SCADA engineers need to understand that there are basic security options SCADA could implement to help protect these systems from attacks. Some ideas of what IT professionals can teach SCADA engineers without disrupting SCADA systems could include, changing default passwords, improving password strength, implementing a cybersecurity awareness program, investigating potential monitoring techniques for traffic, and encrypting traffic. If more IT individuals can help SCADA engineers understand these concepts, it would improve SCADA security without compromising the availability of SCADA networks.

APPENDIX E

BARRIERS TO TESTING ON LIVE SCADA SYSTEMS

For IT to improve the security of SCADA systems through cooperation with SCADA engineers, they need to be able to test the security capabilities of SCADA systems. Unfortunately, there are some barriers to testing on live SCADA systems that make it hard to teach and prepare individuals about SCADA. According to Quieroz et.al (2011), Some practical problems that limit the use of security testing and evaluation on live SCADA systems include, “Cost of scale which is where it is too expensive to implement the SCADA devices due to their size, downtime which is where SCADA systems can be slowed down or shut down due to testing causing disruptions to critical infrastructure, and risk of failure where an untested security measure will fail to protect the SCADA system” (p. 2). Since cybersecurity testing on real SCADA devices is infeasible, virtual, and physical testbeds have been built by different organizations, schools, and government research centers to learn about SCADA. Despite these testbeds being available, there are some drawbacks that make it difficult for people to learn about SCADA with currently available testbeds. According to Tesfahun (2015), “Several virtual or simulation-based SCADA testbeds have been proposed in recent years, however, most of these testbeds aren’t freely available, even those that are built with open software are sector and application-specific and lack reconfigurability of different attack scenarios” (pg. 2). In addition, another problem with virtual testbeds is that military leaders who are engaging with soldiers to train them on SCADA devices and scenarios think it has no real application with real SCADA device.

APPENDIX F

SCADA GLOSSARY

SCADA Glossary.
Glossary of Terms

Table 13.

Term	Definition
Supervisory control and data acquisition (SCADA)	SCADA is a generic name for a computerized system that gathers and processes data and apples operational controls over long distances to critical infrastructure
Monolithic SCADA system	Original SCADA systems that utilize Wide area networks (WAN's) to communicate with remote terminal units and create redundancy with the connection of a backup mainframe for the RTUs.
Distributed SCADA Systems	Second generation of SCADA systems that took advantage of local area networks where information was shared across stations in real time and increased processing power and redundancy of system
Networked SCADA Systems	Third generation of SCADA that built upon the improvements of Distributed SCADA systems by getting connected to the internet
Internet of Things SCADA Systems	Fourth generation of SCADA systems where SCADA systems are combined with Cloud providing SCADA systems with an alternative to programmable logic controllers, use of data modeling and complex algorithms
Threat Actor	Individual or group of individuals seeking to cause damage or a disruption within cyber-space
Hostile Nations / Foreign Intelligence	cyber groups that have access to a wide range of resources from their own government seeking to gain a strategic advantage in war or peacetime through disruption or destruction of essential services
Terrorist Organizations	Terrorist organizations are groups of individuals with an ideological motivation

	seeking to instill fear into nations by damaging critical assets, causing loss of life, and may avoid persecution in the country they operate out of.
Criminal Groups	a group of attackers that are made up of amateurish to professionally skilled individuals seeking to gain notoriety or financial gain through disruption to a computer system
Industrial Spies	Industrial spies can be individuals from competing organizations or foreign adversaries looking to gain a competitive advantage over their competition by stealing company secrets or causing reputational damage through a cyber-attack
Insider Threat	An insider threat is a low or high-ranking individual that has extensive knowledge of how a company's network and computer systems work and who is motivated by revenge on a company for a perceived injustice against them
Script Kiddies	highly motivated individuals seeking to gain notoriety through their ability to break into a computer system using pre-made scripts found on the internet or through their extensive knowledge to prove how capable their skills are.
Field Station Layer	layer where SCADA devices such as actuators and sensors that are setup in a remote site connected to physical SCADA devices to perform measurement analysis
Communication Layer	Acts as an interpreter between SCADA field station and control stations by converting digital signals sent from the control station to analog signals seen in field devices and vice versa
Control Station Layer	Supervises and controls the activities of various remote terminal units and sends commands to physical processes
Man-in-the-middle attack	Where an outside attacker hijacks communication between two networks

	and makes sure his commands are being sent instead of legitimate traffic
Buffer Overflows	Where an outside attacker tries to write more code onto data than it has the availability to thus slowing down traffic
Standard Operating Procedures	security mechanisms, decisions, and actions to protect against malicious attackers.
Cyber-physical Link	is the physical shell such as wires used to transmit signals between the physical system and the PLC
Programmable Logic Controller	Device within the communication layer that interprets messages between SCADA field station and control stations by converting digital signals sent from the control station to analog signals seen in field devices and vice versa
Distributed Control Systems	are devices that provide users with an interface to manipulate the physical devices of the SCADA testbed, such as the actuators and sensors
SCADABR	SCADABR is a web application written in Java and released under GNU General Public License version 3 that offers logging, reporting, and control functionality that communicates with devices using many popular SCADA protocols
Virtual Private Network	Establishing a private connection to a public network using encryption
Colonial Pipeline	One of the largest United States Oil pipelines with a length of 5500 miles distributing oil to the East Coast of the United States that runs from Texas to New Jersey
Omnet++	Object oriented discrete event simulation framework written in C++ based on concepts of modules that communicate with each other using message passing through direct/indirect gates or direct messages
CORE Architecture	A SCADA architecture that uses lightweight virtualization on Linux

	systems where virtual nodes are built on the CORE framework and connected to virtual networks using Linux ethernet bridging
Denial of Service	A cyber-attack where an adversary floods a network with traffic coming from a single computer consuming all of the computer resources resulting in reduction in efficiency or a crash
Distributed Denial of Service	cyber-attack where an adversary floods a network with traffic coming from multiple computers to consume all the computers resources resulting in reduction in efficiency or a crash
TrueTime Architecture	is a beta simulator is a MATLAB/Simulink real-time simulator developed by the Department of Automatic Controls which enables the co-simulation of controller tasks, network transmissions, and continuous plant dynamics

REFERENCES

Ahmad, I., Roussev, V., Johnson, W., Senthievel, S., & Sudhakaran, S. (2016, December 6). *A SCADA System Testbed for Cybersecurity and Forensic Research and Pedagogy*. dl.acm.org. Retrieved April 7, 2023, from https://dl.acm.org/doi/abs/10.1145/3018981.3018984?casa_token=a0xeVZEN8nsAAAAA:NJN3WjgObxyHYP93Wd1AwXU7_A9tl1SQXDhYKrwGYZ11QhCn_vb1TCTx2RqH2IyyCT-ckFc3q5s6

Alim, M. E., Wright, S. R., & Morris, T. H. (2022, April 14). *A Laboratory-Scale Canal SCADA System Testbed for Cybersecurity Research*. https://ieeexplore.ieee.org/. Retrieved April 7, 2023, from https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9750240

Arlow, J., & Neustadt, I. (2005). *Uml 2 and the Unified Process Second Edition Practical Object-Oriented Analysis and Design*. Pearson Education.

Farooqui, A. A., Zaidi, S. S. H., Memon, A. Y., & Qazi, S. (2015, January 22). *Cyber Security Backdrop: A SCADA testbed*. ieeexplore.ieee.org. Retrieved April 7, 2023, from https://ieeexplore.ieee.org/document/7017178/authors#authors

Fruhlinger, J. (2022, April 31). *Stuxnet explained: The first known cyberweapon*. csoonline.com. Retrieved April 7, 2023, from https://www.csoonline.com/article/3218104/stuxnet-explained-the-first-known-cyberweapon.html#:~:text=What%20is%20Stuxnet%3F,about%20its%20design%20and%20purpose

Jeffries, M. (2021, October 4). *Industrial Control Systems: The Four Generations of SCADA Architectures*. www.maderelectricinc.com. Retrieved April 7, 2023, from https://www.maderelectricinc.com/blog/industrial-control-systems-the-four-genertions-of-scada-architectures

Kerner, S. M. (2022, April 26). *Colonial Pipeline hack explained: Everything you need to know*. techtarget.com. Retrieved April 7, 2023, from https://www.techtarget.com/whatis/feature/Colonial-Pipeline-hack-explained-Everything-you-need-to-know

Look How Cute this Military Cyber Warfare Training Ground Is. www.wnycstudios.org. (2014, December 16). Retrieved April 7, 2023, from https://www.wnycstudios.org/podcasts/notetoself/episodes/cyber-city-military-grade-miniature-town

MAZURKIEWICZ , P. (2016, June 6). *An open source SCADA application in a small automation system*. scholar.google.com. Retrieved April 24, 2023, from https://scholar.googleusercontent.com/scholar?q=cache:z5ofk5zF42wJ:scholar.google.com/+An+open+source+SCADA+application+in+a+small+automation+system&hl=en&as_sdt=0

Miller, J. (2018, October 10). What Is SCADA and How It Increases Efficiency [web log]. Retrieved April 7, 2023, from https://www.tiga.us/blog/what-is-scada-and-how-it-increases-efficiency.

One Flaw too Many: Vulnerabilities in SCADA Systems. trendmicro.com. (2019, December 16). Retrieved April 7, 2023, from https://www.trendmicro.com/vinfo/us/security/news/vulnerabilities-and-exploits/one-flaw-too-many-vulnerabilities-in-scada-systems#:~:text=It%20should%20be%20noted%20that,DoS)%2C%20or%20steal%20information

Queiroz, C., Mahmood, A., Hu, J., Tari, Z., & Yu, X. (2009, November 10). *Building a SCADA Security Testbed*. ieeexplore.ieee.org. Retrieved April 7, 2023, from https://ieeexplore.ieee.org/document/5319283

Ryan, J. (2011, February 10). *CIA Director Leon Panetta Warns of Possible Cyber-Pearl Harbor*. abcnews.go.com. Retrieved April 7, 2023, from https://abcnews.go.com/News/cia-director-leon-panetta-warns-cyber-pearl-harbor/story?id=12888905

Robinson, M. (2013, September). *The SCADA Threat Landscape*. scienceopen.com. Retrieved April 7, 2023, from https://www.scienceopen.com/hosteddocument?doi=10.14236/ewic/ICSCSR2013.4

Silverman, D., Hu, Y.-H., & Hoppa, M. A. (2020, July 30). *A Study on Vulnerabilities and Threats to SCADA Devices*. cisse.info. Retrieved April 7, 2023, from https://cisse.info/journal/index.php/cisse/article/view/117

Smith, S. S. (n.d.). *The SCADA Security Challenge: The Race Is On*. infosecwriters.com. Retrieved April 7, 2023, from https://www.infosecwriters.com/text_resources/pdf/SSmith_SCADA.pdf

Stouffer, K., Pillitteri, V., Lightman, S., Abrams, M., & Hahn, A. (2015, May). *NIST Special Publication 800-82 Revision 2 Guide to Industrial Control Systems (ICS) Security*. nvlpubs.nist.go. Retrieved April 7, 2023, from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r2.pdf

Tesfahun, A., & Bhaskari, D. L. (2016, April 5). *A SCADA testbed for investigating cyber security vulnerabilities in critical infrastructures*. link.springer.com. Retrieved April 7, 2023, from https://link.springer.com/article/10.3103/S0146411616010090

Wreski, D. (2022, October 3). *The impact of open source on cybersecurity*. infosecinstitute.com. Retrieved April 7, 2023, from https://resources.infosecinstitute.com/topic/open-source-cybersecurity/#:~:text=Open%2Dsource%20is%20code%20that,can%20greatly%20affect%20everyone's%20security

Zetter, K. (2016, March 3). *Inside the Cunning, Unprecedented Hack of Ukraine's Power Grid*. Wired.com. Retrieved April 7, 2023, from https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/