

Spring 3-2024

## ABCD: A Risk Management Framework for SCADA Systems

Thuy Lam

Follow this and additional works at: <https://scholar.dsu.edu/theses>

---

### Recommended Citation

Lam, Thuy, "ABCD: A Risk Management Framework for SCADA Systems" (2024). *Masters Theses & Doctoral Dissertations*. 449.

<https://scholar.dsu.edu/theses/449>

This Dissertation is brought to you for free and open access by Beadle Scholar. It has been accepted for inclusion in Masters Theses & Doctoral Dissertations by an authorized administrator of Beadle Scholar. For more information, please contact [repository@dsu.edu](mailto:repository@dsu.edu).



# **ABCD: A RISK MANAGEMENT FRAMEWORK FOR SCADA SYSTEMS**

A dissertation submitted to Dakota State University in partial fulfillment of the requirements  
for the degree of

Doctor of Philosophy

in

Cyber Operations

March 2024

By

Thuy Lam

Dissertation Committee:

Dr. Cody Welu

Dr. Yong Wang

Dr. Viki Johnson



**DAKOTA STATE**  
UNIVERSITY.

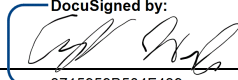
**DISSERTATION APPROVAL FORM**

This dissertation is approved as a credible and independent investigation by a candidate for the Doctor of Philosophy degree and is acceptable for meeting the dissertation requirements for this degree. Acceptance of this dissertation does not imply that the conclusions reached by the candidate are necessarily the conclusions of the major department or university.

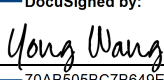
Student Name: Thuy Lam

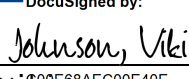
Dissertation Title:  
ABCD: A RISK MANAGEMENT FRAMEWORK FOR SCADA SYSTEMS

Graduate Office Verification:  Date: 03/27/2024  
DocuSigned by: F44C8D9E621C417...

Dissertation Chair/Co-Chair:  Date: 03/27/2024  
Print Name: welu, cody DocuSigned by: 3715959B504F439...

Dissertation Chair/Co-Chair: \_\_\_\_\_ Date: \_\_\_\_\_  
Print Name: \_\_\_\_\_

Committee Member:  Date: 03/28/2024  
Print Name: Yong wang DocuSigned by: 70AB505BC7B649E...

Committee Member:  Date: 03/27/2024  
Print Name: Johnson, viki DocuSigned by: 009E68AEC00E40E...

Committee Member: \_\_\_\_\_ Date: \_\_\_\_\_  
Print Name: \_\_\_\_\_

Committee Member: \_\_\_\_\_ Date: \_\_\_\_\_  
Print Name: \_\_\_\_\_

## ACKNOWLEDGMENT

I would want to use this occasion to extend my appreciation to all individuals who provided guidance and support along my journey in fulfilling the necessary obligations for the PhD program in Cyber operations.

I would like to express my sincere appreciation to Dr. Cody Welu, who served as my dissertation adviser. Furthermore, apart from offering insightful advice and motivation, he dedicated an extensive amount of time assisting me in the completion of my dissertation. Due to his exceptional qualities as a mentor and educator, I find it challenging to sufficiently convey my appreciation for the extensive support and guidance he has provided me. Dr. Cody exhibits a remarkable level of compassion, understanding, and helpfulness. I express gratitude for his attentive receptiveness to my academic difficulties and his adept guidance in formulating an optimal strategy to ensure my timely progression towards the successful completion of my dissertation. The individual in question fostered a sense of inclusion and provided assistance in upholding my dissertation plan, so enabling me to persist in my research and successfully finish my dissertation subsequent to the departure of my previous dissertation adviser, Dr. Joshua Stroschein. Dr. Welu has proven to be an invaluable source of guidance, and I am profoundly grateful for the wealth of information and recommendations he has provided me with. I express my gratitude for the support he provided me in achieving my objective.

I would like to express my profound thanks to Dr. Yong Wang, a valued member of my dissertation committee. Dr. Wang serves as both a member of my dissertation committee and as my advisor for the doctoral program in cyber offensive studies. On several instances, he has consistently demonstrated exceptional dedication in helping me. I would like to extend my gratitude to him for his exemplary performance as a counselor. I am indebted to him for the great guidance he has provided. He is the individual to whom I seek guidance, and I possess a strong belief that he will offer me valuable and meaningful recommendations. The individual in question demonstrates the attributes commonly associated with a conscientious mentor and professor. Due to his valuable assistance and direction, I was able to enhance my

decision-making skills during the entirety of my dissertation completion and successful attainment of a Ph.D. in the field of Cyber Operations.

I would want to express my appreciation to Dr. Viki Johnson. I initiated contact with her at a subsequent phase of my dissertation, recognizing that this timing may provide a difficulty in convincing the professor to agree to participate as a member of my dissertation committee. This is due to the fact that the academics are not only required to assist me in progressing with my dissertation, but also need to acquaint themselves with the research I have previously conducted. I am content and pleasantly taken aback by the amicability, assistance, comprehension, and readiness of Dr. Viki Johnson to acknowledge my offer. I would want to avail myself of this occasion to convey my appreciation to Dr. Johnson for her kind acceptance of my belated invitation and her unwavering commitment to assisting me in enhancing my dissertation through her invaluable counsel.

I would want to extend my appreciation to Dr. Joshua Stroschein. I had the privilege of being instructed by Dr. Joshua, my initial lecturer at Dakota State University. I received guidance from Dr. Joshua in engaging with the realm of binary and malicious software, which provided me with valuable exposure to the diverse prospects available in this industry. Prior to commencing a new professional endeavor, Dr. Stroschein fulfilled the role of my dissertation adviser. I express my gratitude for his extensive expertise, unwavering assistance, and dedicated allocation of time, all of which facilitated my first exploration of the realm of cybersecurity and the formulation of dissertation concepts.

Furthermore, I would like to express my gratitude to every one of my teachers at DSU for the essential lessons and information I have gained from their instruction. Their contributions have played a significant role in my achievement of an extra degree, the exploration of an alternate professional path, and the potential for future success. Upon reflection, it becomes apparent that my overall understanding has greatly improved as a result of the valuable knowledge imparted by the several professors who have instructed me during my time at DSU.

I would like to extend my appreciation to the Orange County Water District, as well as my supervisors, Mr. Bruce Dosier and Mrs. Vickie Nguyen, for their invaluable support, encouragement, and inspiration that they have consistently offered me. I am grateful for the trust they place in me and their readiness to provide assistance in my pursuit of my objective.

I express my gratitude to my parents for always prioritizing the significance of education and for their unwavering support in fostering my ongoing pursuit of knowledge. I would want to extend my gratitude to my family for their steadfast support during this arduous endeavor. In conclusion, I express my gratitude to all individuals, both explicitly acknowledged and those who have chosen to remain anonymous, for their invaluable assistance throughout my personal endeavor, which has significantly contributed to my progress in achieving this goal.

## ABSTRACT

Supervisory Control and Data Acquisition (SCADA) systems are used to run, monitor, and manage large-scale industrial operations. SCADA systems are frequently the target of attackers for political or financial gain due to their increasing exposure to catastrophic destruction. Historically, the overwhelming majority of SCADA networks were completely self-contained, depending on proprietary protocols and software. This has ceased to be the case. As more industrial control systems become networked, their intrinsic security becomes increasingly susceptible to attack. Despite the importance of SCADA systems and their wide adoption, their security flaws have yet to be addressed. According to SecurityScoreCard, more than three-quarters of manufacturing organizations have unpatched high-severity vulnerabilities in their systems, and nearly forty percent of these organizations, which include metals, machinery, appliances, electrical equipment, and transportation, were infected with malware in 2022 (SecurityScoreCard, 2022). Trellix's 2023 Threat Report also reported that malware attacking manufacturers accounted for 12 percent of ransomware campaigns disclosed publicly in 2022 (Trellix, 2023). SynSaber, a security firm that specializes in industrial asset and network monitoring, conducted an analysis of 926 CVEs that were included in ICS advisories from the US Cybersecurity and Infrastructure Security Agency (CISA) during the second half of 2022 and found that 35% of them had no patch or remediation available from the vendor (SynSaber, 2022). Even though compromising these vital systems could lead to catastrophic injury and operating difficulties, their security is still an open subject. In this study, we proposed a risk management framework for safeguarding SCADA systems that is based on the concept of offensive security as a means of bolstering SCADA system overall security. The research proposes a four-step methodology for managing cyber risk in SCADA systems, including assessing, blocking, capturing, and defending, which corresponds to the four primary tasks of risk management: identifying, preventing, detecting, and responding to risk. The term ABCD framework is derived from the initial letter of each of the four stages proposed by the research as well as the model used to illustrate the framework. The primary emphasis areas of the framework are multi-step attack prediction and security awareness, both of which are accomplished by predicting attack

behaviors using recommended algorithms. The model provides an intuitive and adaptable adversarial environment that enables the administrator to predict the security scenario in advance, thereby aiding in the preparation of incident response actions necessary to maintain network connectivity.



Declaration.

I hereby certify that this dissertation constitutes my own product, where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions, or writings of another.

I declare that the dissertation describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

---

Thuy Lam

# TABLE OF CONTENTS

<b>DISSERTATION APPROVAL FORM .....</b>	<b>II</b>
<b>ACKNOWLEDGMENT .....</b>	<b>III</b>
<b>ABSTRACT.....</b>	<b>VI</b>
<b>LIST OF TABLES .....</b>	<b>XII</b>
<b>LIST OF FIGURES .....</b>	<b>XIII</b>
<b>CHAPTER 1 - INTRODUCTION.....</b>	<b>1</b>
1.1    WHAT IS SCADA?.....	2
1.2    BACKGROUND OF THE PROBLEM.....	4
<i>Vulnerabilities</i> .....	4
<i>Attack Vectors</i> .....	11
<i>Attacks</i> .....	13
<i>Standards and Regulations</i> .....	14
1.3    STATEMENT OF THE PROBLEM.....	16
1.4    OBJECTIVES OF THE PROJECT .....	18
<b>CHAPTER 2 - LITERATURE REVIEW.....</b>	<b>19</b>
2.1    DESCRIPTION OF SCADA CYBER SECURITY RISK REDUCTION METHODS AND FRAMEWORKS ....	19
<i>Risk assessment in railway SCADA</i> .....	20
<i>Attack trees for evaluating SCADA vulnerabilities</i> .....	20
<i>Methodology for vulnerability evaluation in SCADA security</i> .....	20
<i>Methodology for quantitative risk reduction estimation</i> .....	21
<i>Approach to risk analysis based on scenarios in support of cyber security</i> .....	22
<i>Method based on two indices for quantifying the vulnerability of the systems</i> .....	23
<i>The risk framework for cyber-terrorism in SCADA</i> .....	23
<i>Petri net analysis is being used to assess the threat of cyberattacks on SCADA systems</i> .....	24
<i>Hierarchical, model-based risk management for critical infrastructures</i> .....	24
<i>Network security risk model (NSRM)</i> .....	25
<i>A tree of counterattack measures</i> .....	26
<i>A system security assessment based on an adversary-driven state</i> .....	27
<i>Model for evaluating the threat of cyberattacks</i> .....	27
<i>Protection against attacks on key infrastructure via computer simulations</i> .....	28

	<i>Model for risk detection and management in SCADA systems based on a graph</i> .....	28
	<i>Detection and reaction to potential dangers</i> .....	29
	<i>Nuclear power plant cyber security risk assessment</i> .....	29
	<i>Markov processes based on boolean logic</i> .....	30
	<i>A risk assessment based on CORAS for SCADA</i> .....	30
	<i>A risk assessment methodology for power control systems based on the PMU</i> .....	31
	<i>Quantitative technique for assessing SCADA cyber security risk</i> .....	31
	<i>Cyber-Attack Anticipation and Detection Kill Chain for Railway Defender</i> .....	32
	<i>Smart Grid Cyber Kill Chain-Based Hybrid Intrusion Detection System</i> .....	32
	<i>Cyber kill chain-based situational awareness framework for industrial control system</i> .....	33
	<i>Decepti-SCADA: Cyber deception architecture designed to actively safeguard infrastructures</i> ...34	
	<i>Kill chain for railway defense to foresee and detect cyberattacks</i> .....	35
2.2	FINDINGS FROM AN EVALUATION OF THE METHODOLOGY AND FRAMEWORK .....	36
<b>CHAPTER 3 - RESEARCH METHODOLOGY.....</b>		<b>38</b>
3.1	RESEARCH METHODOLOGY DESIGN .....	38
3.2	PROPOSED APPROACHES AND STEPS .....	38
3.3	NOVELTY AND JUSTIFICATIONS .....	39
3.4	ABCD'S RISK CALCULATION FRAMEWORK .....	41
3.5	CODE ORGANIZATION .....	42
<b>CHAPTER 4 – THE ABCD FRAMEWORK AND CASE STUDY .....</b>		<b>45</b>
4.1	ASSESSING PHASE.....	46
	<i>Experimental Set Up</i> .....	46
	<i>Experiment Implementations</i> .....	48
	<i>Experiment Results</i> .....	59
	<i>Discussions</i> .....	66
4.2	BLOCKING PHASE .....	67
	<i>Experimental Set Up</i> .....	68
	<i>Experiment Implementations</i> .....	73
	<i>Experiment Results</i> .....	77
	<i>Discussions</i> .....	79
4.3	CATCHING PHASE .....	80
	<i>Experimental Set Up</i> .....	80
	<i>Experiment Implementations</i> .....	92
	<i>Experiment Results</i> .....	106
	<i>Discussions</i> .....	112
4.4	DEFENDING PHASE.....	113

<i>Experimental Set Up</i> .....	113
<i>Experiment Implementations</i> .....	123
<i>Experiment Results</i> .....	129
<i>Discussions</i> .....	135
<b>CHAPTER 5 - CONCLUSIONS</b> .....	<b>137</b>
<b>REFERENCES</b> .....	<b>142</b>
<b>APPENDIX A: USERS' MANUAL</b> .....	<b>148</b>

## LIST OF TABLES

Table 1 The SCADA system's components .....	3
Table 2 Common Risk Category .....	58
Table 3 Modbus Data Types .....	82
Table 4 The format of a Modbus ASCII transmission .....	83
Table 5 The format of a Modbus RTU transmission .....	84
Table 6 The composition of Modbus TCP messages .....	85
Table 7 TCP Header Structure .....	85
Table 8 TCP Header field definitions .....	86

## LIST OF FIGURES

Figure 1 Basic of SCADA System.....	2
Figure 2 Generation of SCADA system.....	6
Figure 3 Power grid and attack entry points. ....	8
Figure 4 Illustration of Selected SCADA Attacks .....	14
Figure 5 The stages of the ABCD of the risk management framework.....	42
Figure 6 Assessing Phase Process Flow .....	49
Figure 7 Component Group Maintenance.....	54
Figure 8 Component Maintenance .....	54
Figure 9 Model Maintenance .....	55
Figure 10 Model and Component Maintenance.....	55
Figure 11 Risk Category Maintenance.....	56
Figure 12 Component - Risk Relationship Maintenance .....	56
Figure 13 Risk Score Calculation SQL Store Procedure .....	59
Figure 14 Illustration of an initial network diagram display layout.....	60
Figure 15 Less-detailed illustration of a condensed variant of the network diagram. .	61
Figure 16 Illustration of a typical first-generation SCADA system .....	62
Figure 17 Illustration of a typical second-generation SCADA system.....	63
Figure 18 Illustration of a typical third generation SCADA system.....	64
Figure 19 Illustration of a typical fourth generation Components of SCADA system.	65
Figure 20 Illustrative example of a component's associated risk score .....	66
Figure 21 Blocking Phase Process Flow .....	75
Figure 22 Illustrative example of a component's associated risk path.....	78
Figure 23 Illustrative another example of a component's associated risk path .....	78
Figure 24 Illustration of components impacted by a security breach. ....	79
Figure 25 Catching Phase Process Flow .....	93
Figure 26 Network Monitoring interface. ....	94
Figure 27 Diagram of the 68-95-99.7% Rule.....	95
Figure 28 Modbus TCP inside Ethernet Frame.....	97

Figure 29 MBAP Header and Modbus TCP/IP PDU.....	97
Figure 30 Interface to upload network traffic information. ....	99
Figure 31 Information to be captured by the system. ....	101
Figure 32 C# code to parse data. ....	102
Figure 33 PCAP data stored in a database table. ....	103
Figure 34 SQL Script to detect ARP poison. ....	104
Figure 35 Example of ARP poison output formatted as a table.....	104
Figure 36 SQL Script to detect DDoS Attack.....	105
Figure 37 Example of DDoS attack output formatted as a table.....	105
Figure 38 Test Dataset's component Structure.....	106
Figure 39 Experiment datasets. ....	107
Figure 40 Clean Dataset vs Query Flood Attack Dataset .....	108
Figure 41 Clean vs Query Flood Attack Dataset between 280 to 380 seconds .....	108
Figure 42 Clean vs TCP SYN Flood Attack Dataset.....	109
Figure 43 Clean vs TCP SYN Flood Attack Dataset between 280 and 380 seconds	110
Figure 44 Clean vs Ping Flood Attack Dataset between 280 and 380 seconds .....	110
Figure 45 Clean Dataset vs MITM Attack Dataset.....	111
Figure 46 TCP Packet Information .....	112
Figure 47 Multiple master and reverse proxy application process flow .....	121
Figure 48 Defending Phase Process Flow.....	124
Figure 49 Server and Client Configure .....	125
Figure 50 Example of Client/Server Interaction .....	126
Figure 51 Example of traffic capture .....	127
Figure 52 Illustration of Slave communication within the framework's Model.....	128
Figure 53 Illustration of Master communication within the framework's Model .....	128
Figure 54 Several clients concurrently update the server .....	130
Figure 55 Communication intercepted from many clients and servers transmission	130
Figure 56 TCP packets were captured and saved to drive .....	131
Figure 57 Colasoft software is used to stimulate the playback attack .....	132
Figure 58 The playback attack was detected, and error was sent back to the client..	133
Figure 59 The clients and servers remain operational despite the playback attack. ..	134

Figure 60 Processing module/Proxy Interface .....	135
Figure 61 An illustration of component groups .....	148
Figure 62 An illustration of component types .....	149
Figure 63 An illustration of components .....	150
Figure 64 A depiction of upload logs for a certain component.....	150
Figure 65 A demonstration of extracted PCAP data.....	151
Figure 66 A depiction the correlation between components and information .....	151
Figure 67 A depiction of the model that is contained in the database. ....	152
Figure 68 An illustration of the component's relationship in the database. ....	152
Figure 69 A depiction of the established risk categories for the experiment. ....	153
Figure 70 A depiction of the correlation between component and risk relationship .	153
Figure 71 A depiction of the view that is contained in the database.....	154
Figure 72 An illustration of the position of the component. ....	155
Figure 73 The database structure of the ABCD risk management framework. ....	155



## CHAPTER 1 - INTRODUCTION

Salesforce defines a cybersecurity risk manager as a process that identifies industry standards and regulatory guidelines for information security to reduce the likelihood that sensitive business systems will be compromised.

Distributed control systems (DCS) and Supervisory Control and Data Acquisition (SCADA) are both included under the umbrella term industrial control systems (ICS). There are numerous similarities and differences between a DCS and a SCADA in terms of functionality. DCS is typically used in large, continuous-processing facilities. DCS is a process-oriented and process-state-driven software system. It is a network-based process control system that links sensors, controllers, operator terminals, and actuators. One or more computers are often included in a DCS for control, and most communications take place across proprietary interconnections and protocols. SCADA is event-driven and data-gathering-oriented. Processes ranging from chemicals to transport can be monitored or controlled with SCADA systems. SCADA systems are more adaptable than DCS systems, which are more integrated. Due to the increased interconnectedness of SCADA systems, they are typically more complex, making it harder to detect malevolent behavior (Lamba et al., 2017). This study will concentrate on the SCADA system rather than the DCS.

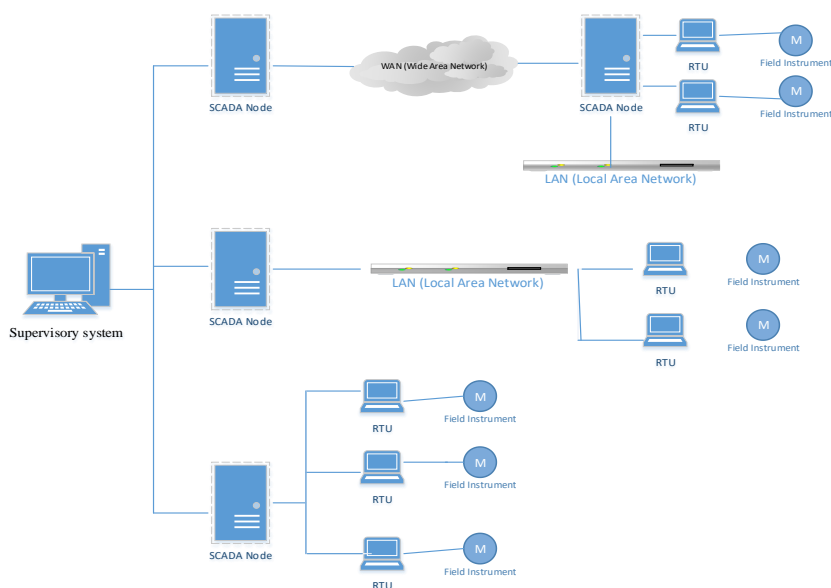
Section 1.1 provides some background information about the SCADA system. Section 1.2 of this chapter discusses the SCADA system's current state of security. The section gives an overview of the SCADA system's security, explores the security concerns connected with it, and discusses SCADA security regulations. Section 1.3 outlines the state problem for this study. Section 1.4 explains the ABCD framework and its contribution to the SCADA system's security. The section also describes how the proposed risk assessment framework and model combination would aid in the discovery of system vulnerabilities, the creation of SCADA fault exploits, and the evaluation of whether a SCADA system conforms to regulatory security standards.

## 1.1 What is SCADA?

A SCADA system is comprised of both physical hardware and software components, in addition to a communication network. SCADA encompasses a collection of control centers and various field devices, including a remote terminal unit (RTU), an intelligent electronic device (IED), and a programmable logic controller (PLC). These components are interconnected by a communication infrastructure. SCADA systems may be designed using several methodologies, encompassing both rudimentary and intricate approaches. The SCADA architecture comprises PLCs or RTUs as its essential components. PLCs and RTUs have the capability to establish communication with a diverse range of devices, including industrial equipment and human-machine interfaces (HMIs). This communication is facilitated by the utilization of SCADA software, which operates on a computer system. The SCADA software system is designed to gather, arrange, and present data in order to facilitate operators in making critical decisions. The components commonly found in SCADA design encompass local processors, operational equipment like PLCs and instruments, RTUs and IEDs, master terminal units, and a supervisory system featuring a HMI. These components are visually depicted in Figure 1, while detailed descriptions of each component can be found in Table 1.

**Figure 1**

### *Basic of SCADA System*



**Table 1***The SCADA system's components*

<b>Abbreviation</b>	<b>Interpretation</b>	<b>Description</b>
PLC	Programmable Logic Controller	A Programmable Logic Controller (PLC) is an electronic device that utilizes digital computing capabilities to monitor various sensors and execute choices according to a program written by the user. These decisions are aimed at controlling the operation of valves, solenoids, and other types of actuators.
MTU	Master Terminal Unit	The control center personnel provide directives to the remote terminal units (RTUs) in order to gather data. The system stores and evaluates data in order to furnish pertinent information to human operators, hence facilitating the process of decision-making.
HMI	Human Machine Interface	The interface is employed by operators to observe and manage the system within a control center.
RTU	Remote Terminal Unit	A Remote Terminal Unit (RTU) is responsible for acquiring data from field equipment, converting it into digital format, and transmitting it to the control center.
CI	Communication infrastructure	SCADA systems commonly engage in communication using a combination of radio and direct cable connections, employing a communication protocol that is well recognized and acknowledged within the industry.
IoT	Industrial Internet of Things	IoT devices are operational technology components such as sensors and monitors from a variety of manufacturers that may be found on or near the equipment.
SS	Supervisory System	The supervisory system enables the HMI software on control room workstations to communicate with SCADA system equipment such as RTUs, PLCs, and sensors through the supervisory system.

## 1.2 Background of the Problem

SCADA systems are indispensable in both the public and private sectors. These devices have a broad range of applications across several industries, including energy, manufacturing, oil and gas, power generation, recycling, transportation, wastewater treatment, and waste management, among other sectors. Due to the organization's and humanity's reliance on these systems, numerous threat actors view them as desirable targets. These threat actors consist of both peer and non-peer nation states and non-state actors such as terrorism, criminals, and insiders. Typically, attacks against SCADA systems are motivated by financial or political advantage.

### *Vulnerabilities*

SCADA systems support a valuable and essential mission, but their security is always a concern. Based on the OT/IoT Security Report authored by Nozomi Networks, it was observed that over the period spanning from July to December 2022, a total of 184 distinct products from 70 diverse manufacturers were impacted by Common Vulnerabilities and Exposures (CVEs) as indicated in the advisories released by the Cybersecurity and Infrastructure Security Agency (CISA). Notably, important sectors such as manufacturing, energy, water systems, healthcare, and transportation exhibited the highest degree of susceptibility to these vulnerabilities. Furthermore, it is worth noting that around 66% of vulnerabilities in Industrial Control Systems (ICS) that were reported during the latter half of 2022 were classified as 58% high severity and 13% critical severity (NozomiNetworks, 2023). There are several flaws in SCADA systems that may be exploited by attackers to get access to the systems. Some of these common weaknesses are discussed in the following sections:

#### **Vulnerabilities in the infrastructure and design.**

SCADA systems are frequently constructed in such a way that necessary security protections are omitted (Sajid et al., 2016). SCADA devices are typically designed to perform alone and without integration with other network components (Huq et al., 2017). In most circumstances, a SCADA network is assumed not to be linked to or have restricted connectivity to a corporate network. Expecting the system to survive on its own, designers seldom consider security throughout the design process. Designers are concerned about two

factors during manufacturing, which are safety and availability. As the internet became increasingly widespread, problems started to arise in these systems, diminishing the potential for isolation (Stouffer et al., 2011). While connecting to the internet is not necessarily harmful, if the default configuration and security measures are inadequate, SCADA devices are put at risk. The population's diversity is also an influence. The bulk of SCADA devices are comprised of commercial-off-the-shelf (COTS) hardware with third-party or proprietary software (Richards, 2008). Many SCADA systems are also being used in risky ways for which they were not intended. Many larger systems are composed of smaller ones, and each has its own unique problems that require their own fixes. This could lead to security and privacy settings being left out of the software development process, which could lead to vulnerabilities. Numerous COTS products include faults and vulnerabilities that may be exploited, and these flaws and vulnerabilities are publicly available over the internet. While some are attempting to resolve the issue by explaining these flaws, others disseminate them with malevolent intentions.

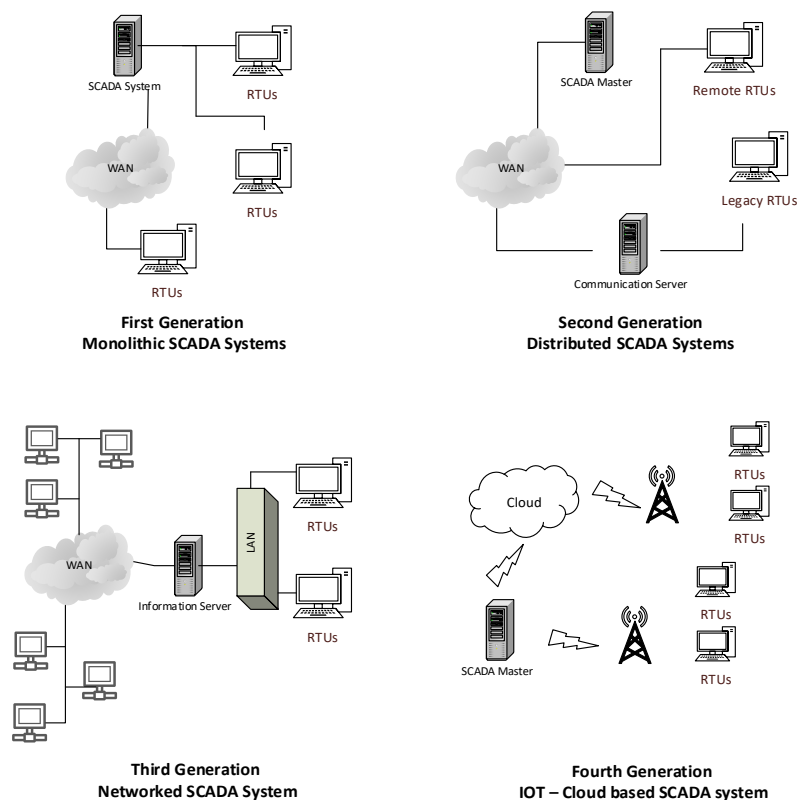
#### **Complications arise because of growth.**

SCADA systems are grouped into four generations: early SCADA systems, distributed SCADA systems, networked SCADA systems, and SCADA systems for the internet of things (IoTs), as shown in Figure 2. SCADAs were initially used to control industrial operations via monolithic systems developed prior to the widespread adoption of the Internet. These kinds of systems are no longer in use. SCADA systems of the second generation are being applied to enterprise-wide systems. The controllers, data gathering servers, control servers, and asset performance management (APM) operators are all connected via an Internet Protocol (IP) network. Third-generation SCADA architectures enable the coordination of independent process control systems located in disparate places. Multiple manufacturing locations and remote monitoring items are included in these systems. To date, third-generation SCADA was a cutting-edge SCADA system that allowed for HMI launches from mobile devices, changes to distant projects to be made on the production server and testing without shutting down the server or creating a new project file. It is the fourth-generation SCADA systems that are IoT-ready. As a result, decentralization and unification are facilitated, and execution points for algorithms can be shifted between SCADA servers and controllers. Additionally, cellular and

satellite connectivity are included, obviating the need for a virtual private network (VPN). Cloud-based SCADA servers can be accessed by controllers without requiring a fixed IP address (Ujvarosi, 2016).

**Figure 2**

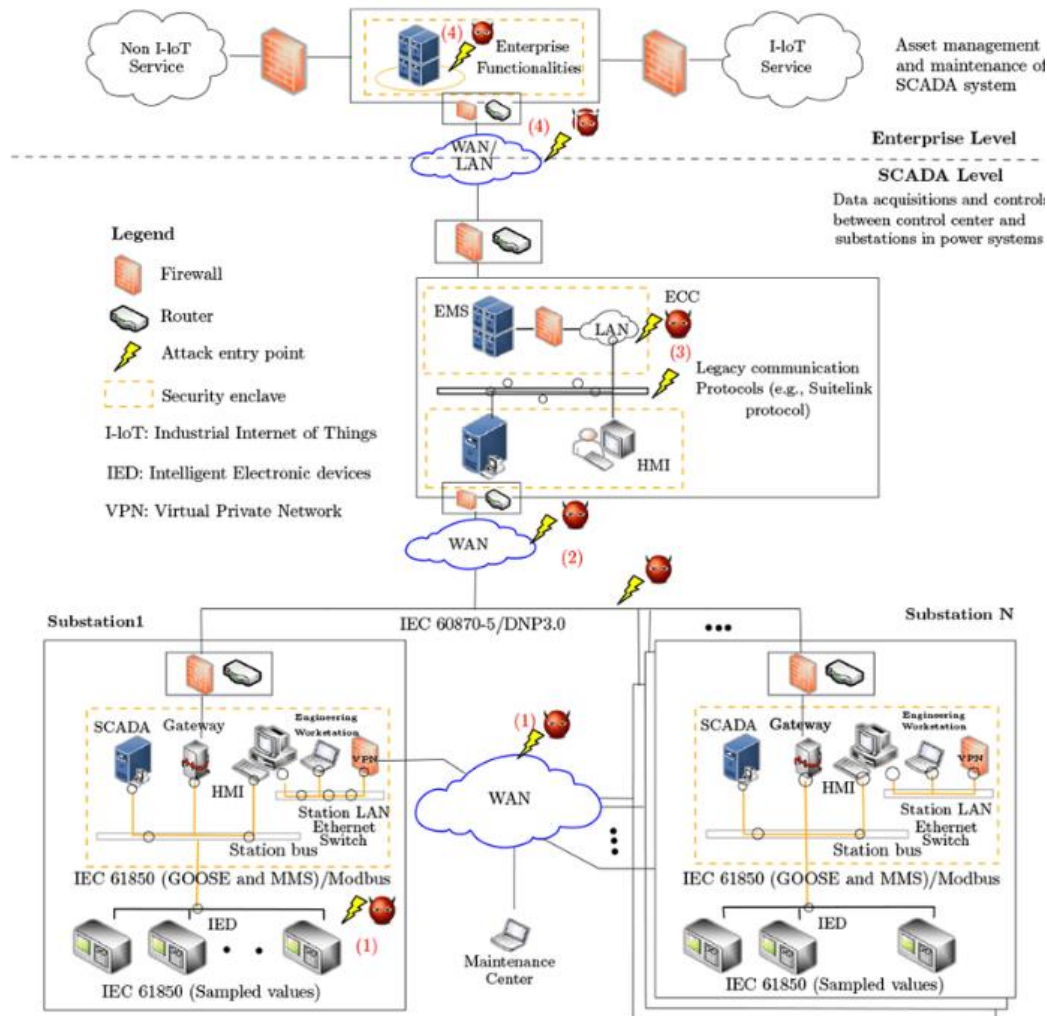
***Generation of SCADA system.***



SCADA systems of the modern era are significantly more susceptible to coordinated cyberattacks than those of the past. They benefit from the efficiency and cost-effectiveness of Internet technologies, IP-based connectivity, and operating systems in today's economy. The technologies that enable command and control in SCADA systems have resulted in a plethora of vulnerabilities and adversaries capable of exploiting them (Viega & Thompson, 2012). They are susceptible to same types of risks as other interconnected computer systems, together with the vulnerabilities inherited from the outdated platforms on which they were originally developed.

For each SCADA system, the level of vulnerability varies based on the system's individual characteristics. Since their inception, the bulk of SCADA systems have been self-contained and run on proprietary protocols and software (Anh & Chau, 2009). Over the last two decades, the architecture of such systems has evolved alongside computer technology, and current designs, while more flexible and functional, may also be more vulnerable. One of the most frequently encountered challenges that organizations face when it comes to SCADA security is integrating their SCADA and IT departments. The majority of SCADA is comprised of operational technology (OT) and information technology (IT). IT is computer hardware and peripherals including keyboards, monitors, and mice. It encompasses all systems and data, including those used to store, process, and deliver data, such as cloud computing, servers, firewalls, and antivirus software. They communicate using a variety of protocols, including HTTP, SSH, and RDP. OT refers to everything from industrial control systems to industrial process assets and everything in between. OT is a term that refers to the way that hardware and software interact with the physical components of a computer. The safety and availability of equipment and processes are OT's top priorities. The devices and PLCs owned by OT lack screens. The interaction between these entities occurs through the utilization of industrial protocols such as Modbus, Ethernet/IP, and Profinet. Due to the complexity of the network created by multiple protocol approaches, it was impossible for IT and OT security solutions to share information and provide complete visibility. Historically, OT cyber security wasn't needed because OT systems with air gaps were impenetrable. They were not vulnerable to intruders because they were not connected to the internet. When an OT system is integrated into an IT network, it opens itself up to attacks from any direction. Figure 3 depicts a potential entry site for a Power Grid SCADA system. It demonstrates that attackers can assault SCADA systems from both the IT and OT sides.

**Figure 3**  
*Power grid and attack entry points.*



(Wang et al., 2020)

SCADA and IT have not historically been integrated, but as industrial automation technology advances, the importance of this integration grows exponentially. Securing proprietary stand-alone SCADA systems solely through isolation is becoming increasingly difficult, particularly for third- and fourth-generation SCADA systems. Additionally, SCADA systems are constantly evolving, exposing them to the public and increasing their attack surface. With the advent of Industry 4.0 technology, which allows industrial automation,



dependability, and control, IoT services and applications are being expanded to industry. Due to its integration with IoT devices, SCADA is subject to the same security risks as IoT devices (Urias et al., 2012). There are several security issues that arise when large-scale IIoT networks are implemented, leading to a huge number of large-scale cyberattacks, such as fraudulent transactions or the destruction of critical infrastructure (OWASP, 2018). There are two primary factors that contribute to IoT device vulnerabilities. The manufacturer is the source of the first factor. To acquire market share, IoT device manufacturers may sacrifice security. Additionally, if the manufacturer is a failed startup, the required security patches will not be applied, leaving the IoT device vulnerable to attack. The second factor is the device's physical construction. There are risks that could compromise IoT devices, making them hazardous and vulnerable, such as obstacles posed by a large number of Internet-connected devices and device resource limitations such as battery life, memory capacity, processing power, hardware restrictions, and computing capacity. A comprehensive security study and possible countermeasures for an IIoT system have not been taken into account in recent years (Pal & Jadidi, 2021). Due to a lack of appropriate cybersecurity measures, stability and trust have become two of the most significant roadblocks to the development of IT-OT environments.

### **Outdated Systems.**

SCADA systems are well recognized for their susceptibility to many security vulnerabilities, including deficiencies in device inventory and evaluation, utilization of antiquated systems and devices with obsolete hardware and software, absence of network segmentation, and inadequate network integration. The majority of SCADA systems exhibit a high level of complexity and possess the capability to operate continuously for extended periods, often spanning many decades. Over the past decade or two, a significant number of the currently employed SCADA systems were not originally developed with a focus on cybersecurity considerations. They are built as distinct systems, and their designers prioritized repairability, dependability, and safety over security in their design. Internet intrusion detection and security networks have evolved from preparatory systems to commercial solutions based on Ethernet, TCP/IP, and Windows as computers become increasingly connected to business networks over the internet. All these technical advancements expose SCADA systems to the same hazards as traditional information technology networks. It will

be necessary to completely redesign SCADA equipment with embedded applications to meet the increased security requirements. In addition, the implementation of enhanced security protocols may result in significant periods of inactivity, a circumstance that is deemed undesirable for systems of utmost importance, such as power plants, water distribution networks, and traffic control infrastructures.

**Incorporation introduces complications.**

In the past, SCADA systems utilized several communication methods like radio, modems, or dedicated serial communication lines. Currently, there is a prevailing tendency for SCADA data to be transferred using Ethernet or IP protocols, utilizing SONET as the underlying transport mechanism. SCADA systems have the capability to accommodate a wide range of protocols, such as Modbus, Meter-Bus (M-BUS), Simple Network Management Protocol (SNMP), Distributed Network Protocol 3 (DNP3), and Building Automation Controls Network (BACnet), contingent upon the specific technological requirements. A protocol is a collection of rules that govern how SCADA systems interact with one another. SCADA systems are increasingly reliant on industry-standard computer networks and protocols to permit communication between disparate manufacturers' equipment. SCADA protocols are typically restricted to separated LANs and WANs for security reasons to avoid exposing sensitive data to the open Internet. True supervisory systems, on the other hand, do not rely simply on electrical impulses to communicate. Previously, SCADA systems relied on proprietary closed protocols; however, open standard protocols and protocol mediation are gradually becoming more popular. This protocol allows the SCADA to connect electrical devices from various manufacturers and interact with them. This implies that when these manufacturers enhance the usefulness and capabilities of their technology, they strive for optimum compatibility. Using an open standard protocol is a significant decision that leads to cost savings and more flexibility. The open standard has various advantages over the proprietary protocol, including vendor independence, open system connections, and scalability, as well as reliable goods at a low cost and widely available knowledge and specifications. DNP3, Modbus, and SNMP are the three most widely used open standards protocols. Modbus is the most extensively used SCADA protocol. It is an open-source product that is utilized by between 80% and 90% of plant components, including inverters and trackers. No authentication is necessary while interacting with the Modbus protocol, which is

based on plain text. An attacker may easily control Modbus HMIs and Modbus devices through a direct network connection.

### ***Attack Vectors***

SCADA system attacks need privileged access and in-depth understanding of the target system. Attackers can target SCADA systems in a number of ways including downloading malicious files or by replacing legal software on the system with malicious software. The objective is to discover the most efficient and extensive distribution possible to maximize the impact. Equipment and operations of the SCADA system may fail if the logic is corrupted or if hazardous settings are downloaded. Listed below are a number of vulnerabilities that SCADA components frequently face. Included are vendor support backdoors, system remote access, Internet connectivity, internal threats, and assaults on the SCADA system's software and hardware.

#### **SCADA vendor support backdoor.**

Dial-in phone ports allow system suppliers' technical support staff to remotely troubleshoot and fix software and configuration issues. Many of these access points featured a secret password and username that were only known to the vendor and anybody working for the vendor (Panguluri et al., 2017). War dialers and password cracking software are two methods through which hackers might acquire access to these ports. These ports commonly offer the most privileges to users.

#### **SCADA system remote access.**

Dial-up telephone connections to support more PCs became conceivable once PCs were established as the basis for operator workstations. In the event of an emergency at the plant or control center, an engineer can phone in from their home computer without having to make a return trip (Ravindranath, 2009). An attacker might take advantage of this flaw to get access to the platform. Using a replay attack is achievable if the dial-in communications channel is accessible, and the protocols can be circumvented. Most commercial protocol analyzers come pre-configured with this feature. Windows was a common operating system choice for remote UNIX workstations. The operator consoles are now accessible to anybody who has a Windows computer, Windows software, and a network connection.

### **Connectivity to the Internet.**

Because of the volume of data and reports generated by SCADA systems, it was inevitable that they would be integrated with business systems for automatic data interchange. A modified piece of application software may have made this possible in the fourth generations of SCADA systems. TCP/IP networking and standardized IP applications for data transmission, such as FTP or XML, would have been employed to do this. There are no limits to the number of computers that can connect to a single TCP/IP target system (Byres & Lowe, 2004). The Internet Protocol (IP) architecture is both elegant and long-lasting, cyber-attackers might use it to their advantage.

### **Internal menace.**

There are several advantages to encasing the SCADA system in a cyber-security perimeter, but the effort and cost of doing so will be enormous. A SCADA system can be disabled, harmed, and destabilized via a number of methods that necessitate the use of an insider. If a person loses trust in the government, their work, or even their relationship, they might become deviants. During a downturn in the economy, former employees and suppliers with potentially damaging information are no longer hidden. Unintentional injury and intentional harm are the two types of internal dangers (Abou el Kalam, 2021). The possibility of either of these events should be greatly reduced by effective security measures. Access control and credential verification are not going to be discussed in this article. However, password management and other internal threat protection procedures are essential.

### **Attacks on the SCADA system component.**

The acquisition of system access by an attacker through the utilization of pilfered credentials from a database is a plausible scenario (Choo, 2011). There are a number of procedures that must be followed before an attack may be launched on a user's machine. Antivirus and intrusion detection systems can be bypassed in order to perform denial-of-service attacks, exploit security weaknesses, and gain access to user accounts. Man-in-the-middle and eavesdropping assaults are part of an attack dataflow that also includes access techniques to targets. Data deletion and reading are among the most popular types of attacks on data storage. Learn New Things by Making a Deal Attacks that exploit firewall holes, such as Entrance, are commonplace. By infecting the ARP cache or exploiting a flaw, antimalware and intrusion detection systems can be evaded. Attacks against keystores enable data to be

accessed and deleted. DDoS assaults, which rapidly overload a target's resources, have compromised several networks. When the integrity of a physical zone is violated, it becomes tarnished. A router can be brought down through denial of service, compromise, and forwarding assaults. Access to a service can be restricted or denied through the use of denial-of-service, deniability-of-service, or compromise assaults (compromise). Both publicly patchable and unpatched zero-day defects are all included in this category. These weaknesses can be used to launch attacks on vulnerable products. Data mining, offline guessing, and online guessing can all be used to get unauthorized access to a user's account. Using CI, RFI, or SQL injection, an attacker can obtain access to and abuse an online application by getting through the firewall.

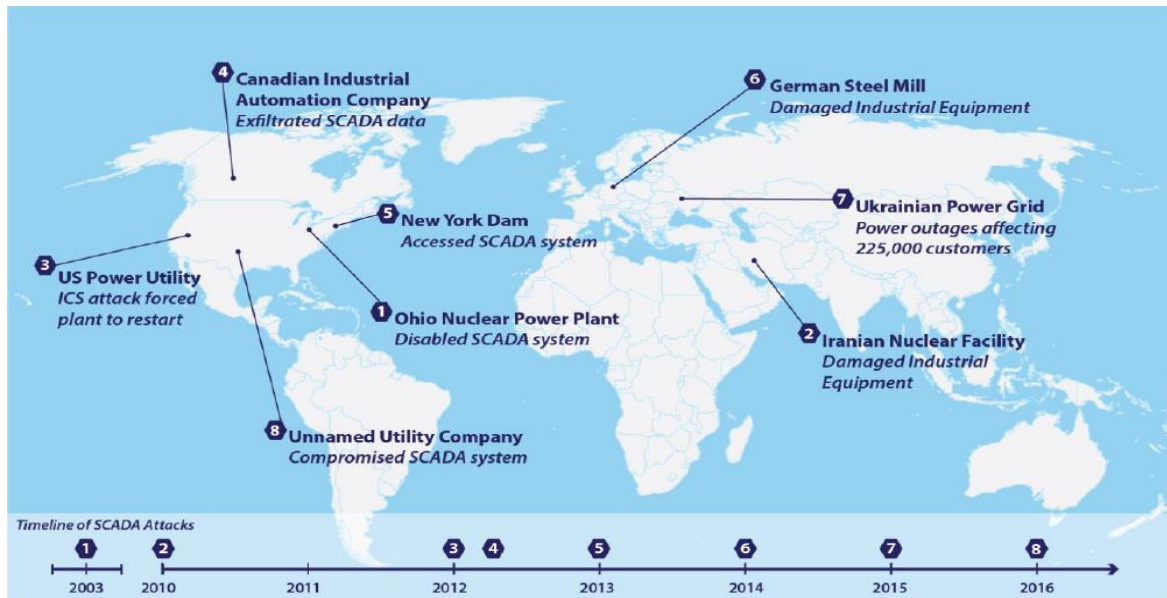
### *Attacks*

There have been various SCADA assaults recorded over the history of SCADA systems. The inaugural recorded cyberattack on critical national infrastructure (CNI) occurred in 1982, specifically targeting the Trans-Siberian pipeline. This assault resulted in a notable explosion that was discernible from outer space (Miller & Rowe, 2012). The Davies-Besse nuclear power station located in Ohio experienced a cyberattack by the Slammer worm in 2003 (Guan et al., 2011), and in Florida, a computer virus called Sobig crippled a train signaling system (Miller & Rowe, 2012). A water treatment plant in Harrisburg, Pennsylvania, was infiltrated by a hacker in 2006 (Guan et al., 2011). The Browns Ferry nuclear power station in Alabama had a manual shutdown as a result of network congestion. In 2007, the SCADA system of the Tehama Colusa Canal Authority fell victim to a cyber intrusion perpetrated by a former employee. In 2010, a significant proportion, namely one-fifth, of Iran's nuclear centrifuges experienced damage as a result of the Stuxnet computer viruses. Stuxnet's significance lies in its ability to effectively illustrate the grave implications of a cyber-attack on critical national infrastructure (CNI), therefore capturing global attention. In the year 2011, a combination of social engineering techniques, trojans, and vulnerabilities specific to the Windows operating system targeted five prominent energy and oil corporations. The Flame virus, which was first identified in 2012, has been operational for a minimum of two years throughout several regions in the Middle East and North Africa (Miller & Rowe, 2012). Additionally as seen in the Figure 4 below, the Public-Private Analytic

Exchange Program found multiple cases of SCADA attacks throughout the world (Program, 2017).

**Figure 4**

*Illustration of Selected SCADA Attacks*



*Standards and Regulations*

The below enumeration comprises a selection of laws and regulations that facilitate the cooperative efforts of public and private sectors in order to sustain and enhance critical infrastructure. There exists a diverse array of international standards that can assist SCADA owners in effectively and securely maintaining their systems. The security profile pertaining to the system. In 2004, the National Institute of Standards and Technology (NIST) released a publication titled “Industrial Control Systems” with the aim of addressing the inherent risks and objectives associated with SCADA systems. (NIST, 2004). In 2005, the Centre for the Protection of National Infrastructure (CPNI), formerly known as the National Infrastructure Security Coordination Center (NISCC), released industry-specific recommendations on best practices for the deployment of SCADA network firewalls in the United Kingdom. 21 Actions to Strengthen Cyber Security of SCADA Networks (US President's Critical Infrastructure Protection Board and the Department of Energy, 2007) outlines the necessary measures that enterprises should do to enhance the security of their SCADA networks during the aforementioned year. The Good Practice Guide (CPNI) published in 2008 by the Center for

the Protection of National Infrastructure (CPNI) outlined the recommended best practices for process control and SCADA security. According to NIST's 2008 security standards on a wide range of issues, such as technological, operational, and managerial aspects of security, a wide range of difficulties need to be addressed. In 2011, a revision to the manual was made (NIST, 2011). European SCADA patching recommendations were published in 2013 by ENISA, the European Union Agency for Network and Information Security. ENISA (2013, p. 13). The North American Electric Reliability Corporation (NERC) is making ongoing efforts to create a wide range of standards that cover the full range of CNI cyber security (NERG, 2014).

Additionally, the America's Water Infrastructure Act (AWIA) was enacted into law on October 23, 2018. Section 2013 of the AWIA mandates the creation or revision of risk assessments and emergency response plans (ERPs) for systems serving more than 3,300 people. Under the Act, water systems must certify to the EPA that they have completed risk assessments and ERPs by certain dates and that they have done so in accordance with certain criteria. It is necessary to get certifications for emergency response plans six months after risk assessment certificates have been received. Strategies and resources to increase the system's resilience, including physical security and cybersecurity, are addressed in one part of the strategy. Code Title 25, Chapter 302 of the Pennsylvania State Regulations governs the use of PLCs and SCADA systems in the water or wastewater sector. There are four parts to this section. The first portion of this paper demonstrates the utilization of a PLC or SCADA system for the purpose of monitoring, managing, and making decisions pertaining to process control activities inside a water or wastewater system. In order to adhere to federal or state legislation, as well as the accompanying rules and regulations, or the conditions and requirements outlined in permits pertaining to the functioning of water and wastewater systems, operators are required to monitor PLC or SCADA systems and possess the capability to modify or oversee the modification of said systems. The inclusion of a catastrophe recovery strategy constitutes the third factor. In the event of a failure in a PLC or SCADA system, it is imperative for the system to possess a contingency strategy to facilitate the formulation of process control determinations. The four sections of the paper address the imperative of effectively managing several SCADA systems concurrently. As per Section 302.1207, it is required that a management plan relevant to the system be established, which encompasses the operation of various treatment systems. This strategy should include a detailed description

and demonstration of the utilization of manual process control alternatives, continuous staffing, and monitoring through the SCADA system (PENNSYLVANIA, 2022).

Modernizing SCADA systems and addressing any vulnerabilities they may have, as mentioned in earlier sections, is notoriously difficult. The United States government encourages owners of SCADA systems to address or mitigate the system's vulnerabilities through the use of regulations and standards. The problem is that SCADA systems are frequently so complicated and have multiple unknown factors that they do not conform to the standards described in the section on standards, making it exceedingly difficult to keep these systems compliant with current requirements. A highly interconnected and intricate supply chain is a prime target for cybercriminals. Since 2020, open-source supply chain attacks have increased by 650% (Howard, 2021). As the system has become increasingly dispersed, compliance with the duty to meet the standard has become more challenging. There is a growing complexity in the relationship between computer communication systems and physical infrastructure as information technologies are increasingly integrated into devices and networks (Ten et al., 2008). No centralized standards or appropriate recommendations for wind turbine SCADA systems were revealed by the paper's author, Marques Cardoso, in his work *Perspectives on SCADA Data Analysis Methods for Multivariate Wind Turbine Power Curve Modeling* (Astolfi, 2021).

### **1.3 Statement of the problem**

A lack of resources allocated to cybersecurity and a lack of awareness of the technological processes required to mitigate risk are two of the most significant challenges that businesses confront. By gaining a deeper awareness of the existing vulnerabilities, businesses may address the necessary modifications to their cybersecurity strategy. Analyzing SCADA systems requires skill, knowledge, and countless hours. As of 2022, the cybersecurity industry was experiencing a workforce deficit. The cybersecurity sector lacks 3.4 million employees (Lake, 2022). In November 2021, a record-breaking 4.5 million people resigned their jobs in the United States, with historic increases in all four regions (NBCNews, 2022). In 2022, more than forty percent of employees worldwide are contemplating abandoning their jobs. To overcome this problem, security techniques must become more



imaginative, and several researchers are examining the concept of machine automation and AI security approaches.

Additionally, established techniques like vulnerability assessments and penetration testing are usually ineffectual, not the preferred solution, or have a limited scope. This is due to the fact that companies do not want to risk taking systems offline or deteriorating or destroying expensive equipment (Urias et al., 2012). Some instances in which penetration testing has gone awry are provided below. A gas utility had contracted with an IT security consulting firm to perform penetration testing on their corporate IT network. During the course of the testing, an employee of the consulting firm made a negligent foray into a section of the network that was directly connected to the SCADA system. As a result of the penetration test, the SCADA system was rendered inoperable, and the utility company was unable to pump gas through its pipes for a period of four hours. The end result was that the company was unable to provide service to its clientele for those four hours (Duggan, 2005). Another example is a ping sweep was being carried out on a PCS network in order to identify all hosts that were joined to the network for inventory reasons. As a result, a system that was managing the manufacture of integrated circuits in the fabrication facility became unresponsive as a result of the ping sweep. As a result, wafers with a value of fifty thousand dollars were ruined (Duggan, 2005). Because of the above reasons, it is usually difficult to determine the effects of a cyberattack on these SCADA systems.

SCADA operations may be disrupted, resulting in financial loss and maybe fatalities. Real-time access to all SCADA systems is vital. Penetration testing should have a zero tolerance for failure in order to avoid causing irreversible damage to the SCADA system. SCADA systems are unique for each environment. Numerous unknown and out-of-date factors exist in SCADA, despite efforts to make the system visible and identifiable to the security administrator. Due to the magnitude of the attack surface and the constant evolution of technologies, defending against IoTs-based SCADA system attacks is an ongoing challenge. All of these factors require an efficient risk management framework for assessing SCADA security risk with minimal risks to system performance, managing unknown SCADA system elements, and proposing a method for protecting the SCADA system with minimal system modifications.

#### **1.4 Objectives of the project**

Networks, like any other technology, are susceptible to security threats. Businesses must be prepared to cope with attacks from all directions due to the continuously changing world of malware and security solutions. Several researchers, as described in the following section, have proposed a number of methods for addressing the problem; however, these methods are incapable of addressing the problem's complexity due to the ever-changing nature of Scada system components and the unknown security level of many of the system components. This paper proposes a risk assessment framework that is based on the analysis of known events to detect future attacks. This serves as a line of defense against unidentified assaults that current technology is unable to detect. The ABCD methodology for this study is intended to assist network administrators in identifying opportunities for improvement and providing a holistic view of any existing network's current state. The framework layers work together, first to stop a cyber threat, then to lessen the damage if the threat comes true, and finally to protect against a specific threat. As a result, network administrators may make more informed and prudent business decisions that directly affect the system's performance. With the ABCD framework, security professionals may dynamically design their model to replicate their real-world system while performing network connection evaluation internally, certain that their actions will have no effect on the system's health. The ABCD framework is a proof-of-concept network risk management tool that offers visibility, risk measurement, and risk detection to help mitigate risk, reduce risk, and offer viable solutions to assist networking design.

## CHAPTER 2 - LITERATURE REVIEW

The rising dependence on interconnected physical and cyber-based control systems for critical infrastructure and industrial automation has increased the threat to SCADA systems from cyber security. Identification of risks is crucial for building a complete, realistic, and long-term SCADA security policy. Various approaches to analyzing SCADA system security concerns are discussed in the sections of chapter 2, the literature review. The initial collection of papers was compiled based on searches conducted on IEEE Xplore and ACM. Based on an evaluation of the titles, abstracts, and full texts of the generated articles, those pertinent to the issue of security and risk in SCADA assessment techniques were selected for manual review. This study exclusively included articles that presented innovative approaches for risk management, specifically designed for or applied to SCADA or ICS systems. The evaluation of the methodologies was conducted using the following criteria: objectivity, application domain, coverage of risk management principles, measurement of effect, data sources for probability computation, assessment methodology, and availability of tool support. The scope of this study was restricted to literature that primarily focuses on technical aspects of cyber security. Consequently, studies that solely address risk assessment from perspectives related to safety, financial considerations, or dependability were excluded from consideration.

### **2.1 Description of SCADA cyber security risk reduction methods and frameworks**

This study exclusively included articles that presented innovative approaches for risk management, specifically designed for or applied to SCADA or ICS systems. The evaluation of the methodologies was conducted using the following criteria: objectivity, application domain, coverage of risk management principles, measurement of effect, data sources for probability computation, assessment methodology, and availability of tool support. The scope of this study was restricted to literature that primarily focuses on technical aspects of cyber security. Consequently, studies that solely address risk assessment from perspectives related to safety, financial considerations, or dependability were excluded from consideration.

### ***Risk assessment in railway SCADA***

Chittester and Haimes offer a risk assessment approach for GPS-based railway SCADA systems that uses the Hierarchical Holographic Modelling (HMM) (2004). It is the approach of capturing and expressing the essence of the system's underlying various traits and attributes. The HMM was used to represent complicated military and civilian systems. – HMM is important in the context of SCADA since it helps identify hazards in subsystems and their impact on the system as a whole. Hierarchical holographic models of SCADA systems have three sub-models: hardware and software, human supervisory, and environment. There are several levels of subdivision within each of these sub-models. To make risk identification easier, the framework advises that the Control Objectives for Information and Related Technology (CobiT) be mapped into the holographic model (Chittester & Haimes, 2004).

### ***Attack trees for evaluating SCADA vulnerabilities***

Analyzing vulnerabilities in SCADA systems that employ the MODBUS and MODBUS/TCP communication protocols using attack trees. An attack tree offers a systematic perspective of the steps leading up to an attack and helps in determining suitable security solutions. Byres et al. state that risk is contingent upon factors such as system architecture, under certain circumstances, mitigation strategies, offensive difficulty, discovery probability, and offensive cost. The Byres review tries to determine the attributes of the most essential attack event and suggest effective strategies to achieve the ultimate purpose of the assault. Prior to undertaking this task, a group of experts from the sector must initially ascertain the probable goals of an aggressor's intentions and construct an attack tree with these goals serving as its nodes. Every individual leaf of an attack tree is subsequently assigned a technical complexity level, which is determined based on the categories of easy, moderate, difficult, and unlikely. The difficulty of each node with child nodes is calculated by taking the maximum value of the child nodes using the AND operator and the minimum value using the OR operator. The level of difficulty may vary at regular intervals (Byres et al., 2004).

### ***Methodology for vulnerability evaluation in SCADA security***

The cyber vulnerability assessment methodology for SCADA systems, as devised by Permann and Rohde, draws upon their extensive experience in evaluating the security of

numerous SCADA systems. Their participation in the US Department of Energy's Office of Electricity, Energy Assurance, and the Idaho National Laboratory SCADA Test Bed program, which sponsors the national SCADA Test Bed program, allowed them to gain this proficiency. There are five distinct phases in the method that Permann and Rohde propose. First phase, assessment plan development: a plan specifies the assessment's budget, timeline, objectives, resources, expert involvement, and expected deliverables. The setting of the testing environment must be secure and non-production-like as in the second phase. Vulnerability assessment is the third phase. The vulnerability assessment is performed using an external penetration test on the system being evaluated. A variety of free-source and commercial vulnerability assessment tools are provided. The fourth phase, reporting, requires detailed documentation of the evaluation and testing methods and findings. The fifth phase pertains to measurements and scoring. The quantification of SCADA system security is necessary to enable comparative analysis with other systems (Permann & Rohde, 2005).

#### ***Methodology for quantitative risk reduction estimation***

Reducing cyber risk in SCADA systems may be quantified using the technique proposed by McQueen and co-authors in their paper. The difference in time-to-compromise between baseline and enhanced systems is evaluated and analyzed in order to estimate the risk reduction of a cyber-attack. Ten steps make up the approach. The first step is to set up the computer's settings. The second step is to determine whether parts of the quantitative risk model apply to the situation. The third step is to determine and prioritize the principal target's security requirements. The fourth step is to look for weaknesses in the system. The fifth step is to categorize each device's vulnerabilities based on the kind of attack. Lastly, figure out how long it will take to reach a compromise for each gadget. A compromise graph and attack pathways are generated in the seventh step. Calculate the most likely assault path will happen in eighth step. Perform steps three to eight both for the baseline and the upgraded version is the step ninth. Using the findings from both systems, assess the amount of risk that has been minimized in step tenth. McQueen et al. present a method for assessing the likelihood of an unwanted event occurring. If the system is included in the target list of an attacker, the likelihood of being selected as a target, the probability of a breach in the perimeter, the probability of a successful attack, and the probability of resulting damage may be calculated

by summing the respective conditional probabilities. When the capacity to evaluate all probabilities is lacking, the assessment of risk reduction focuses on the alteration in the likelihood of a perimeter breach and a successful attack. SCADA security requirements prioritize integrity and availability above secrecy, with the latter taking a back seat. An existing vulnerability library is used to identify a system's vulnerabilities. Recon, breach, penetration, or damage are all examples of vulnerabilities that can be exploited by attackers. A device's time-to-compromise is determined. It is contingent on the target system's known vulnerabilities and the attacker's capability. McQueen et al. provide an inference-based analysis of the strategies for calculating time-to-compromise. Compromise graphs are constructed for both the baseline and upgraded SCADA systems. The significant attack pathways are then identified as the ones that exhibit the shortest time required to compromise the target system. Ultimately, the duration required to breach both the initial system and the enhanced version is assessed. The assessment of system security and risk is predominantly conducted through the utilization of the time-to-compromise approach. With the suggested method, small-scale SCADA systems are used to test how well security countermeasures work (McQueen et al., 2006).

### ***Approach to risk analysis based on scenarios in support of cyber security***

The Control Systems Security Center, on behalf of the National Cyber Security Division of the Department of Homeland Security, conducts the assessment of cyber risk using a scenario-based approach. It is important to consider process flow diagrams that outline the essential elements, infrastructure, and networks. Additionally, the underlying safety analysis and operational history, as well as threat and vulnerability data, should be examined. Attack pathways and critical human-system reactions should also be evaluated, along with probability and quantified consequent damage state. The model of the examined system was developed by experts in industrial processes and security requirements. Operational and cyber professionals worked together to examine the potential for vulnerabilities, threats, and the expected human–system reaction. Using the Delphi method, the opinions of specialists were gathered. The present study involved the delineation of attacker capabilities and probable system consequences inside a designated cyber-attack

scenario targeting a nuclear power facility, as part of the methodology including attack variations based on scenarios (Gertman et al., 2006).

### ***Method based on two indices for quantifying the vulnerability of the systems***

Patel et al. provide an additional way of qualitatively assessing a SCADA system's susceptibility. System administrators can use this strategy to make better-informed judgments on the implementation of security countermeasures. Vulnerability trees enhanced with two risk-impact and cyber-vulnerability indicators are used in the technique. A higher threat-impact index suggests a greater financial impact of a cyber attack. Indicators of a system's vulnerability to cyberattacks are measured using the cyber-vulnerability index (CVI). The higher the index, the more susceptible the system is. Both indices are based on a 0–100 scale. There are a total of six steps involved in this process. The first step is to create a vulnerability tree from the ground up for the original system. Creating a threat-impact index and populating an effect analysis database are the next steps. The threat-impact index values are added to the tree in Step 3. Finally, a cyber-vulnerability index value is calculated. The fifth step is the addition of cyber-vulnerability index values to the tree. Using a security-enhanced system, we repeat steps 2–5 and compare the results. According to Patel et al., a vulnerability tree was created based on the research of previous assaults. Attack-related financial losses were assessed through interviews with engineers, managers, operators, and accountants. Historical data was used to estimate the assault probability. At the University of Louisville, the technique was tested on a SCADA system (Patel et al., 2008).

### ***The risk framework for cyber-terrorism in SCADA***

A panel of five SCADA industry specialists, including Beggs and Warren, has verified a risk framework for cyber-terrorism SCADA. The framework is divided into three sections: risk assessment, a competency model, and controls. It is recommended that the Australian risk management standard AS/NZS 4360:2004 be adapted for SCADA systems. The indicators of the level of capabilities within a cyber-terrorist group include advanced information and communication technology (ICT) skills, utilization of sophisticated hacking tools and techniques, access to cutting-edge ICTs, extensive knowledge of SCADA systems, presence of insiders within the targeted organization, engagement in reconnaissance activities,

availability of funding, and motivation for the development of a cyber-terrorism capability assessment model. Information security management standards AS/NZS 27002:2006 are utilized at the control stage, which are customized to the SCADA context and list eleven security control clauses for the SCADA environment. Key areas in corporate security encompass security policies, the surroundings, physical and operational safety, the safety of information, the management of assets and operations, and privileged access control (Beggs & Warren, 2009).

### ***Petri net analysis is being used to assess the threat of cyberattacks on SCADA systems***

Henry et al. present an approach for estimating the threat of cyber assaults on SCADA systems' computer network operations. This technique utilizes Petri Net condition coverability assessment along with simulation. Identifying all high-risk attack situations is the primary goal of the strategy. Instead of relying on a metric like probability, which can be difficult to accurately assess in many real-world scenarios, the technique considers risk in terms of the resources an attacker might potentially get access to during an assault. In Balasubramanian et al., the approach is proven using a non-automated hazardous liquid loading procedure. The analysis process begins by identifying potential modes of process failure and their associated effects, followed by separating those modes of failure that may result in a process failure. The attacker's tools and equipment are then identified. A breakdown in the SCADA system might potentially result in one or more process failures due to unauthorized access by an attacker to system resources. Process owners have the ability to utilize a metric in order to assess and quantify the effects of their actions. Such metrics are the negative impacts of reduced production throughput and environmental pollution. According to the paper's example, the intensity of impact is quantified in terms of how many people are injured as a result of the operation. Henry et al. present two metrics to assess risk: the median of all potential failure pathways in SCADA and processes, and the most extreme risk measure, which represents the highest number of all potential failure modes (Henry et al., 2009).

### ***Hierarchical, model-based risk management for critical infrastructures***

According to Baiardi and colleagues, an infrastructure's security dependencies may be seen using labeled hypergraphs, which are both hierarchical and labelled. Infrastructural



hypergraphs represent the interconnected parts of a system and show their internal states and interactions. The sequence of actions in an elementary attack is presented in an evolution graph, which is a directed acyclic graph, arranged in a specific order with the ultimate objective in mind. Successive iterations outline novel tactics for launching attacks. The evolution graph has been truncated to exclude evolutions that have a low likelihood for subsequent examination. The efficacy or inefficacy of an attack plan is contingent upon historical facts pertaining to the prevalence of assaults, as well as the intricacy of actions and resources necessitated by an attack. Minimum sets and partial ordering of subsets of countermeasures form the mathematical basis of the proposed method. Evolving graphs, pruning a graph, and selecting countermeasures are all supported by software tools that make it easier and more efficient to implement the method that is outlined in this work. The methodology is demonstrated using generic graphs that have the potential to represent various systems such as a water distribution system, a pipeline system, or an infrastructure designed for gathering data from sales devices (Baiardi et al., 2009).

### ***Network security risk model (NSRM)***

Henry and Haines provide the NSRM, a framework for assessing network security risks. The NSRM is a graph that depicts an assault in a certain direction. Nodes in a graph indicate system components, whereas edges reflect their interconnections. An essential aspect of the model's purpose is to offer a risk measurement and to calculate that value for both a baseline system and a system with security upgrades. A simplified crude oil pipeline pump station, which is part of a larger process control network and run by a SCADA system, is used to illustrate the model's use. There are a total of eight steps in the NSRM. The eight steps of the NSRM are as follows: First, figure out the system-specific risk metrics. Then, divide the controlled infrastructure into a hierarchical model. Third, use adaptive to describe the ways and effects of process failure. Fourth, list the model processes and ways that they can be interrupted. Fifth, use HHM and AMP-HHM to create an attack scenario. Sixth, use a level and barrier diagram to describe the network security structure. Finally, define the process disruption modes and resource requirements for each attack scenario based on the component access requirements. This step is to determine how each assault will interrupt the process and what resources will be needed to get access to the various components. Returning to the

system, the ideal attacker policy is established that describes which components and in what order an attacker can take advantage of the system. There is an assessment of the amount of crude oil lost as a result of an attack, as well as the attack's success rate. The same criteria are used to evaluate a more secure system. It is possible to determine the most cost-effective security solution by comparing the risks and costs of several expanded versions of the system. All the system's parameters may be calculated using the approach outlined by Henry and Haimes. When attempting to estimate the parameters of a computation in the absence of statistical data and given that each system is unique, it is important to contact specialists (Henry & Haimes, 2009).

### *A tree of counterattack measures*

Risk assessment using the Attack Countermeasure Tree (ACT) is described by Roy and colleagues, who add information on security countermeasures to the widely used attack tree idea. Attack, detection, and mitigation events all fall under the ACT umbrella. The cost of an assault and the level of security expenditure can both be factors in determining the effectiveness of an ACT. The cost of an assault is limited by the attacker's budget, which includes the cost of events leading up to the attack. For example, attack scenarios can be generated, including with the necessary information for qualitative as well as quantitative risk assessment, from an ACT. It is possible to identify the smallest possible attack tree using qualitative analysis. Attack probabilities may be derived from the probabilities of individual attacks. There are further recommended formulas for estimating return on investment and return on attack. Each attack vector has at least one defense mechanism that must be included in the minimal set of defense mechanisms. When multiple factors, like set cost or attack probability, are known, they can be combined to find the optimal set. To demonstrate how effective an ACT is, a SCADA system is attacked. In their study, Roy et al. utilized the software program Symbolic Hierarchical Automated Reliability and Performance Evaluator to examine stochastic models of reliability, availability, performance, and performability. MATLAB was used to carry out the optimization (Roy et al., 2010).

### *A system security assessment based on an adversary-driven state*

According to LeMay and colleagues, the ADversary View Security Evaluation (ADVISE) approach may be used to evaluate the security of an application. To an attack graph, it appends the characteristics of a threat. Simulating an attack on a system, identifying the most probable attack vector, and estimating the attack's success probability are the objectives of the approach. According to ADVISE, a security question may be answered in three steps: characterization of the adversary; defining security metrics; creating an executable attack graph; and finally executing the attack graph to get a response. In a security model of a system, a collection of attack stages and an adversary's attributes are provided as a security-relevant system feature. Preconditions, execution time, cost of an attack step, a set of results, distribution of results, distribution of results, distribution of results, distribution of detection results, payment, and changes to state variables are all part and parcel of an attack step. An adversary's attributes include attack preference weight and attack skill level, which are both independent of the system they're attacking from (LeMay et al., 2010).

### *Model for evaluating the threat of cyberattacks*

The method for evaluating the risk of cyber attacks on Information Systems is outlined in Patel and Zaveri and is applied to assess the SCADA system of an industrial chemical plant to showcase its efficacy. Risk assessment, cost–benefit analysis for IT component purchasing, and insurance premium calculation are all possible uses of the approach. Replay capture, spoofing, and denial-of-service are among the seven attack types identified in the literature study and research conducted in this work. The authors further enumerate six distinct categories of loss that can result from an assault, including control loss, product loss, staff time loss, equipment damage, and prevention. They also provide the corresponding probabilities of occurrence for each category of loss at each given time  $t$ . Chemical plant professionals assess the extent of damage caused by an assault in Patel and Zaveri, considering several aspects such as the nature of the attack and associated losses. Each sort of loss has a formula presented in the article. To determine the cost of preventing a certain form of attack, for example, the cost of upgrading IT components that are resistant to this type of attack is multiplied by the risk of preventing this sort of attack. The suggested approach might be used to determine the overall expected revenue loss caused by all forms of cyber assaults.

Although it is indicated, there are no information given about what kind of program was used to automate this procedure (Patel & Zaveri, 2010).

### ***Protection against attacks on key infrastructure via computer simulations***

RAIM is a four-part SCADA security architecture. In Ten et al., we learn about real-time monitoring, anomaly identification, impact analysis, and mitigation measures. The framework's real-time monitoring and anomaly detection modules rely on system logs to gather data for the effect analysis that follows. Four phases make up an impact study, which looks at how an assault on a SCADA system could affect intrusion behavior and the system's overall security. A Cybernet's system configuration; power flow modeling; vulnerability index computation; and security enhancement A cyber security vulnerability index illustrates the probability of a specific intrusion scenario, the likelihood of penetrating a specific leaf, or the potential of an entire attack using an attack tree. Impact analysis is built this way. Indexes are built based on information gleaned from prior invasions, as well as information on the effectiveness of security measures and the rules that govern passwords. You must do port audits and use secure passwords in order to determine the leaf vulnerability index. Using a test subnet of the network that regulates electric power, the framework may be shown (Ten et al., 2010).

### ***Model for risk detection and management in SCADA systems based on a graph***

Guan et al. provide a digraph model of a SCADA system for a laboratory-scale chemical distillation column. For risk assessment and fault diagnostics, this model gives a formal description of a SCADA system's structure and behavior. An edge with a directed direction exists when a security vulnerability at one vertex might affect the security of another. Using the reachability matrix and partitioning of a graph, it is possible to identify the parts of a digraph that are most likely to be affected by a threat to the starting vertex. A digraph is used for defect diagnosis in a similar fashion to a fault tree. When a failure is discovered in one of the components, it is utilized to determine the cause of the problem. Fault origins are the family tree of all problematic parts. There is now just one source of failure for all the problematic components in the set. A hacker breaks into a company's

network and puts harmful code into SCADA DNP3 communication. This shows how digraph can be used to find problems (Guan et al., 2011).

### ***Detection and reaction to potential dangers***

Cardenas et al. investigated an approach by which sensor networks would identify attacks and respond autonomously. This technique acknowledges and explains the conventional method of calculating risk as the average loss within the framework of a network of sensors in their research. A network attack model that takes integrity and denial-of-service risks into account is being proposed. In order to detect outliers, a linear model is employed to approximate the behavior of a physical system. Non-parametric cumulative sum statistics are used to discover abnormalities. Automated responses to attacks are triggered when anomalies are discovered, and they wait for human intervention before they are completed. Using a model of the Tennessee-Eastman process control system given in Ricker, cyberattacks were simulated on a chemical reactor. Trials showed that the suggested risk assessment model helps decide which types of attacks and sensors should be given the most financial attention (Cárdenas et al., 2011).

### ***Nuclear power plant cyber security risk assessment***

It is recommended by Song et al. that an evaluation approach for cyber security risks may be used in the development of nuclear power plant instrumentation and control systems. For system and component design and equipment supply stage cyber security risk assessment, six steps are laid forth in this technique. The first step is to identify the system and create a cyber security model. Asset and effect assessment is the second step in the process. The third step is threat analysis. Vulnerability assessment is the fourth stage. Security control design is the fifth stage. Stage 6 is a penetration test. The article sums up the applicable NIST standards for each phase, describing the tasks that must be carried out. A compilation of potential attack scenarios has been produced to facilitate the investigation of threats. It is recommended to adapt existing vulnerability lists to match the unique features and complexities of the system under investigation when conducting vulnerability analysis. NIST SP 800-82, for example, contains security controls that can be used. Vulnerability scans and penetration testing are the final steps in validating the security control design (Song et al., 2012).

### ***Markov processes based on boolean logic***

Kriaa et al. describe the Boolean logic Driven Markov Processes (BDMP) modeling technique. Combining fault trees and Markov processes, the BDMP formalism allows the modeling of an assault on a system. A BDMP model can produce both qualitative and quantitative results that are valuable for risk assessment. Attacker activity, a dated security incident, and an instantaneous security occurrence are three of the many components used by the BDMP formalism to model attacks. Additionally, gates such as AND and OR, as well as links like classical logic links, Trigger Link, and Before Link, are utilized within the BDMP formalism for the purpose of attack modeling. This study by Kriaa et al. showcases an instance of a STUXNET attack model that has been constructed using the BDPM modeling technique. The success rate and probability of a BDPM model's leaf are outlined below. This means that all possible attack routes may be located and graded in terms of their likelihood or influence on the attack's success. STUXNET BDPM model was studied quantitatively by Kriaa et al. using the KB3 modeling tool. The researchers utilized their own calculations and the writings of security experts to determine how probable it was that the model's ideas would actually function (Kriaa et al., 2012).

### ***A risk assessment based on CORAS for SCADA***

One method for evaluating potential threats to systems that are vital to maintaining security is CORAS. It can be traced back to ISO/IEC 31000. CORAS is built with security-critical systems in mind, with an emphasis on information technology security. Using a wide variety of models across all stages of risk management, the CORAS framework covers every aspect of risk management. This study focuses on the analysis of Francia et al.'s paper about the deployment of CORAS within the framework of a SCADA system. Francia et al. employ the CORAS framework to conduct a risk analysis of a SCADA system. Initially, the determination of assets and their respective levels of significance is undertaken. Subsequently, a compilation of potential hazards and vulnerabilities ensues. A set of threat diagrams is generated using the CORAS modeling language. The threat diagrams depicted in the paper were generated through a participatory brainstorming process involving diverse system stakeholders, including security and risk experts. The paper primarily highlights the fundamental results of a research endeavor and outlines a significant amount of forthcoming

study that has to be conducted. The study conducted by Francia et al. largely showcases the suitability of the CORAS modeling language for the purpose of threat modeling within the specific setting of a SCADA system (Francia III et al., 2012).

#### ***A risk assessment methodology for power control systems based on the PMU***

Yan proposes a Phasor Measurement Unit (PMU) risk assessment approach for SCADA systems. The efficacy of the framework is showcased by simulating a SCADA system for power grids, specifically the IEEE 10 Generator 39 Bus. A system's initial configuration must be determined. The process of identifying and assessing vulnerabilities is accomplished through the utilization of the Duality Element Relative Fuzzy Analysis Method (DERFEM). Subsequently, a comprehensive assault graph is constructed and employed to discern potential infiltration situations while also giving probabilities to each scenario. The document also provides a full description of the System Stability Monitoring and Response System (SSMARS), which is designed to oversee and address possible threats to the stability of the system. The SSMARS system is an online system that is based on Phasor Measurement Units (PMUs). Voltage control techniques are implemented in response to adversarial events that occur on a power system that is being actively monitored (Yan et al., 2013).

#### ***Quantitative technique for assessing SCADA cyber security risk***

Woo and Kim describe a method for assessing cyber security risk in SCADA systems using optimum power circulation and power flow identification, verification, and traceability. The significance of each threat to each component must first be recognized before vulnerabilities can be quantified. Then, each system component is allocated a vulnerability index. Vulnerability indexes are based on historical data and the security aspects of a component. Each SCADA system component's level of hazard is assigned a normalized weighted index for the purpose of calculating risk. Treatment applicability, component vulnerability index, and damage capacity all play a role. In order to determine the asset's worth, the outage cost is used. Given the constraints imposed by generator and line capacity, the optimal power flow is defined as the lowest cost of power generation for all generators. The interdependencies between generators and load terminals in a SCADA system may be evaluated using a graph-theory-based power flow tracing technique. A risk's monetary value

is determined by multiplying its likelihood by its corresponding asset's price (Woo & Kim, 2014).

### ***Cyber-Attack Anticipation and Detection Kill Chain for Railway Defender***

The aim of this study is to employ the Railway Defender Kill Chain (RDKC) framework to anticipate, minimize, identify, and eliminate cyber threats in the railway sector, with a particular emphasis on ensuring prompt and effective response measures. The RDKC system utilizes a wide range of safeguards, technological measures, industry standards, and security protocols to efficiently reduce the likelihood of cyberattacks on railway infrastructure. This study integrates a diverse array of contemporary techniques, regulations, frameworks, designs, and methodologies in order to precisely detect and mitigate vulnerabilities associated with cyber-attacks on railway systems. The proposed architecture would incorporate defensive measures at every step of the IT and OT/ICS cyber-kill chains. These studies aim to develop a systematic process for rail defense that integrates security measures for both IT and OT systems. The RDKC framework encompasses cybersecurity guidelines, resources, a comprehensive RDKC matrix, and defense-in-depth security measures. The RDKC matrix outlines the strategic implementation of IT and OT security measures to effectively counteract and prevent cyber-attacks at every level of the cyber death chain. During the early stages of the CKC, the defender's main goal is to mitigate or restrict the risk of cyberattacks by implementing security measures based on the RDKC matrix. The cells in the matrix can be interpreted as representations of how a specific defensive control could impact a Cyber Kill Chain phase. During the reconnaissance phase, defensive measures are utilized to pinpoint the specific cell where cyber incidents take place, as this is where the detection approach and Cyber Kill Chain converge (Kour et al., 2020).

### ***Smart Grid Cyber Kill Chain-Based Hybrid Intrusion Detection System***

This study presents a proposed methodology for developing a hybrid intrusion detection system (IDS) that combines a network-based IDS, a model-based IDS, and a state-of-the-art machine learning-based IDS. The objective of this hybrid system is to effectively detect and identify unknown and stealthy attacks specifically targeting SCADA networks. The researchers have successfully employed the cyber-kill concept to construct and demonstrate



assault paths and their associated procedures. In accordance with the kill-chain methodology, the hybrid Intrusion Detection System (IDS) examines attack fingerprints included in grid measurements, network packets, and secure phasor measurements to discern various stages of cyber-attacks. In order to demonstrate the validity of our approach, we present an experimental case study within the domain of centralized wide-area protection (CWAP) cybersecurity. This study utilizes the Iowa State University PowerCyber testbed (ISU) as a means of conducting the experiments. The IEEE 39 bus architecture is utilized in our study, alongside the generation of heterogeneous datasets. Additionally, we provide an explanation of several forms of cyber-attacks that have been accomplished. The efficacy of the hybrid intrusion detection system (IDS) is evaluated based on its real-time threat detection capabilities within a cyber-physical context (Singh & Govindarasu, 2021).

***Cyber kill chain-based situational awareness framework for industrial control system***

Situation Awareness (SA) refers to a comprehensive framework that enables the observation, understanding, and anticipation of the current state and future developments pertaining to information security within the context of an Incident Command System (ICS). A suggested architecture for tackling the cyber kill chain for Industrial Control Systems (ICS) is presented, utilizing the Purdue Enterprise Reference Architecture (PERA) as its foundation. The suggested structure consists of the IT SA Centre, OT SA Centre, and Comprehensive SA Centre. The primary responsibility of the Comprehensive Security Assessment Centre is to build and sustain a significant level of security visibility across various contexts. Perception probes may be found inside three distinct areas: the enterprise zone, the demilitarized zone (DMZ), and the control zone. Professional equipment with flow analysis capabilities may consist of several components, such as a firewall, an intrusion detection system, an intrusion prevention system, a log audit system, and a network switch, among other possible options. In the domains of IT) and OT, especially in the context of warning messages, probes play a significant role in detecting and capturing essential signals and elements. The data that has been acquired will be later transferred to either the IT SA Centre or the OT SA Centre. The IT Security Assessment Center is tasked with the storage, integration, and analysis of perceptual data derived from various enterprise zones and probes placed in the demilitarized zone (DMZ). The relationships among the information are afterwards examined. The IT SA Centre

offers a comprehensive range of capabilities for real-time, integrated situational awareness (SA) in addition to various measures for preventing, detecting, responding to, reporting, and mitigating cyber-attacks across the organization's IT networks. These capabilities are based on the center's perception and comprehension of the IT environment, as well as its understanding of the initial stage of the ICS Cyber Kill Chain. The OT SA Centre provides comprehensive and up-to-date situational awareness capabilities by integrating data obtained from OT networks and devices. The OT SA Center offers a range of capabilities for preventing, detecting, responding to, reporting, and mitigating ICS cyberattacks within the organization's control zone. These capabilities are determined by the center's perception and awareness of the OT environment, taking into account the ICS Cyber Kill Chain Stage 2. The Comprehensive Sensor Data Centre facilitates the integration of crucial sensor data from the IT Sensor Data Centre and the OT Sensor Data Centre, with the purpose of generating actionable alerts (Wang et al., 2021).

***Decepti-SCADA: Cyber deception architecture designed to actively safeguard infrastructures***

The Decepti-SCADA architecture exhibits notable improvements in terms of performance and usability compared to previous iterations of SCADA honeypots. The Decepti-SCADA framework's decoys include a modular structure, facilitating ease of construction and integration of supplementary decoys by contributors. The code bases of current honeypot systems often exhibit a high degree of interdependence, hence presenting a significant obstacle for engineers seeking to make contributions to these projects. The majority of current honeypot systems are primarily characterized by their low level of interactivity, which often compromises their effectiveness in deceiving potential attackers. The Decepti-SCADA system employs the use of decepti-SCADA technology in order to create decoys that closely resemble authentic operating systems. This enables more intricate interactions with the decoys and enhances the credibility of the deceptive tactics employed. The utilization of Docker mitigates cross-platform dependencies, hence facilitating the general adoption of the framework. The Decepti-SCADA system differs from prior honeypot systems by offering a visually appealing web-based graphical user interface (GUI) for the deployment of decoys. This feature enhances the usability of the system, particularly for users

who lack knowledge in this domain. The primary aim of the Decepti-SCADA framework is to create an illusion for a network intruder, leading them to believe that they are engaging with legitimate SCADA network components. Concurrently, the framework also serves the purpose of impeding the intruder's progress and promptly notifying security analysts about the presence of an attacker (Cifranic et al., 2020).

### ***Kill chain for railway defense to foresee and detect cyberattacks***

The study utilizes an expanded cyber kill chain (CKC) model and an industrial control system (ICS) cyber kill chain to facilitate detection. It also proposes predictive measures that can aid railway enterprises in anticipating intrusions and effectively responding to them. The comprehensive CKC model encompasses an internal and external cyber kill chain, whereby disrupting the chain at its initial stages empowers the defense to effectively counteract the adversary's deleterious actions. This project integrates an Open System Architecture (OSA) for railroads with the OSA-CBM (Open System Architecture for Condition-Based Maintenance) framework for enhancing cybersecurity in the railway domain. The architecture proposed by OSACBM for railway cybersecurity has a hierarchical structure with eight distinct levels. These levels facilitate the flow of cybersecurity information, commencing from the first stage of data collecting, followed by subsequent stages of data processing, data analysis, incident detection, incident assessment, incident prognostics, decision support, and finally, visualization. The main objective of this study is to utilize the Railway Defender Kill Chain (RDKC) in order to anticipate, mitigate, identify, and promptly address cyber-attacks inside the CKC context. The subsequent statements outline the contributions of the research study: Initially, the OSA-CBM architecture is modified and tailored for the purpose of enhancing railway cybersecurity. Furthermore, the model is adapted to suit the specific characteristics and requirements of the railway environment, hence adjusting the cyber kill chain accordingly. Furthermore, the Railway Defender incorporates the concept of the kill chain. Additionally, the research presents illustrative instances of cyberattack scenarios within the train system (Kour et al., 2020).

## 2.2 Findings from an evaluation of the methodology and framework

In most publications, qualitative and quantitative risk assessment approaches are differentiated, with semi-quantitative methods also being defined. While qualitative approaches classify risk subjectively, quantitative methods attempt to quantify risk mathematically. Most quantitative techniques are probabilistic. In contrast, risk assessment methodologies are categorized as either traditional assessments or baseline controls. Graph-based risk assessment methodologies are widely used in several domains. A number of tree-based risk assessment methods are included in probabilistic methodologies. These include attack tree analysis, incident tree analysis, vulnerability tree analysis, and many different combinations of these. Additionally, risk assessment methods founded on directed graphs are also employed in this domain. In a similar vein, tree-based methodologies aim to assess the probability or reliability of the primary event. The primary differentiation among different tree-based methodologies is in the selection of the top event. The methodologies employed for SCADA systems may be categorized into two distinct approaches: inductive and deductive probabilistic tree-based methods. The purpose of inductive methods is to identify the most probable causes of unfavorable events, whereas the objective of deductive methods is to determine the most likely explanations for unfavorable events. Deductive methods are referred to as backward search techniques, whereas inductive methods are referred to as forward search strategies (Taylor et al., 2002).

The majority of the methodologies discussed in the preceding section rely on historical data, such as event records, to determine the probabilities utilized in risk or effect calculations. Furthermore, case studies or demonstrations based on a simplified generic model of a system or a testbed are frequently used to evaluate these methodologies. The primary concern is the fact that the risk assessments previously examined lack a pragmatic framework for effective risk management. The absence of a technique evaluation was not addressed in the many recommendations. Some recommendations propose the future implementation of a method on a real-world system, whereas only a few papers illustrate the practical application of the method on an operational system. These frameworks provide methodology and then use a testbed to validate the methodology. The International Standard for Risk Management recommends that a methodical approach to risk management should include three main key actions. First, the risk management framework should be able to identify risks. Second, it

should be able to evaluate the probability of an event associated with a risk that has been identified. Finally, it should be able to determine the severity of the problems caused by the event (ISO, 2019). The framework described previously handles one or two essential risk management actions, but not all of them. ISO 31000:2019 offers the following risk management and assessment definitions. Risk management involves a systematic set of planned actions designed to direct and regulate an organization in response to potential hazards. Risk assessment is the whole process of identifying, analyzing, and evaluating risks. Risk identification is the process of locating, detecting, and characterizing potential threats. Risk analysis is the process of understanding the nature of risk and determining its magnitude. Risk evaluation is the comparison of the results of risk analysis with risk criteria to determine if the risk and/or its level are acceptable or bearable. The predominant emphasis in the realm of risk management strategies lies on the stages of risk identification and risk analysis, with comparatively little attention given to the remaining phases. The risk assessment stage is frequently overlooked. Assessing quantitative risk measurements on an absolute scale can be challenging during the evaluation process. As a result, it becomes necessary to provide a comparison basis to facilitate security decision-making. An absence of a clearly established protocol for assessing the results of risk analysis in relation to the specified risk criteria in the proposals was recognized. There are a number of approaches that compare the risk metrics of a system's various security setups. Several solutions for securing SCADA systems have been developed in the literature; however, there is a dearth of research on the development of evaluation tools for these solutions (Al-Dalky et al., 2014).

## CHAPTER 3 - RESEARCH METHODOLOGY

### 3.1 Research Methodology Design

Research design is a technique for collecting data in order to answer the research objectives (Schumacher & McMillan, 1993). Similarly, Mouton defined the research design of a study as the architectural design or blueprint of the study and the subsequent execution of the design, whereas the research process or methodology was the construction process employing methodologies and tools (Mouton, 2001). The problems encountered by modern SCADA systems have been described in previous chapters, and the most effective method for resolving these problems is using a design science research approach. The design science research paradigm derives from engineering and the artificial sciences (Simon, 1996). The axiology of design research emphasizes problem solutions (Vaishnavi & Kuechler, 2004). The design science research is a problem-solving paradigm that aims to expand human knowledge through the development of novel artifacts (vom Brocke et al., 2020). The design research was appropriate for this study because the proposed ABCD framework seeks to expand technology and science knowledge through the creation of novel artifacts that address problems and enhance the environment in which SCADA systems are implemented.

### 3.2 Proposed Approaches and Steps

Design science may be distinguished from formal sciences, such as philosophy and mathematics, due to its focus on constructing systems of logical ideas. Additionally, it varies from explanatory sciences, such as physics and sociology, which aim to describe, explain, and predict observable events (Aken, 2004). The acquisition of design competence is a teachable skill. The transmission of knowledge within the field of design can occur between designers or between a seasoned senior designer and a novice apprentice, facilitated, in part, by the utilization of an expressive medium. When design is articulated as design principles and design rules, it can undergo iterative cycles of explanation and experimentation that bear resemblance to the theory-building and theory-testing cycles observed in the scientific process. Romme and Endenburg proposed a five-step cyclical design process that makes all of these themes and ideas explicit, including the concept of design principles (Romme &

Endenburg, 2006). The present study utilizes the Romme and Endenburg approach and concepts in order to provide a security framework for SCADA infrastructure architecture. The construction of the ABDC Risk Management framework involves a design-science technique consisting of five steps. The initial phase involves the compilation of a comprehensive collection of fundamental ideas, theories, and experimentally substantiated connections that are pertinent for elucidating the present condition of security in SCADA systems. Therefore, the main content consists of the corpus of information pertaining to SCADA architecture and the corpus of information pertaining to cybersecurity. The subsequent stage involves the formulation of a cohesive collection of imperative statements derived from both theoretical frameworks and practical applications. The objective is to establish resilient critical infrastructures that are safeguarded against cyberthreats. Consequently, the design principles of concern should encompass the context-specific design measures necessary to accomplish this objective. The third stage involves the creation of comprehensive and contextually tailored guidelines, which are derived from one or more design concepts. These regulations provide the fundamental basis for design work. The fourth stage involves the use of design principles to provide a visual depiction of the design. Design representations often include of mathematical models and software representations implemented in framework forms. During the fifth step of the process, a design-implementation artifact is developed; the thing can be assessed and altered. The objective is to articulate design information in a tangible format that facilitates its explanation, dissemination, scrutiny, and evaluation, in contrast to the implicit design knowledge that resides inside the cognitive frameworks of designers.

### **3.3 Novelty and Justifications**

As stated in Chapter 1, the term Industrial Control Systems (ICS) comprises both Distributed Control Systems (DCS) and SCADA. The primary focus of this examination will be directed towards the SCADA system in contrast to the DCS. Given the dynamic nature of malware and security solutions, it is evident that malware possesses the ability to launch attacks from several perspectives. The ABCD framework has been developed with the purpose of safeguarding enterprises from malware that seeks to infiltrate their infrastructure by utilizing a solitary domain within a simulated environment. The future expansion of the

model will involve expanding its scope to encompass other domains. Nevertheless, the core notion of the framework will remain adaptable to many domains. By employing statistical data and a mathematical formula, the framework is capable of generating predictions regarding potential instances of assaults. Furthermore, this method has the potential to be extended and utilized in several other sectors. The model's attention will be limited to the provided information and it will refrain from attempting to identify or validate any false or incomplete information, among other considerations. The similarities between IoT and ICS devices lie in their respective features and concerns. Both systems exhibit design, privacy, and security vulnerabilities that render them highly susceptible to exploitation when linked to the internet. Moreover, these gadgets exhibit similar advantages and drawbacks. This phenomenon has the potential to lead to the integration of vulnerable devices into interconnected networks. The challenges described above can only be effectively addressed with the current resources and knowledge available to us.

It is impossible to provide absolute security in any system or environment, hence a degree of risk is inherent. The primary aim of the ABCD framework is to assess the likelihood of a cyber-attack occurring. The approach fails to adequately manage certain persistent concerns, including residual risk. Moreover, Information and Communication Systems (ICSs) are intricate entities that establish connections between many components of information technology, including sensors, actuators, and other OT devices. An interconnected ecosystem including several components might potentially obscure vulnerable points of attack, which may arise from device-specific vulnerabilities or misconfigurations. A comprehensive comprehension of many assault types is important in order to effectively develop and evaluate defensive strategies. This study suggests techniques for modeling realistic attacks, emphasizing the need for datasets that encompass both normal operational data and attack data. The challenge of reproducing attacks on computer systems lies in the exact simulation of genuine abnormal operating situations within simulations. Nevertheless, the replication of every type of assault is unattainable because to the potential harm inflicted by these applications. Certain assaults have the potential to induce a hazardous condition in the operating behavior, resulting in substantial damage to the devices. Furthermore, the limited scope of implemented assault classifications may give rise to concerns regarding the



generalizability of the detection methodology, since it may not be applicable to novel or previously unidentified attack types.

SCADA systems exhibit compatibility with a wide range of protocols, encompassing several options including but not limited to Modbus, M-BUS, SNMP, DNP3, and BACnet. The reliance of SCADA systems on common computer networks and protocols is growing in order to facilitate interoperability across devices manufactured by various suppliers. This examination will mostly examine Modbus, a prevalent open standard protocol utilized in SCADA systems (Chochtoula et al., 2022; Jakaboczki & Adamko, 2015; Kuchar et al., 2022).

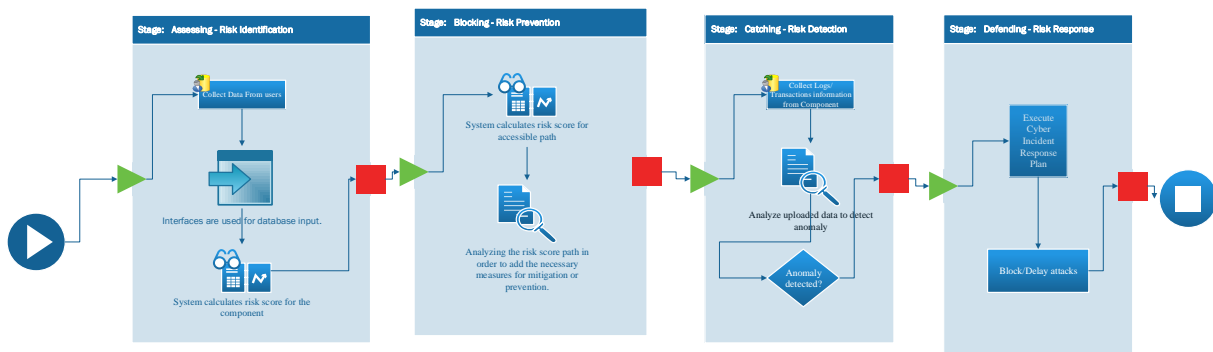
### **3.4 ABCD's Risk calculation framework**

In 1799, George Washington uttered the statement "The best defense is a good offense." The adoption of an offensive cyber security approach can yield significant benefits for cyber professionals in their efforts to protect against threats. Offensive cybersecurity encompasses a proactive security strategy that involves various techniques such as penetration testing and threat hunting. In order to have a comprehensive understanding of the intricacies involved in a cyberattack, it is exceedingly advantageous to assume the perspective and conduct of an unauthorized individual seeking unauthorized access. The field of cybersecurity revolves around the strategic management and mitigation of digital threats by proactive measures and strategic maneuvers. The game in question is a psychology-based guessing activity wherein participants are unaware of the information being concealed or withheld by another. The adoption of an attacking posture as a complement to a defensive plan is now considered the most compelling approach to defense. Moreover, offensive cybersecurity might potentially assume a pivotal function in the detection of vulnerabilities or deficiencies inside a defensive system. The ABCD risk management framework is a proactive approach specifically developed to mitigate the danger of illegal access to SCADA systems. The system encompasses four distinct processes, namely assessment, blocking, capture, and defense. The nomenclature of the framework, ABCD, is derived from the tripartite delineation of its constituent stages. The layers function in a collaborative manner, initially aiming to mitigate a cyber danger, subsequently aiming to minimize the potential harm in the event that the threat materializes, and ultimately aiming to safeguard against a particular attack. The phrase ABCD

framework denotes a risk management framework and model used in combination. The key focal points of the ABCD framework are the prediction of multi-step attacks and the cultivation of security awareness. These objectives are achieved through the use of statistical algorithms and data to forecast forthcoming attack behaviors. The framework presents a pragmatic approach for the identification, mitigation, detection, and mitigation of risks. The proposed model offers a user-friendly and flexible adversarial network framework, allowing administrators to proactively anticipate security situations and facilitate the development of a comprehensive incident response strategy to ensure uninterrupted network connectivity. The underlying concept of this framework enables network administrators to conduct what-if simulations, hence enhancing the efficacy of their decision-making processes.

**Figure 5**

*The stages of the ABCD of the risk management framework.*



### 3.5 Code Organization

The ABCD application is a software program that utilizes the libraries of the Microsoft.NET Framework 4.0, which is built on a windowing system. A user-friendly interface and visuals are recommended for the framework, since it is intended to cater to SCADA network administrators and security managers, who form its target audience. The model's programming language and framework are C# and Windows. C# is characterized as a statically typed and compiled programming language, whereas Python is classified as a dynamically typed and interpreted programming language. This implies that C# exhibits superior speed and efficiency compared to Python, particularly in terms of runtime performance. C# is classified as a compiled language due to its ability to immediately

transform source code into machine code, which can be executed by a processor. The presence of an interpreter is unnecessary. As lines of code are repeated and translated while a program is operating, the interpreter technique is inefficient and sluggish. In contrast, the compiler technique converts the entire program into a single machine code program and executes it. Compiler-based code execution is extremely rapid; however, the code cannot be executed on any other platform besides the one it was developed for (Kwame et al., 2017). Compiler-based programming languages, such as C#, are employed in the context of SCADA systems operating on an Operational Technology (OT) network, which facilitates the execution of real-time data processing tasks. In comparison to Linux or Mac OS, Windows was chosen due to its widespread usage in the workplace. Due to its cost-effectiveness and seamless integration with the .NET framework, SSRS reports are employed as the designated report engine for the assessment reporting within the program. The database engine for the application was selected as Microsoft SQL Server 2019 Express Edition. The model employs many Microsoft .NET libraries, encompassing those inherent to the .NET framework and other libraries like the pcap parser Kaitai library and the modbus emulator easyModbus library. The model code leverages these libraries to facilitate background processing and expands their functionality to meet the framework's specifications. A comprehensive explanation of the utilization of these libraries will be provided in the next chapter.

A component refers to an inclusive enumeration of software, hardware, and network components that establish direct or indirect connections with the SCADA system. The ComponentGroup refers to the categorization of components based on their respective functionalities, such as a database component or a computer component. This categorization will facilitate the recognition of the risk associated with each individual component. For example, a database system may exhibit vulnerability to a SQL injection attack, but a sensor system is often immune to such attacks. The ScoreType refers to a categorization of a risk score that encompasses significant criteria relevant to an organization, such as susceptibility, replaceability, accessibility, and so on. Components have the capacity to offer a substantial amount of data, which will be saved within the Component and Information database known as ComponentType. As mentioned in the preceding chapter, this study focuses only on the

Modbus protocol's relevance to component-specific data, which will be kept in the pcapinfo table.

## CHAPTER 4 – THE ABCD FRAMEWORK AND CASE STUDY

Not only are SCADA systems a prime target for attackers due to their cruciality and visibility, but they are also incredibly vulnerable to security breaches. The background sections of this research highlight the inadequate allocation of resources to cybersecurity. Additionally, the literature review section points out the lack of information regarding the technological processes required to mitigate or address threats. It is necessary to implement an automated processing and machine learning methodology to resolve the problem of insufficient resources dedicated to cybersecurity for the SCADA system. In order to address the oversight in risk assessment during investigations mentioned in the literary review section, the frameworks suggest implementing a systematic approach to gather and evaluate data specifically for the assessment stage. This study also proposes the use of a framework model as an evaluation tool to demonstrate the effectiveness of the framework. The ABCD risk management framework is a four-step framework for effectively managing cyber risk in SCADA systems. Each phase of the framework corresponds to one of the steps in the risk management process proposed by Wheeler, which include assessing, identifying, monitoring, reporting, and responding to the risk (Wheeler, 2021). Each phase of the ABCD framework will be described in detail in this chapter. Phases 1, 2, and 4 of the experiments will be conducted in a simulated environment. The SQL server will be utilized to create random numbers for risk scores, risk classifications for the components, and establish a direct link. The phase 3 experiment will center around a case study that investigates multiple Modbus TCP strategies for attack using a publicly available dataset.

As mentioned from previous chapter, the experiments for this research will follow a five step process for design science research as suggested by Romme and Endenburg (Romme & Endenburg, 2006). The first step involves the compilation of pertinent academic and practical information. The second step involves the development of guiding design concepts. The following step involves the formulation of design guidelines. The framework is designed in the fourth step. The fifth step comprises experimentation and implementation. This chapter will undertake a comprehensive analysis of the ABCD framework proposed by this study, utilizing a series of four experiments. The overall experience will be contingent upon the

probability associated with the successful prevention or mitigation of SCADA assaults at each step of the framework. The ABCD risk management framework provides a realistic strategy to identify, prevent, detect, and respond to risk, much to the NIST Cybersecurity Framework (CSF) to safeguard systems.

#### **4.1 Assessing Phase**

The main aim of this phase is to identify risks. The framework suggests that risk identification should involve a comprehensive analysis of all components that are directly or indirectly involved in the communication or composition of the SCADA system. The proposed methodology recommends the utilization of a risk score to enable the quantification of risk, hence facilitating comparisons across different components. This approach offers benefits for resource management, particularly in situations when resources are limited. Risk scoring further offers a graphical depiction, therefore establishing a sense of responsibility, and can be associated with subsequent measures.

##### ***Experimental Set Up***

###### **Step 1 - Compile theoretical and practical principles learned.**

As outlined in Chapter 1, SCADA systems exhibit vulnerabilities due to inherent infrastructure and design deficiencies, the system's ongoing expansion, and the cost and scale limitations that impede the replacement, modification, or updating of outmoded system components. SCADA system attacks manifest through several methods, encompassing the exploitation of vendor support backdoors, remote access infiltration, as well as direct or indirect assaults on SCADA system components. Perpetrators employ several methods to illicitly infiltrate a network or computer system with the intention of exploiting inherent weaknesses. Given the extensive magnitude of SCADA systems, it is unfeasible to effectively address all conceivable avenues of attack. During this phase, our main goal is to mitigate the occurrence of attacks on SCADA system components that have either direct or indirect links. Several factors contribute to the issue that will be examined in this phase. The interconnections between components of SCADA systems are characterized by a lack of comprehensive visibility. Therefore, it is important to have a comprehensive inventory of components inside a SCADA system. Furthermore, the components that possess direct or

indirect connectivity to SCADA systems often exhibit a diverse range of characteristics, including varying sizes, forms, technical implementations, and manufacturing methods. Hence, it is argued that the classification of vulnerabilities should not be confined to a rigid set of predetermined categories.

### **Step 2 - Design Principles.**

Conducting a risk assessment enables the identification and classification of the vulnerabilities discussed in the preceding section. Enhancing a company's cybersecurity posture is of utmost importance in order to mitigate the risks posed by potential attacks and vulnerabilities that may compromise its SCADA system. SCADA systems have issues arising from various cyberattack techniques, operational risk factors, and risks particular to different sectors. The implementation of routine risk assessments allows organizations to enhance their protective measures and ensure the uninterrupted operation of their commercial activities. Assessments play a crucial role in enabling firms to discover possible vulnerabilities and proactively address them before they may be exploited by malicious actors in the cyber realm. The process of assessment plays a crucial role in the realm of risk management by facilitating the prevention or mitigation of security incidents, such as data breaches. Additionally, it assists companies in evading regulatory and compliance violations, therefore averting the related financial burdens.

### **Step 3 - Formulate design principles.**

During this step, the assessment and communication of cyber risk will be conducted through the utilization of quantitative models. The utilization of quantitative methods may be employed to assist in the identification and selection of effective mitigation solutions. The utilization of a quantitative methodology offers the advantage of necessitating a clear and explicit delineation of factors that are deemed to be potential threats, the extent of harm inflicted by these threats, the possible measures to mitigate them, as well as the effectiveness and associated expenses of these mitigation strategies. Given the distinct and complex nature of each component inside a SCADA system, it is recommended that they be classified based on their individual features. For example, information technology components may exhibit susceptibility to SQL injection vulnerabilities, but operational technology (OT) components, which mostly lack interfaces, may not demonstrate the same vulnerability. The establishment of a framework for the quantification of cyber risk enables the enhancement of decision-

making processes in the selection of mitigation strategies. When employing more realistic assumptions, it becomes possible to predict not just the effects of a particular mitigation approach, but also the consequences of other combinations of mitigation measures.

### *Experiment Implementations*

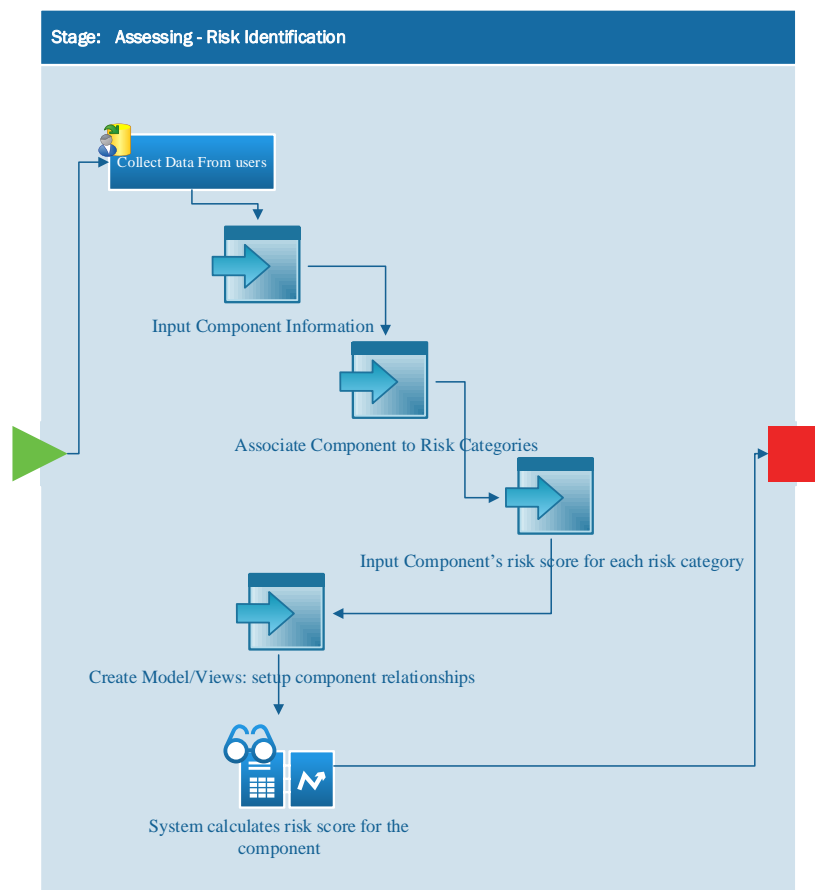
#### **Step 4 - Construction of the Framework.**

The process flow for the assessing step is illustrated in Figure 7. However, the process starts with gathering data from both IT and OT staff. At this juncture, it is imperative to secure the backing of both the IT and OT divisions of the system, as their involvement is vital for ascertaining the significance of assets and processes, identifying risks, assessing consequences, and establishing risk tolerance. Subsequently, every individual component will be inputted into the database and allocated specific risk classifications. The primary aim of this endeavor is to ascertain and systematically document all resources encompassed within the purview of the evaluation. Subsequently, a score is allocated to each risk category. The computational model will generate a probability value corresponding to the risk score. The risk probability refers to the chance or possibility of a particular occurrence occurring. One issue pertaining to these classifications is their frequent ambiguity and varying interpretations, contingent upon whether the analyst is engaging with IT or OT staff. The study proposes the allocation of risk probability ranges to each of the aspects within the group as a means of classifying the components. To derive a mathematical equation for determining the risk score of a collection of assets, it is important to conduct an examination of the security attributes associated with these assets. The objective of this task is to provide a comprehensive analysis of the potential outcomes resulting from the exploitation of a vulnerability, specifically in the context of targeting an asset associated to a project. Threats encompass a range of approaches, processes, and strategies employed by threat actors, which has the capacity to inflict harm upon an organization's assets. Prominent sources of cyber threat information and emerging threats within specific industries, verticals, geographies, and technologies include security vendor reports and advisories, such as the Threat Library, the National Institute of Standards and Technology, and the Cybersecurity and Infrastructure Security Agency. These sources are highly regarded for their ability to provide valuable insights into the ever-evolving landscape of cybersecurity threats. Once all input data has been collected and stored in the database, a



model may be constructed to visually represent the interrelationships among the various components. Developing a network architecture diagram based on the asset inventory list is an effective method for visually representing the interconnectedness and communication channels of assets and processes, as well as the network entry points. This approach facilitates the identification of possible risks, hence enhancing the overall understanding of the network's structure and vulnerabilities. The process flow diagram illustrates that the ultimate phase entails the computation of the risk score. The computation of the risk score involves a multi-level scoring process, in which a score can be designated as either a dependent or independent event. The state of the risk is determined by the level of the model.

**Figure 6**  
*Assessing Phase Process Flow*



***Calculation.***

The calculation of the component risk score was specified in the following manner: The sublevel will assume responsibility for the assessment and management of independent risk, whereas the level will assume responsibility for the assessment and management of dependent risk. Independent risks refer to dangers that are present in isolation from other risks. A dependent risk refers to a danger that manifests in a certain order and is influenced by the presence of other hazards.

The subsequent equation for conditional probability is employed to ascertain the probability of events that are reliant on one another.

$$P(A \cap B) = P(B | A) \times P(A)$$

The conditional probability  $P(B | A)$  represents the likelihood of event B happening, provided that event A has already occurred. The conditional probability of event B, given the occurrence of event A, can be denoted as  $P(B | A)$ , where  $P(A)$  represents the probability of event A and  $P(B)$  represents the likelihood of event B.

The symbol  $P(A)$  represents the probability of event A.

The symbol  $\cap$  is used to denote the mathematical operation of intersection. The notation  $P(A \cap B)$  denotes the probability of the intersection of occurrences A and B, indicating the chance of both events occurring concurrently.

This study suggests using conditional probability to compute the component's risk, since SCADA has been protected for years due to its self-contained nature. Accessibility plays a crucial role in safeguarding the SCADA system as a result of the fact that the historical security of its components has remained largely the same or fever-proof. This is a scientific method for demonstrating that conditional probability is a viable candidate for calculating the risk score for Scada components. Conditional probabilities permit estimations of probabilities to be enhanced by a deeper comprehension of the situation. In this instance,  $P(B | A)$  represents the likelihood of exploiting the previous level of a multilevel risk category configuration for components. Following is the formula to calculate the sublevel risk score.

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

The likelihood of event A occurring is denoted as  $P(A)$ .

The likelihood of event B occurring is denoted as  $P(B)$ .

The probability of the intersection of events A and B, denoted as  $P(A \cap B)$ , represents the likelihood that both events A and B will simultaneously occur.

The probability of the union of events A and B, denoted as  $P(A \cup B)$ , represents the likelihood that either event A or event B will occur.

This study suggests the use of independent event probabilities to compute sublevels, given that SCADA components frequently have numerous characteristics, , manufacturer, operating system, technology, etc. Despite the fact that some of these factors have no actual relationship with one another, they may each pose unique vulnerabilities. We have an HMI system on the OT side and a user computer on the IT side, for example. SCADA HMI is a type of software-based control system architecture that permits operators to monitor the performance of numerous pieces of apparatus and issue process commands and settings. This can be done from a PC with a web browser connected to the control network. Thus, both HMIs and user PCs are PCs and susceptible to numerous security flaws. If a user's PC is compromised, a SCADA system can still function relatively normally. If an HMI system is compromised, SCADA will be severely impacted. This is an example of replaceability and vulnerabilities being independent from one another, so they should be considered two independent events whose risk scores are unrelated.

#### **Step 5 – Implementation and experimentation.**

The structure of SCADA systems often exhibits a high degree of complexity. The vulnerabilities of each component may be categorized into many groups, such as vendor concealed access, hardware vulnerabilities, software vulnerabilities, network vulnerabilities, integration vulnerabilities, and so on. At this stage, our objective is to ascertain the potential occurrence of the risk scenarios delineated during the assessment step. The risk likelihood in a cybersecurity risk assessment refers to the probability that a certain attack will successfully exploit a known vulnerability. Instead of focusing just on historical occurrences, it is more appropriate to evaluate the likelihood of threats and vulnerabilities based on their discoverability, exploitability, and repeatability (Zografopoulos et al., 2021). The probability of cybersecurity threats occurring is not linearly correlated with the historical frequency of such incidents, in contrast to natural disasters like floods and earthquakes, due to the dynamic nature of these threats. At this particular level, the assigned numerical values for Risk likelihood are as follows: Extremely Low corresponds to zero, Low corresponds to one,

Medium corresponds to two, High corresponds to three, and Extremely High corresponds to four. The term "risk score" pertains to the degree of likelihood that an organization will be targeted by attackers and the potential outcome of a threat exploiting a vulnerability. In any given scenario, it is essential to assess the effects on confidentiality, integrity, and availability, ultimately determining the highest impact as the ultimate score.

The model of the framework allows system administrators to tailor the variables for each specific component or collection of components. The nomenclature used in the framework's model refers to individual facilities or groups of components. The experiment starts by employing a diagram as the designated scope. Initially, the component groupings will be categorized according to their suitability for the given environment. As depicted in Figure 8, the creation of the component will be undertaken, followed by its association with the corresponding group, as seen in Figure 9. Figure 8 displays the configuration screen for the component group. Component groups refer to collections of components. Depending on the system, the system administrator may employ various methods for organizing these components. The components can be categorized based on their roles, for example, sensors can be included in the category of field controllers. The system administrator may categorize the component into groups based on its location. The component screen enables the network administrator to categorize their system components according to their requirements. Figure 9 displays the interface for configuring components, allowing the system administrator to perform actions such as adding, deleting, and editing components inside their system. This screen also allows for the addition, deletion, or editing of various component information, including images, network details, and log data. The model maintenance screen is utilized to categorize the many perspectives that are being modeled, as seen in figure 10. The model management panel allows for the addition, deletion, or modification of models. Every system might potentially possess numerous models. Each model can be configured as either a partial or comprehensive representation of the system. Due to the large scale of SCADA systems, it is beneficial for system administrators to have the option of a partial view or a complete view. A partial view allows for a deeper examination of specific component groups, while a comprehensive view provides an overview of the whole network. The subsequent step involves utilizing the Model and Component Maintenance screen to establish the desired

component to be displayed in a certain view, as depicted in figure 11. Model and component maintenance involves the retrieval and presentation of components on the model. The screen depicted in Figure 12 will be utilized for the purpose of risk category maintenance. As stated in the previous chapter, SCADA systems are susceptible to a range of risks. The purpose of Risk category maintenance is to enable the system administrator to classify the risks that are relevant or vital to their system environment. Lastly, the risk category may be linked to each component, along with its corresponding score, as seen in Figure 13. The system administrator will utilize the risk relationship Maintenance screen to assess the severity of their component potential dangers.

Figure 7

### Component Group Maintenance

The ABCD Risk Management Modeling - [Component Group]

Pcap Analyzing Display Component Model Maintenance Models And Components Component Relationships Categories Component Groups

ID : .....

Name :

Description :

		ID	Name	Description
<a href="#">Edit</a>	<a href="#">Delete</a>	5	Database or historian	Physically and digitally secure places to store data gathered, analyzed and processed by SCADA system
<a href="#">Edit</a>	<a href="#">Delete</a>	2	Field controllers	RTUs and PLCs collect and compile data supplied by field instrumentation, preparing it for display and analysis by the human-machine interface
<a href="#">Edit</a>	<a href="#">Delete</a>	1	Field instrumentation	The array of monitors and transmitters on the factory floor that SCADA applications use
<a href="#">Edit</a>	<a href="#">Delete</a>	3	Human-machine interface	Master units that allow humans to supervise the SCADA data acquisition process
<a href="#">Edit</a>	<a href="#">Delete</a>	4	Network connectivity	The SCADA system relies on maintaining integrated network connectivity throughout its operation

Figure 8

### Component Maintenance

The ABCD Risk Management Modeling - [Components]

Pcap Analyzing Display Component Model Maintenance Models And Components Component Relationships Categories Component Groups Component Information Risk score/path

ID : .....

Name :

Description :

Component Group :

Component Color :

Is Active?

Image File :

		ID	Name	Group Name	Is Active	Description	Color
<a href="#">Edit</a>	<a href="#">Delete</a>	78	AMI	Network connectivity	<input checked="" type="checkbox"/>		
<a href="#">Edit</a>	<a href="#">Delete</a>	83	Communication Links	Network connectivity	<input checked="" type="checkbox"/>		
<a href="#">Edit</a>	<a href="#">Delete</a>	61	Communication Server	Field controllers	<input checked="" type="checkbox"/>		#80FF00
<a href="#">Edit</a>	<a href="#">Delete</a>	76	Corporate LAN	Network connectivity	<input checked="" type="checkbox"/>		Blue
<a href="#">Edit</a>	<a href="#">Delete</a>	70	Field Devices	Field instrumentation	<input checked="" type="checkbox"/>		Blue
<a href="#">Edit</a>	<a href="#">Delete</a>	77	Firewall	Network connectivity	<input checked="" type="checkbox"/>		#FF8040
<a href="#">Edit</a>	<a href="#">Delete</a>	66	Historian	Database or historian	<input checked="" type="checkbox"/>		Silver
<a href="#">Edit</a>	<a href="#">Delete</a>	86	HMI	Human-machine interface	<input checked="" type="checkbox"/>	Human Machine Interface, often known by the acronym HMI, refers to a dashbo...	Blue
<a href="#">Edit</a>	<a href="#">Delete</a>	60	LAN	Network connectivity	<input checked="" type="checkbox"/>		Line
<a href="#">Edit</a>	<a href="#">Delete</a>	59	Operating Stations	Human-machine interface	<input checked="" type="checkbox"/>		#FF8000
<a href="#">Edit</a>	<a href="#">Delete</a>	71	Other Control Center	Network connectivity	<input checked="" type="checkbox"/>		
<a href="#">Edit</a>	<a href="#">Delete</a>	63	PC	Human-machine interface	<input checked="" type="checkbox"/>		Aqua
<a href="#">Edit</a>	<a href="#">Delete</a>	73	PLC	Field controllers	<input checked="" type="checkbox"/>		
<a href="#">Edit</a>	<a href="#">Delete</a>	72	PMU	Field controllers	<input checked="" type="checkbox"/>		
<a href="#">Edit</a>	<a href="#">Delete</a>	80	Port Server	Network connectivity	<input checked="" type="checkbox"/>		Gray
<a href="#">Edit</a>	<a href="#">Delete</a>	84	Redundant LAN	Network connectivity	<input checked="" type="checkbox"/>		
<a href="#">Edit</a>	<a href="#">Delete</a>	57	Remote Terminal Unit (RTU)	Human-machine interface	<input checked="" type="checkbox"/>	A remote terminal unit (RTU) is a microprocessor-controlled electronic device that...	#004080
<a href="#">Edit</a>	<a href="#">Delete</a>	81	RTU	Field controllers	<input checked="" type="checkbox"/>		
<a href="#">Edit</a>	<a href="#">Delete</a>	56	SCADA Master	Human-machine interface	<input checked="" type="checkbox"/>	The master station displays the acquired data and also allows the operator to perf...	#FF8080

Figure 9

*Model Maintenance*

The ABCD Risk Management Modeling - [Models]

Pcap Analyzing Display Component Model Maintenance Models And Components Component Relationships Categories Component Groups

ID : .....

Name :

Description :

Search Clear Save Add .....

	ID	Name	Description
<a href="#">Edit</a> <a href="#">Delete</a>	3	1 Monolithic SCADA System	First Generation
<a href="#">Edit</a> <a href="#">Delete</a>	4	2 Distributed SCADA Systems	Second generation
<a href="#">Edit</a> <a href="#">Delete</a>	5	3 Networked SCADA Systems	Third Generation
<a href="#">Edit</a> <a href="#">Delete</a>	6	4 Internet of Things SCADA System	Forth Generation
<a href="#">Edit</a> <a href="#">Delete</a>	7	Testing Connection	

Figure 10

*Model and Component Maintenance*

The ABCD Risk Management Modeling - [Model Components]

Pcap Analyzing Display Component Model Maintenance Models And Components Component Relationships Categories Component Groups

ID : .....

Component :

System Model :

Name :

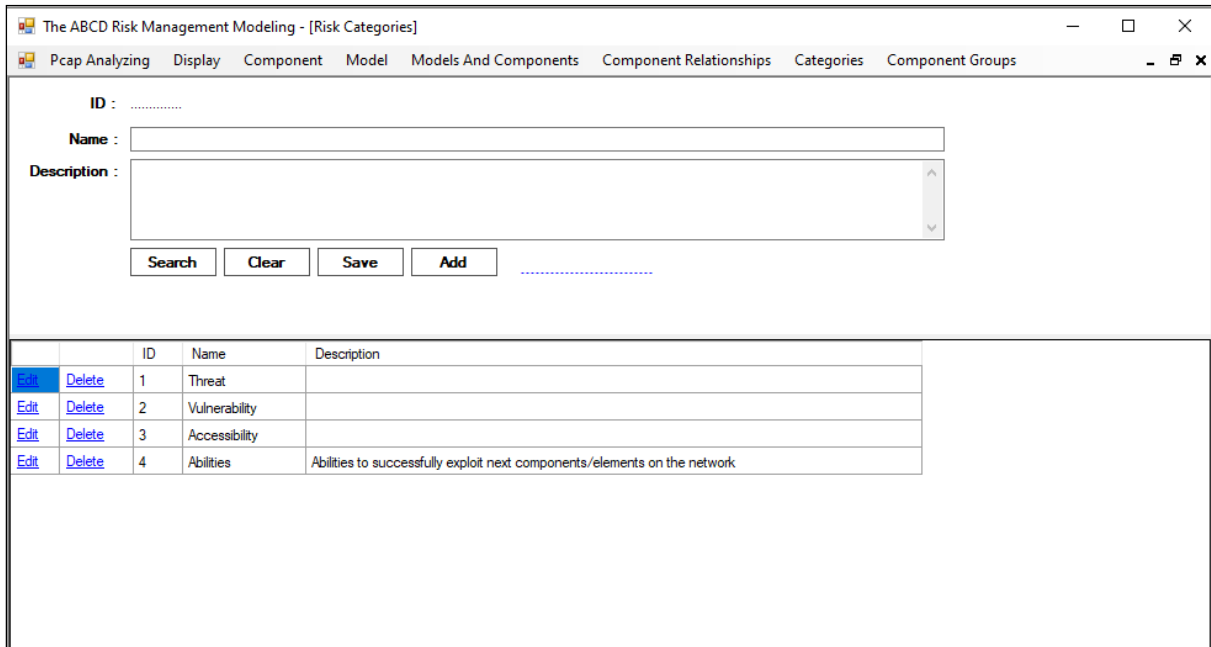
Description :

Search Clear Save Add .....

	ID	Name	Description	Component	Model Name
<a href="#">Edit</a> <a href="#">Delete</a>	60	AMI		AMI	4 Internet of Things SCADA System
<a href="#">Edit</a> <a href="#">Delete</a>	83	Application Server		HMI	4 Internet of Things SCADA System
<a href="#">Edit</a> <a href="#">Delete</a>	81	Communication L...		Communication Links	4 Internet of Things SCADA System
<a href="#">Edit</a> <a href="#">Delete</a>	51	Communication S...		Communication Server	3 Networked SCADA Systems
<a href="#">Edit</a> <a href="#">Delete</a>	42	Communication S...		Communication Server	2 Distributed SCADA Systems
<a href="#">Edit</a> <a href="#">Delete</a>	62	Communication S...		Communication Server	4 Internet of Things SCADA System
<a href="#">Edit</a> <a href="#">Delete</a>	58	Corporate LAN		Corporate LAN	4 Internet of Things SCADA System
<a href="#">Edit</a> <a href="#">Delete</a>	57	Engineering		PC	4 Internet of Things SCADA System
<a href="#">Edit</a> <a href="#">Delete</a>	63	Field Devices		Field Devices	4 Internet of Things SCADA System
<a href="#">Edit</a> <a href="#">Delete</a>	59	Firewall		Firewall	4 Internet of Things SCADA System
<a href="#">Edit</a> <a href="#">Delete</a>	64	Historian		Historian	4 Internet of Things SCADA System
<a href="#">Edit</a> <a href="#">Delete</a>	85	HMI		HMI	Testing Connection
<a href="#">Edit</a> <a href="#">Delete</a>	41	Local Area Netw...		LAN	1 Monolithic SCADA System
<a href="#">Edit</a> <a href="#">Delete</a>	66	Local Terminal		PC	4 Internet of Things SCADA System
<a href="#">Edit</a> <a href="#">Delete</a>	43	Network LAN		LAN	2 Distributed SCADA Systems
<a href="#">Edit</a> <a href="#">Delete</a>	44	Operating Syste...		Operating Stations	2 Distributed SCADA Systems
<a href="#">Edit</a> <a href="#">Delete</a>	45	Operating Syste...		Operating Stations	2 Distributed SCADA Systems

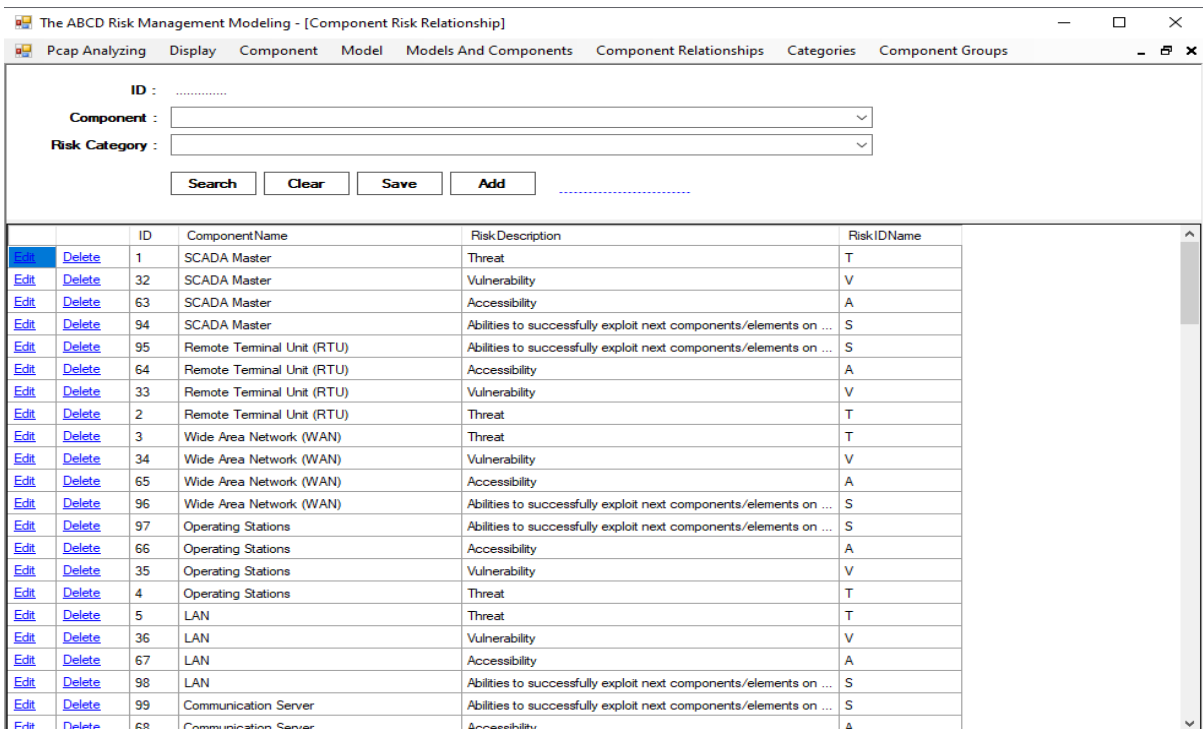
**Figure 11**

***Risk Category Maintenance***



**Figure 12**

***Component - Risk Relationship Maintenance***





The assessment of the risk score for individual components inside a SCADA system requires a hierarchical methodology, wherein each level signifies a dependent event that relies on the successful execution of the preceding level. Moreover, the sublevels included inside each level exhibit different phenomena that are not mutually reliant. In this endeavor, we will classify our hazards into four discrete categories. The classification of SCADA hazards that has been offered is widely acknowledged as a reputable approach (Cherdantseva et al., 2016). Nevertheless, it is advisable for network administrators to tailor the risk classification of SCADA systems to align with their specific environment and operational requirements, owing to the inherent intricacy and ever-changing characteristics of these systems. The risk categories that have been put up encompass accessibility, vulnerability, technological skills, and beneficence, with each category signifying a unique degree of peril. The initial categorization is based on the notion of accessibility. As stated previously, the early and subsequent versions of SCADA systems have the capacity to effectively reduce a significant number of cyber hazards because of its self-contained design and absence of internet connectivity. Therefore, the necessity of accessibility in risk categorization is quite significant, justifying its classification as a level 1 risk. The outermost components of the system architecture are considered as the Level 1 layer. Subsequent components within the system are considered as the Level 2 layer and so on. The use of the rule of dependent event probability calculation enables the potential for level 1 risk to possess a score of 0, resulting in its inaccessibility and subsequently leading to the absence of the remaining risk levels. The third phase of the experiment is concerned with the concept of vulnerability. The susceptibility of IoT devices, such as iPads and smartphones, to security breaches is widely acknowledged to be greater than that of computer components that are equipped with antivirus and endpoint protection software (Butun et al., 2019; Sadeeq et al., 2018). Therefore, it is crucial that the risk score attributed to the vulnerability category of IoT exceeds that of traditional computer or SCADA devices. The technological proficiencies will be encompassed within the third layer of our risk categorization methodology. The prevention of cyberattacks is influenced by a wide range of circumstances, as evidenced by the existing literature on offensive cyber security. The use of diverse methodologies, such as end-point protection and computer configurations encompassing DEP and encryption, plays a pivotal role in fortifying IT components. Nevertheless, it is crucial to acknowledge that within the domain of OT, there

is a notable scarcity or absence of these safeguarding mechanisms. As a result, the risk category associated with IT components will have lower ratings in comparison to OT components in this context. The third step of our project will involve the implementation of beneficence. As stated earlier in the preceding chapter, SCADA systems are mostly targeted by attackers with the main goal of attaining financial gains. In instances of system failure, a system component that lacks inherent benefits or exhibits replaceability will be awarded a diminished score, since attackers prefer to allocate less attention to systems that do not provide advantages or possess reduced effect. The following table, Table 2, provides an illustrative sample of the scoring system to be utilized for this particular experiment. We are evaluating 5 criteria. Accessibility refers to the ability of a component to be accessed via an external link. Vulnerability refers to the presence of recognized weaknesses or flaws in devices or components. An example of vulnerabilities is outdated software. The capacity to efficiently exploit a vulnerability in the system. For instance, a software is recognized to possess a buffer overflow issue, but the system has Data Execution Prevention (DEP) enabled, which therefore diminishes the potential to exploit this vulnerability, resulting in a lower risk score. The correlation between the monetary gain from the attacks and the replaceability score will be independent of each other. If a component lacks financial viability, the likelihood of it being targeted is reduced. Furthermore, if a component is easily replaceable, it also has a reduced likelihood of being targeted for assault. As previously said, SCADA systems are often intricate and may be utilized across several industries. Consequently, the particular factors to be considered may vary depending on the business and the scale of the SCADA system.

**Table 2**

***Common Risk Category***

<b>ID</b>	<b>Name</b>	<b>Description</b>	<b>LevelID</b>	<b>SubLevelID</b>
3	A	Accessibility	1	0
2	V	Vulnerability	2	0
4	S	Abilities to effectively leverage a weakness	3	0
1	T	Threat based on the monetary benefit of assaults	4	0
5	R	Replaceability	4	1

The model utilizes the below SQL script to compute the risk score of the SCADA component. The process of determining the risk score involves three primary components. Initially, the probabilities linked to each risk score will be computed for the component. Furthermore, the risk probabilities for each sublevel will be computed based on the independent occurrences associated with that particular sublevel. Finally, the risk probabilities for each level will be calculated using conditional probabilities.

**Figure 13**

### *Risk Score Calculation SQL Store Procedure*

```

SQLQuery2.sql - DE...Modeling (sa (62))  SQLQuery7.sql - DE...Modeling (sa (68))
set @B = (select top 1 Probability from #tmp where LevelID=@minLevel and SubLevelID=@minSubLevel);
set @A = @A + @B - (@A * @B)
set @minSubLevel=@minSubLevel+1;
end
update #tmp set SubLevelRiskProbability=@A where LevelID=@minLevel;
set @minLevel=@minLevel+1
end

--3rd step calculate the risk score the component
--The calculation is based on independent events
set @minLevel=(select min(LevelID) from #tmp);
set @maxLevel=(select max(LevelID) from #tmp);
set @A=(select top 1 SubLevelRiskProbability from #tmp where LevelID=@minLevel);
set @minLevel=@minLevel+1

while (@minLevel<=@maxLevel)
begin
-- If Events A and B are dependent, the probability that Event A occurs before Event B is: P ( A n B ) = P ( B | A ) * P ( A )
-- P(B|A) represents the probability that event B will occur if event A has occurred.
-- Example if P(A) = 0.25, P(B) = .50, then P(B|A)= 0.25 * 0.5 = 0.125 and P ( A n B ) = 0.25 * 0.125 = 0.03125
set @B=(select top 1 SubLevelRiskProbability from #tmp where LevelID=@minLevel);
set @A = @A + (@A*@B);
set @minLevel=@minLevel+1
end
update #tmp set LevelRiskProbability=@A, RiskScore=round(@A*4,0);
update [dbo].[ModelComponents] set [RiskScore]=round(@A*4,0), [RiskProbability]=@A where ID=@ModelComponentID;
END

```

### *Experiment Results*

Upon the initial loading of the network design, it will assume the appearance depicted in the Figure 15 provided. Individuals are granted permission to relocate and reorganize the various elements in order to accommodate their requirements.

**Figure 14**

*Illustration of an initial network diagram display layout.*

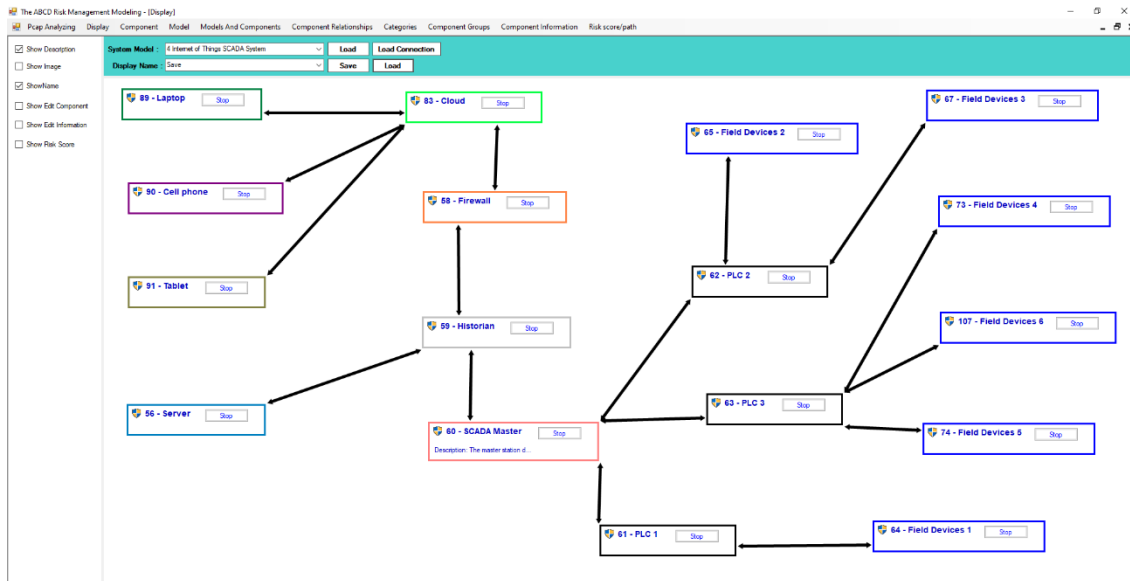
The screenshot displays a software interface titled "The ABCD Risk Management Modeling - [Display]". The interface includes a menu bar with options: Pcap Analyzing, Display, Component, Model, Models And Components, Component Relationships, Categories, Component Groups, Component Information, and Risk score/path. On the left, there is a sidebar with checkboxes for: Show Description, Show Image, ShowName, Show Edit Component, Show Edit Information, and Show Risk Score. The main area shows a vertical stack of five components, each with a "Stop" button and a "Risk Score : 1 Risk probability: 32%" (or similar) display. The components are:

- 106 - Remote Terminal Unit (RTU) 2**: Risk Score : 1 Risk probability: 32%. Controls: V: 3, S: 4, T: 4, R: 4.
- 61 - SCADA Master**: Risk Score : 1 Risk probability: 32%. Controls: V: 3, S: 4, T: 3, R: 4.
- 62 - Wide Area Network (WAN) 1**: Risk Score : 2 Risk probability: 53%. Controls: V: 4, S: 3, T: 3, R: 3.
- 63 - Communication Server**: Risk Score : 2 Risk probability: 56%. Controls: V: 4, S: 3, T: 4, R: 4.
- 65 - Remote Terminal Unit (RTU) 1**: Risk Score : 1 Risk probability: 32%. Controls: V: 3, S: 4, T: 4, R: 3.

The information display pertaining to each component may be adjusted to exhibit a higher or lesser amount of information, hence enabling more flexibility and visibility of the diagram.

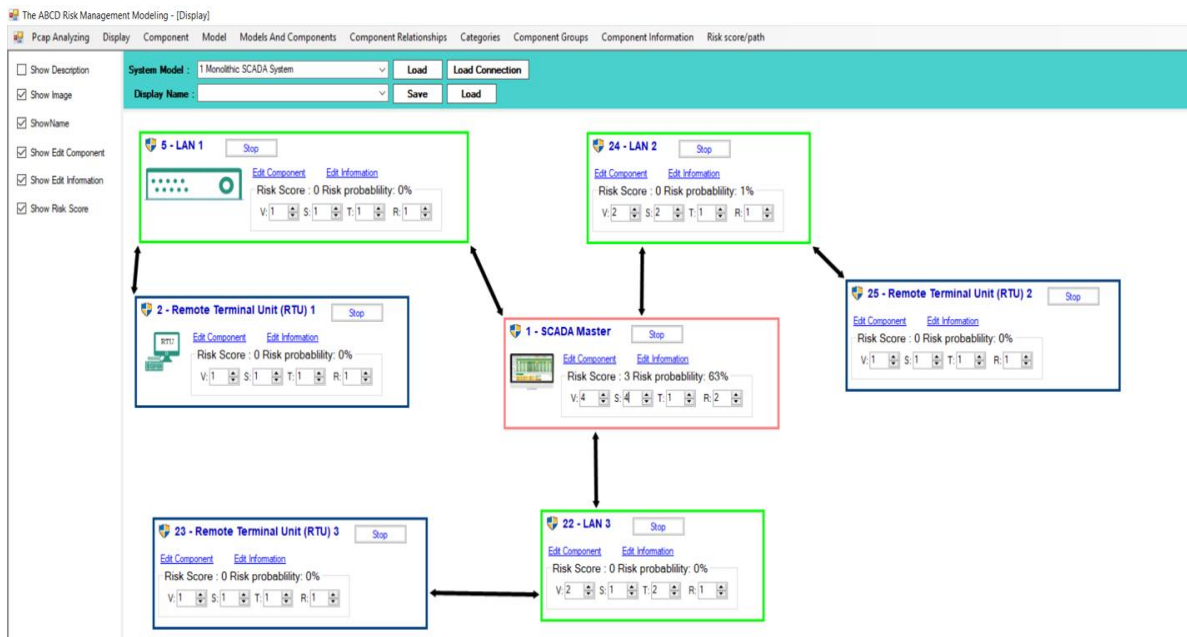
**Figure 15**

*Less-detailed illustration of a condensed variant of the network diagram.*



The network diagram illustrating a conventional first-generation SCADA system, known as Monolithic, is presented below. The first generation of SCADA systems emerged during a time when network infrastructure was not yet established. The initial systems were not designed with the purpose of establishing communication amongst one another. RTUs communicated via LANs.

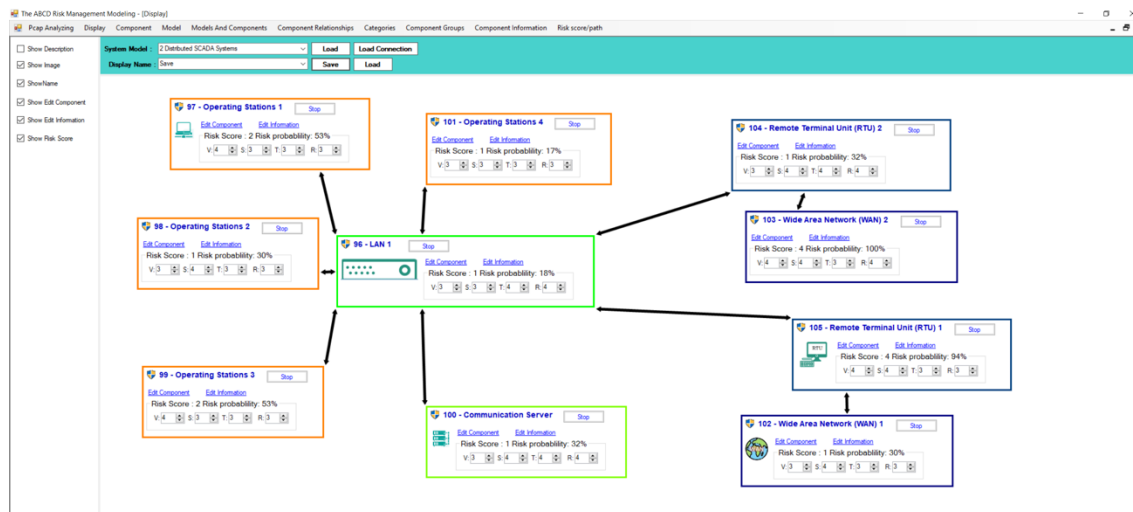
Figure 16

*Illustration of a typical first-generation SCADA system*

The network diagram illustrating a standard second-generation SCADA system, known as Distributed, is presented below. The utilization of Local Area Networks (LAN) may be a prominent feature in the development of the second iteration of SCADA systems. The transmission of information occurred in an almost instantaneous manner across stations, each with their own distinct aims.

Figure 17

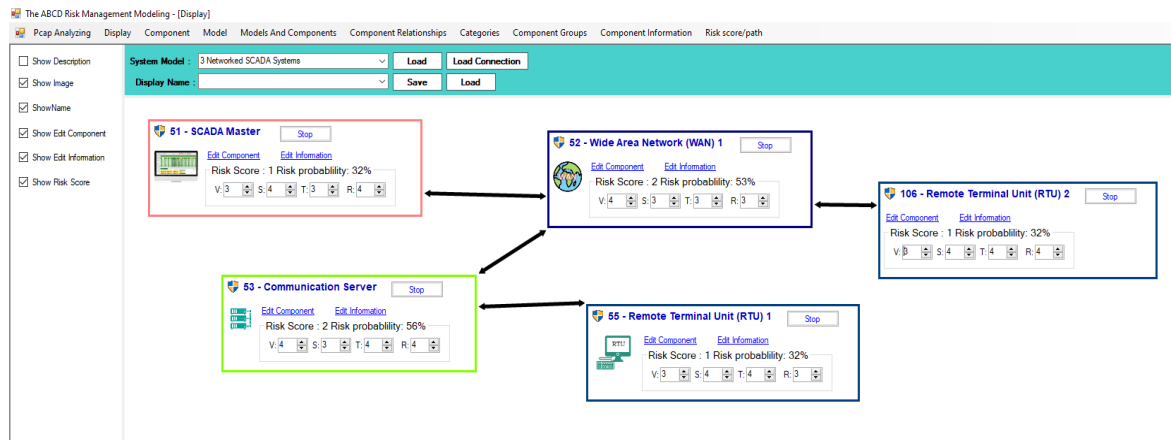
*Illustration of a typical second-generation SCADA system.*



The below diagram illustrates the network configuration of a representative third-generation SCADA system, known as Networked. The present generation exhibits similarities to the preceding generation with regard to wide area network connectivity and monitoring using PLCs. Nevertheless, it has the capability to establish a connection with the internet as well as external devices.

**Figure 18**

*Illustration of a typical third generation SCADA system.*

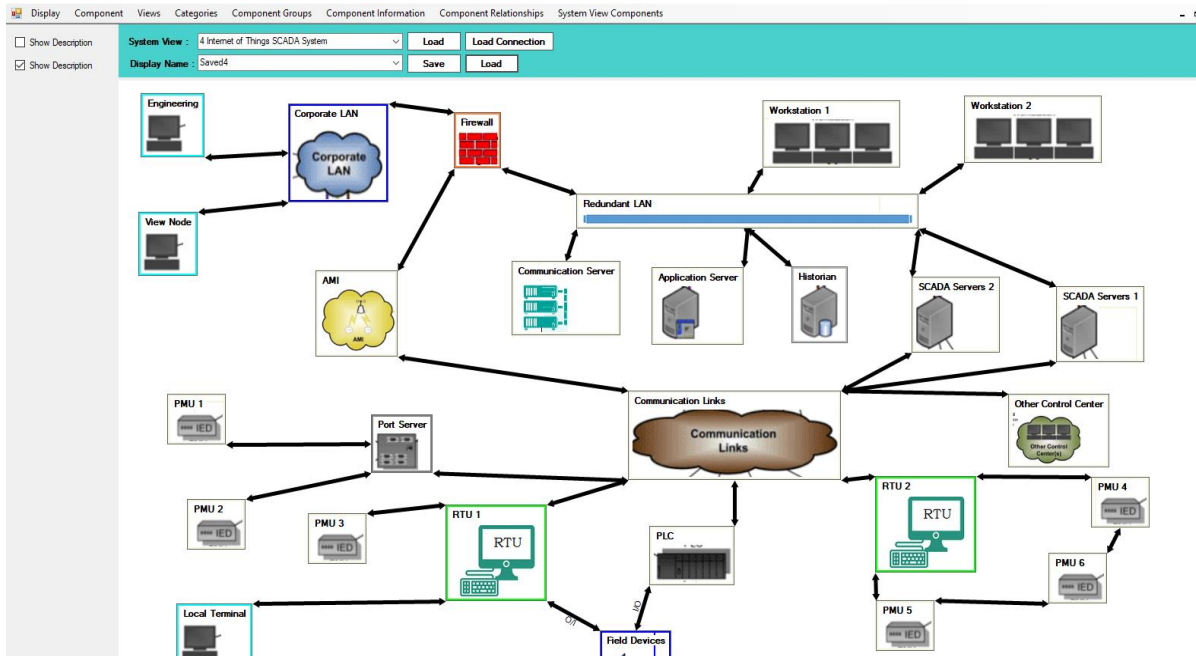


The network diagram depicting a standard fourth-generation SCADA system within the context of the IoT is presented below. By integrating SCADA with cloud computing, the IoT presents SCADA systems as a viable substitute for PLCs. This current generation of SCADA systems incorporates data modeling and intricate algorithms.



**Figure 19**

*Illustration of a typical fourth generation Components of SCADA system.*



The following diagram depicts the visual representation of the risk category and risk score associated with a particular component.

Figure 20

*Illustrative example of a component's associated risk score*

The figure displays three screenshots of a risk management interface, each enclosed in a colored border. Each screenshot shows a component name, a 'Stop' button, 'Edit Component' and 'Edit Information' links, a risk score, a risk probability, and four adjustable parameters (V, S, T, R).

Component	Risk Score	Risk Probability	V	S	T	R
79 - SCADA Master	3	63%	4	4	1	2
96 - LAN 1	1	18%	3	3	4	4
97 - Operating Stations 1	2	53%	4	3	3	3

### *Discussions*

The experimental findings demonstrated that the framework model has the capability to generate a diverse range of SCADA system types, encompassing both rudimentary early-generation SCADA systems and more complex later-generation SCADA systems. The model has the capability to be proportionally adjusted or enlarged in order to accommodate various SCADA configurations. The combined utilization of risk score calculation and SCADA diagram visibility provides a holistic and thorough understanding of the SCADA system, hence exposing potential vulnerabilities with significant risk factors. The present cyber asset inventory integrates IT and OT data in order to offer comprehensive visibility and contextual

understanding of devices, applications, and cloud services within a business environment. Phase 1 of the framework offers a proactive security strategy aimed at preventing unnecessary attacks through the implementation of an efficient network design and the prioritization of risks. The risk scoring method enhances decision-making by prioritizing security risks based on their respective low or high-risk scores, so promptly mitigating cyber risk through improved visibility.

The current stage of the ABCD risk management framework facilitates a comprehensive evaluation of the SCADA infrastructure through the utilization of suitable tools and methodologies, with the aim of identifying concealed vulnerabilities and weaknesses. The framework provides a methodology for computing risk ratings and a system for aggregating and evaluating data. The technique of determining the risk score for a component inside a SCADA system is not universally applicable, given that the SCADA system comprises diverse components with distinct security considerations. As elucidated in a preceding portion of this chapter, elements such as accessibility and replaceability have equal or more significance compared to aspects such as exploitability and vulnerability. The framework offers a holistic perspective of a system in its whole, hence facilitating the integration of knowledge across the OT and IT sectors. As commonly said, visual representation has the capacity to convey a multitude of meanings and messages that would need a substantial amount of textual description. The model diagram serves as a valuable tool for network managers, enabling them to make informed decisions and effectively mitigate risks through elimination and reduction strategies. The ability to determine the directionality of a connection, whether it is unidirectional or bidirectional, and to identify key component attributes such as IP address, MAC address, upload logs, and associated files, is of paramount importance for later stages.

## **4.2 Blocking Phase**

The primary aim of this phase is to mitigate risks. The framework offers an automated approach to giving risk scores to individual components, as discussed in prior sessions. Additionally, this part will provide a comprehensive explanation of the risk scoring process for assault pathways. By utilizing the grading methodology outlined in the framework,

network administrators are able to identify strategies to assist in the mitigation of potential issues. The current phase of the framework is based on a paradigm that allows network administrators to do what-if simulations. This approach facilitates the measurement of progress towards risk reduction and enhances the decision-making process by increasing the quality of decisions made.

### *Experimental Set Up*

#### **Step 1 - Compile theoretical and practical principles learned.**

The concept of offensive security forms the basis of this study. Computer security professionals consistently participate in the mental exercise of hacking. The hacker mindset may be characterized by three fundamental qualities: a strong inclination towards inquiry, an adversarial disposition, and unwavering perseverance. Hackers possess an inherent curiosity and a desire to acquire knowledge pertaining to computer systems, networks, and software, with the primary objective of identifying and exploiting security vulnerabilities. In addition to continuously acquiring new knowledge and techniques to enhance their capabilities and maintain a competitive edge over security measures, they consistently apply the newly obtained plans, methods, and tactics across diverse computer systems. Hackers are often driven by the aspiration to showcase their own expertise and to investigate the limitations of the systems and networks they specifically target. Hackers frequently engage in introspective inquiries, pondering the feasibility of breaching a certain system or network by asking questions such as "Is this susceptible to compromise?" Alternatively, "how might technology be leveraged for unintended purposes?" Alternatively, "how might something be manipulated to maximize its detrimental effects?" However, cybersecurity personnel primarily focus their emphasis on safeguarding against potential threats. Conversely, adopting an adversarial mindset is an essential strategy for fostering critical thinking, which may effectively bolster an organization's cybersecurity stance through the proactive identification and remediation of vulnerabilities. Hackers often engage in iterative experimentation with diverse tactics and processes to identify vulnerabilities within a system. Consequently, their persistence in conducting thorough investigations becomes crucial. Individuals may encounter obstacles and encounter disappointments, yet, they demonstrate resilience and perseverance by not readily surrendering. The individuals in question will persistently exert effort until they have

successfully achieved their predetermined objectives. Hackers often emphasize the importance of cybersecurity teams being able to identify and rectify all vulnerabilities, whereas a hacker only has to identify a single weakness in order to attack the system. The perpetual pursuit of vulnerabilities is a fundamental aspect of their endeavors. Upon seeing a novel system, individuals promptly use their minds to conceive potential methods of circumventing its security measures. By embracing the perspective of a hacker, one may effectively discern security flaws within a system and subsequently build appropriate countermeasures. The development of the most efficient safeguards against unauthorized access by hackers is mostly undertaken by those with expertise in hacking. Hackers will utilize any tactic they deem effective. When faced with an impediment, individuals promptly modify their path in order to circumvent it. Acquiring information is a potential endeavor, however individuals may have alternative objectives in mind. A cyberattack that achieves its objectives may be conceptualized as a tree diagram including interconnected nodes and edges, where the edges symbolize the movement of attackers between nodes. Instead of directing attention on a solitary prospective avenue for assault, the key is in discerning all the boundaries, removing superfluous ones, and establishing points of congestion. Threat modeling can prove to be an invaluable tool in several contexts. The death chain refers to a sequence of operations carried out by an attacker within a system. In order to fully understand this concept, it is necessary to examine the process of threat modeling. Subsequently, precautionary steps might be employed to deter unauthorized individuals from replicating such actions. In the event that an assailant is able to discern credentials by examining the source code of an application, a potential resolution is refraining from utilizing hard-coded credentials.

In order to impede the progress of an assault or redirect it towards an alternative portion of the SCADA system, a viable approach involves augmenting the system's architecture with increased intricacy and more layers. The objective of this phase is to devise a strategy to bypass a security measure implemented for a component that is currently lacking substantial knowledge or information. As elucidated in Chapter 1, a significant obstacle in ensuring the security of a SCADA system lies in the presence of several uncertainties. Traditionally, the OT and IT environments are not commonly seen in conjunction. The OT security specialist will possess knowledge pertaining to a certain segment of the SCADA

architecture, whilst the IT security specialist would possess knowledge pertaining to the remaining segment. Furthermore, SCADA systems consist of diverse interconnected components, each possessing individual attributes like interface, hardware, communication protocols, and more. The presence of vulnerabilities in network devices, protocols, operating systems, SCADA software, and other applications deployed on SCADA computers might potentially enable an adversary to gather information, disrupt, or manipulate SCADA operations. Hence, it is important to acknowledge that any elements, devices, or evaluators linked to a SCADA system have the potential to function as a point of entry, hence leading to data breaches and other vulnerabilities in terms of security. The inclusion of unknown hazards and the provision of responsibility and visibility for each component will offer significant advantages to the network administrator of a SCADA system. The thorough comprehension of a SCADA system may be facilitated for SCADA network managers by visualizing it from both the OT and IT viewpoints.

### **Step 2 - Design Principles.**

During this phase, the utilization of offensive security will be employed to cultivate a mindset like to that of hackers, with the objective of fortifying our defenses against potential assaults. In order to have a more comprehensive comprehension of the design logic of the ABCD modeling system, it is imperative to understand the potential strategies an attacker may employ to breach a SCADA system. Before an attacker may remotely control or attack a SCADA system, they must successfully complete the following stages. The initial stage in gaining command over a SCADA system is obtaining entry to its LAN. As a consequence of this factor, a significant proportion of SCADA networks are no longer accessible over the Internet. The use of a firewall as a means of segregating the SCADA LAN from the business LAN is widely acknowledged and endorsed as a best practice. The implementation of a firewall serves to mitigate the risk of system compromise by thwarting potential attackers. The demilitarized zones (DMZs) inside the SCADA network are equipped with individual firewalls and access control systems, which may be tailored to align with the specific requirements of the company. The establishment of subnetworks facilitates secure data transfer between the corporate LAN and the SCADA LAN through their interconnection with the SCADA LAN. Implementing a Firewall with a Demilitarized Zone (DMZ) may effectively safeguard the SCADA network by employing encryption for all communication

channels and data related to the SCADA system. However, it is important to exercise caution while setting up the DMZ to avoid potential risks. DMZs are employed to segregate internet-accessible services, hence augmenting the vulnerability of these services. For instance, an unpatched application server located in a DMZ and lacking proper management might provide several vulnerabilities. This server can function as an intermediary between the corporate network and the SCADA network by utilizing a demilitarized zone (DMZ). If the server possesses and activates remote access capabilities, it allows a third party to retrieve data from the SCADA system. Consequently, a determined adversary could exploit this mechanism to infiltrate the computer and gain access to the SCADA HMI. In its unsegmented state, this would result in access to the entire network.

Typically, in order to gain unauthorized access to a computer network, an assailant must initially circumvent the perimeter defense mechanisms provided by the firewall. In order to bypass a firewall, individuals have several options at their disposal. They may opt to exploit an open connection, identify a modem or connection that responds autonomously, or get entry through a reputable peer website. The use of SCADA control is important for the achievement of an effective attack on this network. The subsequent phase involves the exploration and comprehension of the procedure. Familiarizing oneself with the system rules is crucial for an intruder who successfully infiltrates a computer system's local area network (LAN). In the event that the assailant's principal objective is to disrupt the process, there will be no requirement for them to engage in any form of reconnaissance. To successfully breach computer systems, an assailant must possess precise information at their disposal. The individual possesses full authority over the HMI displays and the points database within the system. The database is responsible for storing many types of information, including descriptions, points, and data. To ascertain the identity of a device at the protocol level, it is necessary to utilize the serial number. It is a prevalent practice to establish a connection between a HMI and a database that delineates the operator's interaction with tangible equipment. This connection serves the purpose of comprehending a given procedure and providing the displayed points with significant contextual information. Furthermore, the HMI screen displays the HMI navigation points and process logic. The ultimate stage involves exercising control over the process. Upon acquiring a sufficient amount of information, the assailant initiates an assault with the intention of modifying the aforementioned process. The

most straightforward approach for controlling the computer system is issuing direct orders to the front-end equipment. This equipment is responsible for converting the computer system's point data into protocols that can be transferred to and received by field devices and controllers. The capability for basic authentication is confined to a select few protocol converters or front-end processors (FEP). In many instances, an assailant may easily get control over those devices by initiating a connection and issuing a command. Consequently, an assailant has the capability to assume command of the operation by extracting the operator's screen. Man-in-the-middle attacks have the potential to undermine the computer system protocols employed by LAN devices. In order to modify packets during their transmission, it is necessary to possess a comprehensive understanding of the specific protocol involved. Commands can be transmitted to the network by means of packet addition. He possesses the ability to control the responses of the operator with the intention of deceiving him. As a result, the perpetrator gains the ability to assume control of the operation by generating a graphical user interface under the guise of the operator. An unauthorized individual has the potential to gain entry to the computer science local area network (LAN), discover the computer science and process configurations, and manipulate the process by exploiting vulnerabilities in the security system.

### **Step 3 - Formulate design principles.**

Perpetrators consistently devise innovative approaches to circumvent cybersecurity safeguards. Enterprises often have the daunting task of anticipating the timing, nature, and method of future security breaches, as new threats continually emerge. Although there is no infallible method to protect a system, adopting an offensive mindset by thinking like potential attackers can assist SCADA administrators in maintaining a proactive stance against hostile endeavors. The concept of offensive security involves the adoption of an attacker's attitude and strategic approach in order to exploit a given system. At the outset, assailants often possess a limited grasp of the target system. The individuals utilize a technique often referred to as education guessing and investigating, wherein they closely observe the system subsequent to identifying and exploiting a vulnerability to obtain unauthorized entry. The initial stage of the framework is to evaluate every element of the network with the objective of eliminating vulnerabilities, therefore diminishing the probability of a component being exploited as a potential entry point for attacks. The subsequent stage of the framework will



endeavor to counteract the assailants by employing the strategies of decoy and detour. Cybersecurity decoys bear a striking resemblance to their real-world counterparts. Explore the utilization of the learning process inside an unexpected context. Network breaches can occur as a result of the network administrator's inadequate understanding of the system, hence enabling unauthorized access by attackers. Likewise, individuals attempting to breach a network with inadequate understanding may exhibit imprudent behavior by targeting a decoy system or deviating from a critical network element. These principles have the potential to impede or terminate an assault and perhaps expose the names of the perpetrators. The implementation of a detour will result in a decrease in the risk score associated with the component that possesses an unidentified vulnerability in this particular case. The concealment of the real vulnerability of the components inside the SCADA system is attributed to the undisclosed nature of their underlying technology. This lack of transparency introduces unknown aspects that can contribute to a significant quantity of vulnerabilities that remain undisclosed. An undertaking carried out without prior understanding of its potential results. In order to mitigate the impact of an assault or redirect it towards a different segment of the SCADA system, the framework proposes the incorporation of further layers of intricacy into the system's architecture. The aim is to bypass a security mechanism for a component that possesses either a high level of uncertainty in terms of risk or is deemed essential. Adopting the perspective of the attacker and considering asymmetrically, such as contemplating the methods employed to breach the system, might prove useful for security managers. Due to the provision of more initial information to system administrators in phase one of the framework, they are afforded a comparative advantage over assailants at the outset.

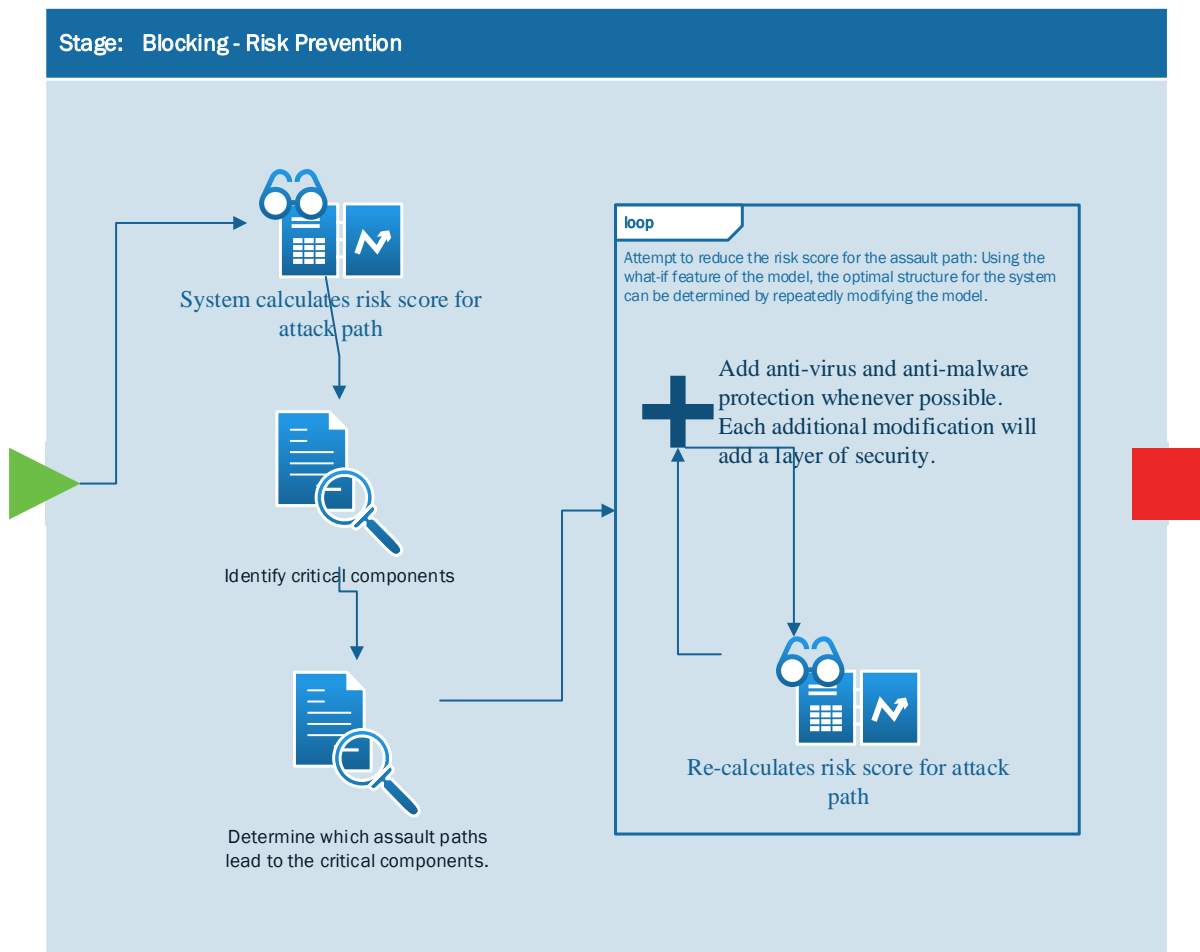
### ***Experiment Implementations***

#### **Step 4 - Construction of the Framework.**

The initial stage of the framework offers a thorough comprehension of all pertinent components within the SCADA system. This diagram facilitates the collaborative efforts of IT and OT in the identification of hazards associated with SCADA systems, as well as the exploration of potential mitigation strategies. The commencement of the second phase of the framework involves the assignment of individual risk ratings to each component. The bidirectional and unidirectional information, which has been categorized in Phase 1 of the

framework, will be employed to ascertain the assault path for every component. The attack path for each component will be calculated using the information pertaining to the connection direction and component risk score. The risk score that can be substituted, as classified in phase 1, is employed for the purpose of identifying the critical element of the component. The primary aim is to shift the assailant's focus from the component with a significant replaceability factor, in order to safeguard either the vital component or the component with a high vulnerability score, hence protecting the vulnerable component. The hypothesis posits that the inclusion of a firewall or decoy component will result in a decrease in the risk score associated with these crucial pathways. The next experiment aims to assess the validity of this hypothesis. During this phase, the utilization of what-if scenarios by the model will serve to elucidate the approach.

Figure 21

**Blocking Phase Process Flow****Calculation.**

In a manner akin to the initial phase, the assessment of risk will use the utilization of mathematical probability calculations to convey the likelihood of both dependent and independent occurrences. The assault trajectory of each component is contingent upon the attacker's proficiency in effectively exploiting the external layer of the trajectory prior to infiltrating the component itself. The conditional probability will be employed to compute the attack route of the component. The computation of access probabilities to our system is contingent upon the occurrence of dependent events. The occurrence of the first event has an impact on the probability of the second event. To get access to the subsequent component in the system architecture, an assailant must initially obtain access to the prior component. The possibility of gaining access to the targeted component is contingent upon the likelihood of

compromising all preceding components within the outermost layer of the schema network. For occurrences to be labeled dependent, one must alter the likelihood of another. Stated differently, a dependent event is contingent upon the occurrence of a previous event. If events A and B are dependent, the probability of both events A and B occurring is expressed as follows.

$$P(A \cap B) = P(B | A) \times P(A)$$

$P(B | A)$  is the conditional probability that B occurs given A. If the probabilities of events A and B are  $P(A)$  and  $P(B)$ , respectively, then the conditional probability of event B given the occurrence of event A is  $P(B | A)$ .

$P(A)$  is the probabilities of events A.

The symbol  $\cap$  represents an intersection.  $P(A \cap B)$  represents the probability of A and B, or the likelihood of two events occurring simultaneously.

The ultimate attack score for a component will be a discrete event consisting of scores derived from all assault pathways that lead to the said component. The events are considered independent as the successful exploitation of any attack path will inevitably lead to the successful exploitation of the component. Hence, the utilization of independent probability computation would be employed for this objective. The calculation of the component attack risk score involves the utilization of the likelihood associated with either event A or event B. The below equation presents the methodology for computing the score of the component attack risk.

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

$P(A)$  is the probability that event A will occur.

$P(B)$  is the probability that event B will occur.

$P(A \cap B)$  is the probability that both A and B will occur.

$P(A \cup B)$  is the probability that either A or B will occur.

Formula to calculate attack path's risk score is as follows:

### **Step 5 – Implementation and experimentation.**

The cognitive process of adopting a hacker's mindset is regarded as the field of offensive security. Adopting a proactive perspective akin to that of a potential attacker, one should engage in a comprehensive analysis of the many avenues via which unauthorized access to a system may be obtained. Subsequently, this acquired awareness may be effectively

utilized to implement appropriate measures aimed at fortifying these vulnerable entry points. The ABCD framework, as advocated in Phase 2, encompasses the fundamental principle being discussed. Attackers commonly target a specific point of entry when attempting to gain unauthorized access to a computer system. They are achieving this through the utilization of several methodologies. The potential causes for this issue include a misconfigured firewall, inadequate server configuration, a vulnerable password, or the acquisition of sensitive information through methods such as phishing or drive-by downloads. The individuals in question will persist in their efforts until they identify a feasible resolution. Understanding the objectives and strategies of attackers can aid in the identification of their techniques and motives. Engaging in a comprehensive consideration of all potentialities is deemed advantageous. Consider the potential utilization of this knowledge. Additionally, it is important to contemplate the potential utilization of this tool in a manner that is adversarial towards the system. In the event that an unauthorized individual successfully infiltrates one of the servers. Subsequently, the individual can employ this tool to get entry into more servers, systems, or networked devices. The initial stage of the proposal involves evaluating the system itself with the objective of eliminating any risks. Every individual element along the assault trajectory may be seen as a distinct layer, and Phase 2 of the architecture suggests the implementation of supplementary layers in order to effectively manage the attack. Considering that all necessary measures to ensure system security have been implemented utilizing the assist model during phase 1, the tasks to be accomplished in phase 2 present an advantageous prospect to include obfuscation techniques that might potentially perplex potential attackers.

### ***Experiment Results***

In the forthcoming experiment, our focus will be directed towards the network diagram depicted below. This figure was generated as part of the initial phase of the experiment. The experiments conducted in Phase 2 offer a systematic approach to determining the risk score of a component by analyzing several attack vectors. The model has the functionality to provide all attack channels linked to a certain component, as seen in the image provided. The primary aim is to decrease the quantity of danger pathways and their corresponding risk ratings.

Figure 22

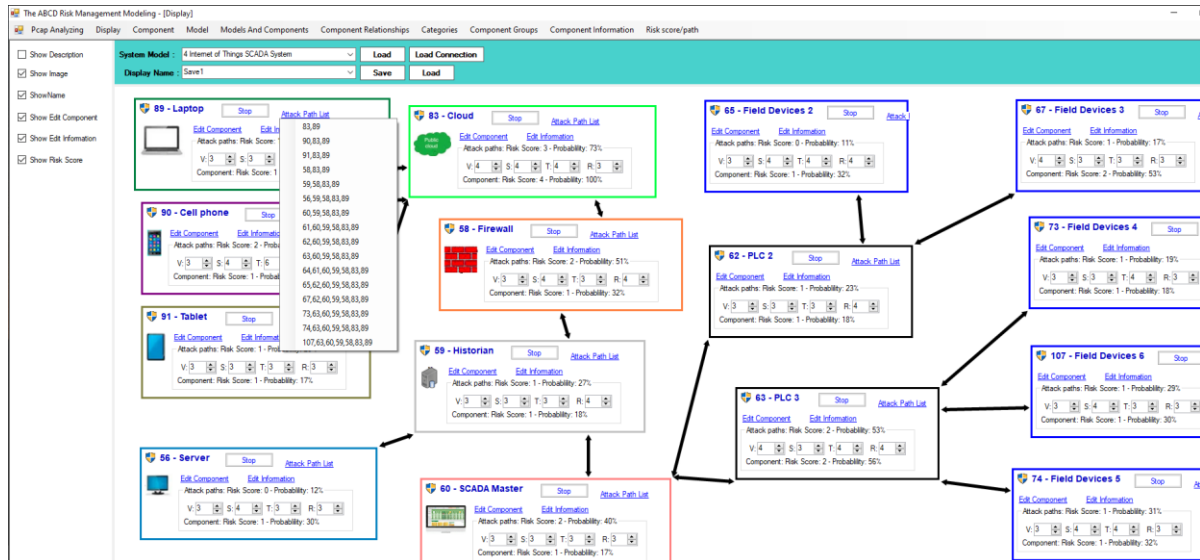
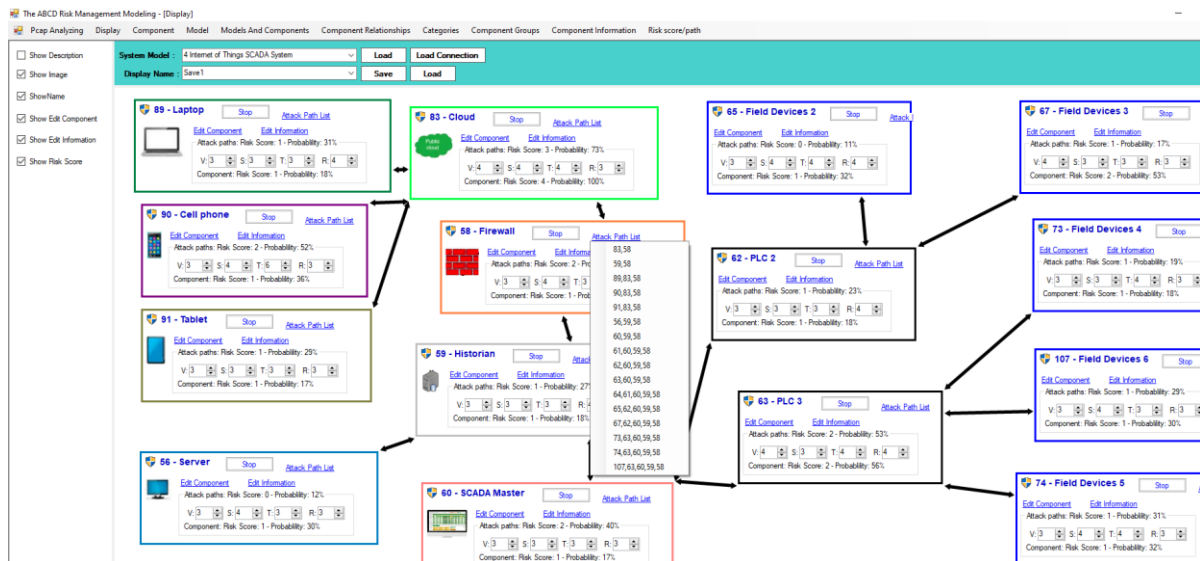
*Illustrative example of a component's associated risk path*

Figure 23

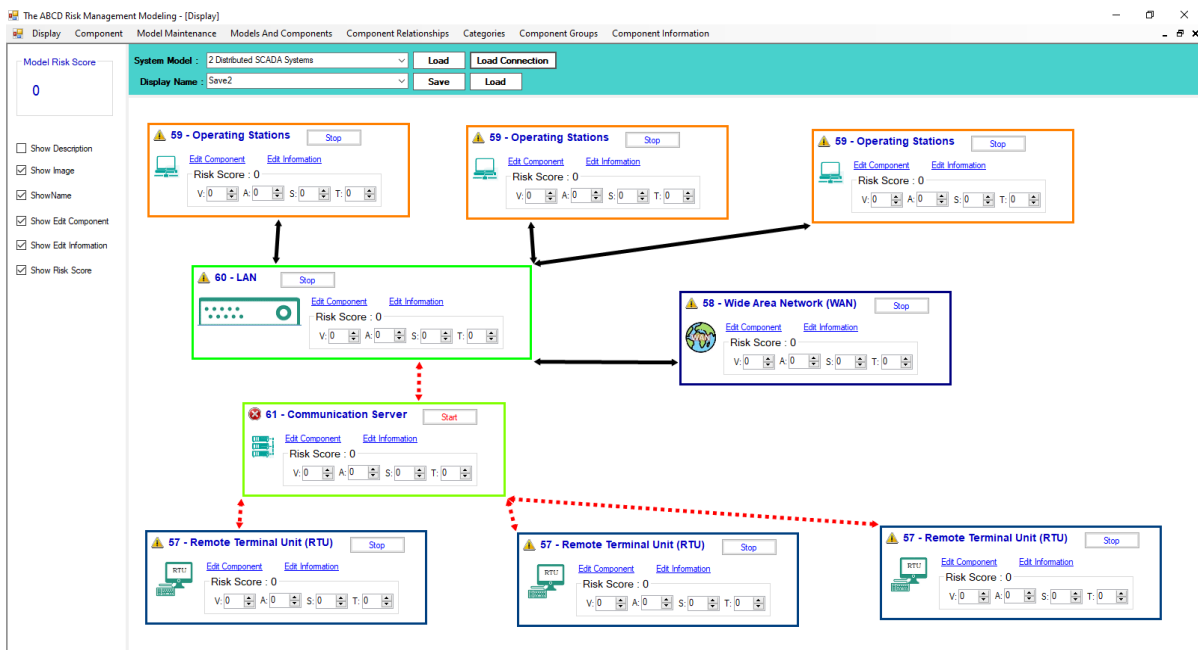
*Illustrative another example of a component's associated risk path*

The figures presented above illustrate that the location of a SCADA component influences its risk route, leading to varying risk path scores.

The inclusion of an additional component inside the system is expected to result in a reduction of the risk score. This is mostly due to the fact that the introduction of this new component would effectively elongate the attack route, hence necessitating the attacker to thoroughly study, exploit, and compromise this additional layer before gaining access to the system's interior layer.

**Figure 24**

*Illustration of components impacted by a security breach.*



## Discussions

In reality, it is not feasible to completely remove danger. Acceptance and transference represent the sole viable approaches to managing risk. When the level of risk exceeds the capacity of a system or the ability of a business to manage it, the process of mitigation becomes necessary. Mitigation is a procedural approach aimed at minimizing risk by implementing actions that effectively counteract the likelihood of an event or the extent of its potential harm. The aim is to decrease the risk score to a degree that aligns with acceptable levels of risk acceptance or transference. The model offers a full perspective of the SCADA system, enabling the selective activation and deactivation of individual components in response to a compromised component resulting from an assault. Additionally, it assesses the

cascade consequences of an attack on a single component within the system. The previous section provides a depiction of the capabilities of this model. As demonstrated before, adopting a comprehensive viewpoint of a system not only facilitates the mitigation of risks but also evaluates the consequences of an attack. To effectively forecast and assess the potential hazards associated with a SCADA system, it is recommended to employ a comprehensive computational approach. This approach entails conducting a multi-tiered analysis, whereby sublevels and risk categories are essential in determining the risk score for each individual component, as proposed in Phase 1. Subsequently, the components of a path will be considered as a layer of attack paths. Each individual layer will be perceived as a protective barrier that effectively hinders any potential attack on the component. The preceding experiment demonstrated that the introduction of an imposter component into the system effectively reduced the likelihood of a component attack. The risk score of a component is influenced not only by its own vulnerabilities and exploitability, but also by the vulnerabilities and exploitability of other components within the attack path. The framework offers a comprehensive methodology for assessing component risk in SCADA systems. The present study not only presents a methodology for mitigating SCADA risk, but also elucidates the influence of cyber risk on subsequent iterations of SCADA systems by employing probabilities of independent and dependent events.

### **4.3 Catching Phase**

The main goal of this phase is to identify and assess risks. During this phase, the framework offers an algorithm that enables the detection of network intrusions through the analysis and monitoring of network traffic, in addition to incorporating event-based learning from historical data.

#### ***Experimental Set Up***

##### **Step 1 - Compile theoretical and practical principles learned.**

The subsequent stage of the framework offers a method for impeding and prolonging an assault. In the event that the attack is impervious to blocking, phase 2 offers a mechanism for postponing its occurrence. The third stage of the framework offers administrators with a mechanism to identify the presence of a cyber assault. Gaining entry to critical systems is just



a portion of the overall challenge, since the subsequent step involves the exploitation of the system. Once an unauthorized individual gains entry into a network, they can exploit the knowledge they have obtained to strategize their progression across interconnected networks, acquire further privileges inside the system and its data, and employ techniques to avoid discovery, among other actions. Once hackers have successfully located a secure access point within a system, they possess the capability to duplicate the assault several times, as required. Certain types of malicious software can remain undetected for extended periods of time, enabling hackers to clandestinely carry out their activities without being noticed. In order to effectively address the conflict, it is imperative to promptly identify the assault, enabling the security expert to launch the cyber response strategy.

Effective communication plays a crucial role in the successful operation of a SCADA network. The exchange of messages between master devices, such as PLCs, and slave devices, such as sensors or actuators, is facilitated by a peer-to-peer communication paradigm. In this paradigm, the slave devices broadcast messages to the master devices and perform actions as instructed by them. The communication protocols utilized in SCADA systems include DeviceNet, ControlNet, Profibus, Modbus, DNP3, and Foundation Fieldbus. Wide Area Networks (WANs) are frequently employed in SCADA systems due to their extensive geographical coverage. Various forms of infrastructure, such as satellites, radio systems, and power lines, can be employed individually or in combination to establish a comprehensive communication network. The software utilized in SCADA systems commonly encompasses the following functionalities: The functionalities encompassed in this system include the presentation of synoptic diagrams and textual information, along with the capability to see them on numerous screens. Additionally, it offers general editing features such as resizing and scrolling, trend analysis, alarm management, logging, archiving, reporting, and the execution of automated control actions triggered by certain events. As previously mentioned, the primary subject of inquiry in this examination is the Modbus protocol. Subsequent investigations will be conducted to explore other approaches.

### ***What is Modbus?***

In the year 1979, the company Modicon, now known as Schneider Electric, developed a communication protocol called MODBUS with the purpose of facilitating communication

between many devices through the utilization of a solitary twisted-pair line. The initial implementation of the method utilized the RS232 protocol, however, it was subsequently modified to operate on the RS485 protocol in order to enhance its speed, extend its range, and enable the establishment of a true multi-drop network. The MODBUS protocol quickly gained widespread adoption throughout the industry, eventually establishing itself as a standard. Modicon, recognizing its significance, took the decision to provide it to the public without any associated royalties. MODBUS is highly versatile in terms of its compatibility with many communication media. These media include twisted-pair cables, wireless connections, fiber optics, Ethernet, telephone modems, cellular phones, and microwaves. This implies that the process of establishing a MODBUS link at a new or existing facility is quite straightforward. One emerging use of MODBUS is leveraging pre-existing twisted-pair cable infrastructure to provide digital communication in aging facilities. The Modbus protocol is an unauthenticated communication protocol that transmits data in plain text format. Currently, the three most often utilized types of MODBUS are MODBUS ASCII, MODBUS RTU, and MODBUS/TCP.

**Table 3**

***Modbus Data Types***

<b>Data Type</b>	<b>Access</b>	<b>Description</b>
Coil	Read/Write	Single bit outputs.
Discrete Input	Read	Single bit inputs.
Input Register	Read	16-bit input registers.
Holding Register	Read/Write	16-bit output registers.

(FernhillSoftware, 2012)

In the MODBUS ASCII protocol, all messages are encoded using hexadecimal representation, specifically utilizing 4-bit ASCII characters. In order to transmit each unit of information, a total of two units of communication are necessary, representing a doubling in comparison to the requirements of both MODBUS RTU and MODBUS/TCP protocols. The MODBUS ASCII protocol is considered to be the slowest among the three protocols. However, it is still suitable for applications using telephone modem or radio communications. The reason for this is that ASCII messages are bounded by characters. As a result of the

delineation of this message, any disruptions in the transmission channel will not lead to the misinterpretation of the message by the receiving device. The consideration of this aspect is of utmost importance when encountering slow modems, mobile devices, unreliable connections, and other problematic transmission channels.

**Table 4**

*The format of a Modbus ASCII transmission*

Start	Unit Address	Message	LRC	End
ASCII 58	2 characters	N characters	2 characters	ASCII 13 + ASCII 10

Field Name	Definition
Unit Address	The PLC Address encoded using two hexadecimal digits.
Message	A Modbus PDU in which each byte is represented by two hexadecimal characters.
LRC	The Address and Message fields undergo a Longitudinal Redundancy Check.
Modbus ASCII	513 characters are the utmost message length

(FernhillSoftware, 2012)

In the MODBUS RTU protocol, binary encoding is utilized to represent data, with the transmission of a single communication byte being sufficient for each data byte. This device is compatible for use over RS232 or RS485 networks that support multiple drops, operating at data transfer speeds ranging from 1,200 to 115Kbaud. The transmission rates of 9,600 and 19,200 baud are the most commonly observed. The Modbus RTU protocol utilizes binary communication, wherein every message is consistently accompanied by a cyclic redundancy check checksum. This checksum serves the purpose of detecting any potential transmission mistakes.

**Table 5*****The format of a Modbus RTU transmission***

Unit Address	Message	CRC
1 Byte	N Bytes	2 Bytes

Field Name	Definition
Unit Address	Address of the PLC encapsulated as a single byte
Message	The utmost length of the Message element in a Modbus PDU is 253 bytes.
CRC	The Unit Address and Message fields' Cyclic Redundancy Check
Modbus RTU	The utmost length of a message is 256 bytes.

(FernhillSoftware, 2012)

MODBUS over Ethernet refers to the utilization of the MODBUS protocol over the Ethernet network, which is effectively the same as MODBUS over TCP. In the realm of networking, IP addresses are employed as a means of communication with subordinate devices, replacing the conventional device addresses. The transmission of MODBUS data is accomplished by encapsulating it within a TCP/IP protocol, namely MODBUS/TCP. As a result, it is imperative for Ethernet networks that are compatible with TCP/IP to also possess the capability to handle MODBUS/TCP. The subsequent section titled "MODBUS Over Ethernet" will provide a more comprehensive examination of this particular iteration of MODBUS. The Modbus TCP protocol incorporates an Modbus Application Protocol (MBAP) message header into the existing Modbus RTU protocol. As the TCP protocol ensures reliable connection service, the inclusion of a CRC check code, which was necessary in the RTU protocol, is no longer needed in Modbus TCP. Consequently, the Modbus TCP protocol does not include a CRC check code. Given its widespread use in the industrial sector, this study will primarily focus on the fundamental aspects and issues pertaining to MODBUS TCP, which is often regarded as the most common industrial protocol (Kim & Tran-Dang, 2019; Leyva et al., 2004; Shukla et al., 2017; Swales, 1999).

**Table 6*****The composition of Modbus TCP messages***

Transaction Id	Protocol	Length	Unit Address	Message
2 Bytes	2 Bytes	2 Bytes	1 Byte	N Bytes

Field Name	Definition
Transaction Id	Identifies the transaction
Protocol	Protocol field value of zero indicates Modbus protocol
Length	Length is the number of bytes that follow.
Unit Address	Address of the PLC encapsulated as a single byte
Message	The utmost length of the Message element in a Modbus PDU is 253 bytes.
Modbus TCP	260 bytes is the utmost message length allowed.

(FernhillSoftware, 2012)

TCP is a networking protocol used for the safe transfer of data between two parties on the Internet. The size of the preamble in a TCP segment exhibits variability within the range of 20 to 60 bytes. The available memory allocation for options is 40 bytes. In the absence of alternative choices, the length of a preamble is 20 bytes; however, if options are available, it must not exceed 60 bytes.

**Table 7*****TCP Header Structure***

Source Port Address (16 bits)							
Destination Port Address (16 bits)							
Sequence Number (32 bits)							
Acknowledgement Number (32 bits)							
H	Rese	U	A	P	R	S	F
LEN (4 bits)	rved (6 bits)	RG (1 bit)	CK (1 bit)	SH (1 bit)	ST (1 bit)	YN (1 bit)	IN (1 bit)
Window Size (16 bits)							
Checksum 16 bits							
Urgent Pointer (16 bits)							
Options/Padding (up to 40 bytes)							

**Table 8*****TCP Header field definitions***

Field Name	Definition
Source Port Address	A 16-bit field containing the port address of the transmitting application.
Destination Port Address	A 16-bit field containing the application's port address on the host receiving the data segment.
Sequence Number	A 32-bit field that contains the sequence number, or the byte number of the first byte sent in the segment. It is utilized to reconstitute the message at the receiving end from the out-of-order segments.
Acknowledgement Number	A 32-bit field that contains the acknowledgement number, the number of bytes the receiver anticipates receiving next. It serves as confirmation that the preceding bytes were successfully received.
Header Length (HLEN)	A 4-bit field that denotes the size of the TCP header in terms of the number of 4-byte words.
Control flags	These six 1-bit control bits govern connection-related operations. These six indicators are SYN (synchronize sequence numbers), ACK (acknowledgement number is valid), FIN (terminate connection), PSH (request for push), URG (urgent pointer is valid), and RST (reset connection)
Window size	A 16-bit field that specifies the window size in bytes for the TCP sender.
Urgent pointer	A 16-bit field is only valid when the URG indicator is set. It is used to indicate data that is required immediately and must reach the receiving process as soon as possible. This field's value is appended to the sequence number to determine the number of the final urgent byte.

The susceptibility of the Modbus TCP protocol to many sorts of attacks has been identified, including SYN flood, TCP Reset, and TCP session hijacking. The predominant types of Modbus TCP attacks may be classified into one or a combination of the following three groups. The subsequent section will provide a description of these various sorts of assaults of SYN Flood attack. A SYN flood refers to a form of distributed denial-of-service (DDoS) attack that aims to render a server unresponsive to genuine traffic by depleting all the server resources at its disposal. The TCP SYN Flood Attack leverages the TCP three-way handshake protocol to create a dependable session between a sender and a recipient. The TCP SYN flood attack involves the deliberate inundation of the Modbus Master with a high volume of TCP connection requests. These requests originate from potential Modbus customers and are accompanied by falsified source IP addresses and random destination TCP

ports. The initiation of the assault normally involves the utilization of a port scanner in order to ascertain the inventory of accessible TCP ports on the targeted host. Subsequently, the assailant has the option to select a certain open TCP port number and employ it as the destination port number inside the TCP SYN flood assault packets. Moreover, the assailants have the capability to transmit many SYN packets in succession to each individual port on the server being targeted. The server, unaware of the ongoing assault, receives several requests to initiate contact that seem to be authentic. The system generates a SYN-ACK packet in response to each connection attempt made on an open port.

The TCP Reset Attack is a type of malicious activity wherein assailants transmit counterfeit TCP RST (Reset) packets to the targeted host. The TCP reset assault involves the utilization of a manipulated TCP segment, generated and transmitted by an assailant, with the intention of misleading two targets into terminating a TCP connection, thereby disrupting potentially critical communication channels. Both the transmitting and receiving computers have the capability to send packets with the reset indicator enabled. In the event that an unauthorized individual intercepts network traffic, it is possible for them to simulate the sending computer and transmit a packet to the receiving computer, wherein the RST flag is enabled. Subsequently, the computer that is receiving the data will proceed to terminate the connection. The individual responsible has since severed the link.

The act of hijacking a TCP session, commonly referred to as a man-in-the-middle (MITM) assault, A Man-in-the-Middle (MITM) attack transpires when an assailant clandestinely acquires access to the communication channel connecting two entities. The phrase "man-in-the-middle attack" encompasses a range of methods, such as fake command injection and false access injection, which have resemblance to replay attacks. Replay attacks commonly entail the interception and subsequent use of unaltered valid network traffic. In contrast, a man-in-the-middle assault involves the manipulation of pre-existing network packets or the creation of new ones. In the context of a false command injection attack, the perpetrator possesses the ability to transmit inaccurate orders to the PLC with the intention of disrupting the operational sequence. This disruptive behavior may manifest in several forms, including but not limited to halting valve operations while the pump remains active or triggering an unscheduled shutdown of the plant, among other potential actions. The objective is to execute arbitrary commands to seize control. In the context of a fake access injection

attack, an adversary possesses the ability to transmit deceptive access commands through the use of Modbus requests that incorporate function codes and beginning addresses obtained through a process of trial-and-error. The assailant has the capability to consistently submit erroneous Modbus requests. These requests will cause the PLC to remain engaged and occupied while it handles atypical replies, leading to the PLC's inability to react to legitimate HMI instructions and inquiries. The consequence of this action is the occurrence of a denial-of-service attack. In a replay assault, the assailant intentionally stores Modbus signals and subsequently transmits them to certain nodes, such as an HMI or PLC. Due to the absence of a time stamp field in Modbus frames, PLCs and HMIs face the challenge of discerning whether a received answer corresponds to a current request frame or an earlier one. The answer inside the framework may suggest that the field parameters are no longer up to date, nevertheless, the HMI will still process the framework, resulting in the SCADA replicas displaying inaccurate information. Likewise, the PLC will read the control command and initiate the activation of the actuators, therefore disrupting the ongoing activities. The aforementioned assault has a very discernible pattern characterized by the inclusion of seemingly up-to-date data, but in actuality, it comprises obsolete information (Rajesh & Satyanarayana, 2021). In order to achieve a successful Man-in-the-Middle (MITM) assault, the perpetrators must possess two key prerequisites: firstly, they must be situated inside the same subnet as the targeted personal computer (PC); secondly, they must possess the capacity to corrupt the Address Resolution Protocol (ARP) caches of the individuals being victimized. The individual intentionally manipulated the Address Resolution Protocol (ARP) caches of the Rx Router and the PXI Modbus Master with the aim of carrying out the malicious act. Upon executing ARP poisoning, the perpetrators have the capability to intercept network communication from several victims and afterwards relay it as if they were the legitimate router. Subsequently, the assailants were capable of monitoring the transmission of Modbus query and Modbus answer packets, therefore functioning as a packet analyzer and sniffer. Once the assailants had knowledge of the server and slave IP addresses, they proceeded to replicate the Modbus client and dispatched a Modbus command with the intention of triggering the detonation of the slave.



## **Step 2 - Design Principles.**

In order to effectively mitigate assaults, it is important to possess a comprehensive understanding of attack techniques, enabling the identification of preventive and detection measures aimed at minimizing or averting potential damage. Every act of attack starts with a systematic survey of the target area, aimed at identifying vulnerabilities and potential entrance sites. After successfully infiltrating a system, the attacker will initially engage in passive activities such as monitoring and gathering information, followed by the implementation of ARP spoofing and TCP session hijacking techniques. Once an adversary has control of the communication channel, they possess the capability to carry out a wide range of attacks, encompassing denial of service as well as the injection of malicious data or orders into the system.

Based on the data obtained throughout the evaluation stages of this framework, a comprehensive inventory of all network equipment and components has been compiled. The initial steps used for identifying ARP poisoning demonstrate a strong foundation. The occurrence of ARP spoofing resulted in the establishment of a connection between the MAC address of the attacker and the IP address of a valid machine or server within the network. By using our existing knowledge of the correspondence between IP addresses and MAC addresses, we may promptly identify network intrusions upon the detection of any discrepancies in the sent data.

Man-in-the-middle (MITM) attacks are a commonly observed form of cyber assault that allows unauthorized individuals to intercept and monitor communications between two entities. The man-in-the-middle attack is a type of cyber assault that takes place when an unauthorized individual intercepts and potentially alters communication between two valid hosts. This attack enables the attacker to surreptitiously eavesdrop on a conversation that would otherwise be inaccessible to them. The monitoring mode of a device can be exploited by an attacker to inject malicious packets into data communication streams. The malevolent packets have the potential to intermingle with legitimate data transmission streams, so creating the illusion of being an integral component of the communication process. In the typical process of packet injection, the initial step involves engaging in packet sniffing to ascertain the precise timing and methodology for generating and transmitting packets. Detecting a man-in-the-middle attack can pose challenges in the absence of appropriate measures. However, by proactively conducting investigations to ascertain whether our

communications have been intercepted, there exists a significant probability of successfully identifying such an attack. The detection of a man-in-the-middle attack may be delayed until it is too late if there is a lack of proactive monitoring. The examination of network protocols and the implementation of tamper detection mechanisms are frequently seen as crucial steps for identifying possible attacks. Nevertheless, the implementation of these processes may necessitate supplementary forensic inquiry and human resources, hence rendering the use of the ABCD framework advantageous.

In contrast to the majority of IT security breaches, attacks directed at SCADA and ICS systems primarily focus on compromising the operational processes rather than the theft or compromise of sensitive data. Although a Distributed Denial of Service (DDoS) attack targeting a web server might result in significant financial expenses and inconvenience, it is crucial to recognize that a DDoS attack directed at a SCADA system has the potential to cause catastrophic consequences, including loss of life. A distributed denial-of-service (DDoS) attack targeting a valve or subsystem has the potential to effectively incapacitate a whole facility, resulting in significant and perhaps catastrophic consequences. Communication protocols represent a significant distinction between conventional IT systems and SCADA systems. SCADA and Industrial Control Systems (ICS) are designed using traditional serial communication protocols, such as Modbus in the present context. Despite being encapsulated within the TCP/IP framework for external communication, the underlying nature of these protocols remains fundamentally sequential and uncomplicated. While Modbus was originally developed for serial networks like RS232 and RS485, it has been adapted for utilization on TCP/IP networks. The identification of SCADA assaults can be facilitated by the utilization of tools such as Wireshark or automation tools, as elaborated upon in the subsequent part of this phase. This particular tool possesses the capability to analyze the communication protocol employed within a facility, hence enabling the identification and mitigation of any security risks associated with SCADA and ICS systems.

The utilization of Wireshark enables the analysis of Modbus TCP connections, including both typical, deviant, and malevolent patterns of activity. The MITM attack exposed that the master workstation, acting as the origin, transmitted a Modbus TCP packet to the PLC workstation destination in order to trigger the activation of a single coil. In the context of the intermediate assault, the issue at hand is to the discrepancy between the MAC addresses

associated with the PLC workstation. This observation indicates the presence of a Man-in-the-Middle (MITM) attack within the network (Sanchez, 2019).

### **Step 3 - Formulate design principles.**

The vulnerability of Modbus TCP arises from the inherent limitations of Modbus TCP, which include its susceptibility to clear text communications and its absence of inherent authentication mechanisms. The identification of assaults on the Modbus TCP protocol, such as man-in-the-middle attacks, may be facilitated by the utilization of protocol analyzers. By using the Ettercap filter specifically designed for the Modbus TCP protocol, weaknesses inherent in the protocol can be effectively detected. The technique of full packet capture involves the acquisition of a comprehensive copy of network packets, encompassing both the payload and header, during their transmission over a network. This method has been employed for the purpose of observing and analyzing network data. The analysis of TCP packets has significant importance as it facilitates the detection of possible risks and harmful behavior inside a network. The detection of an attacker's approaches can be facilitated by implementing traffic filtering mechanisms that focus on specific protocols, ports, and other relevant characteristics. For example, the implementation of automatic detection rules targeting port 502, which is the default port for Modbus TCP, and the continuous monitoring of abnormal system use. Upon analysis of the TCP header, pertinent information such as the target port and protocol may be ascertained, hence facilitating the identification of a potentiality MITM attack. Denial-of-Service (DoS) assaults can be identified by the identification of certain instructions. These directives include Diagnostic Code = 4, which indicates the activation of Force Listen Only Mode, the modbus diagnostic restart communication option, which signifies the resumption of PLC communication, and modbus register uint16, which is employed for the retrieval of data from a 16-bit register. The aforementioned instructions, commonly employed in various SCADA Denial-of-Service (DoS) attacks, direct the PLC to refrain from transmitting any data to the actuators, alarms, or other interconnected PLCs. Nevertheless, the process of manually examining network data is a laborious task (Cappers et al., 2018; Holkovič et al., 2019; Letavay et al., 2019; Mirzaev, 2021; Rawat et al., 2022). This stage of the system offers an automated approach for

evaluating PCAP files with the aim of identifying particular patterns that might be indicative of network assaults.

### ***Experiment Implementations***

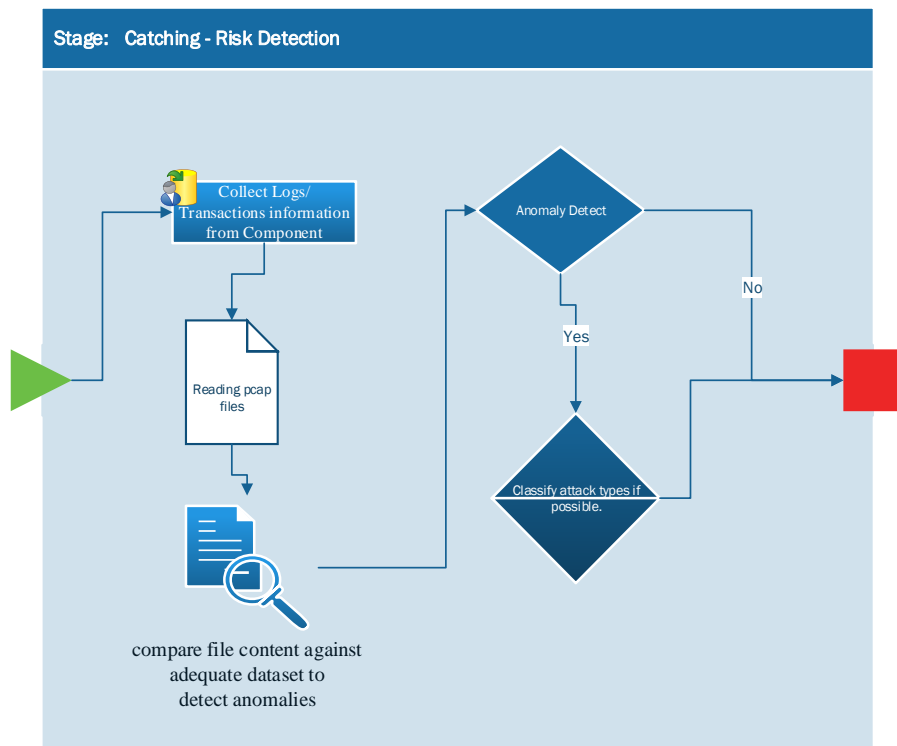
#### **Step 4 - Construction of the Framework**

The first stage of the framework involves the assessment of vulnerabilities in the components of the SCADA system. The subsequent stage of the architecture facilitates the proactive measures taken to avert and postpone such assaults. The third element of the architecture involves the identification and detection of a potential security breach within a SCADA network. During the initial phase of the framework, network administrators will have the capability to upload and input their network information. This data will then be utilized for the purpose of comparing past data with the current information available. Furthermore, the framework's concept allows real-time monitoring of TCP traffic, enabling administrators to observe network activity on any designated computer inside the network. Importantly, this continuous monitoring does not compromise the performance of the framework's components. The TCP traffic monitoring display is seen in Figure 19. During this stage, the network traffic that was gathered in phase 1 will be examined in order to identify any irregularities. The initial step of the framework encompasses an inventory comprising of Media Access Control (MAC) and Internet Protocol (IP) addresses, with log and network traffic data. This comprehensive inventory aids in the identification and detection of Address Resolution Protocol (ARP) spoofing. The presence of unregistered IP and MAC addresses is indicative of the presence of an intruder. The provision of information in the initial phase of the framework enables the establishment of this capacity. Malware may be categorized and its purpose

discerned through network traffic analysis. Figure 18 illustrates the whole flow of phase 3 inside the framework.

**Figure 25**

***Catching Phase Process Flow***



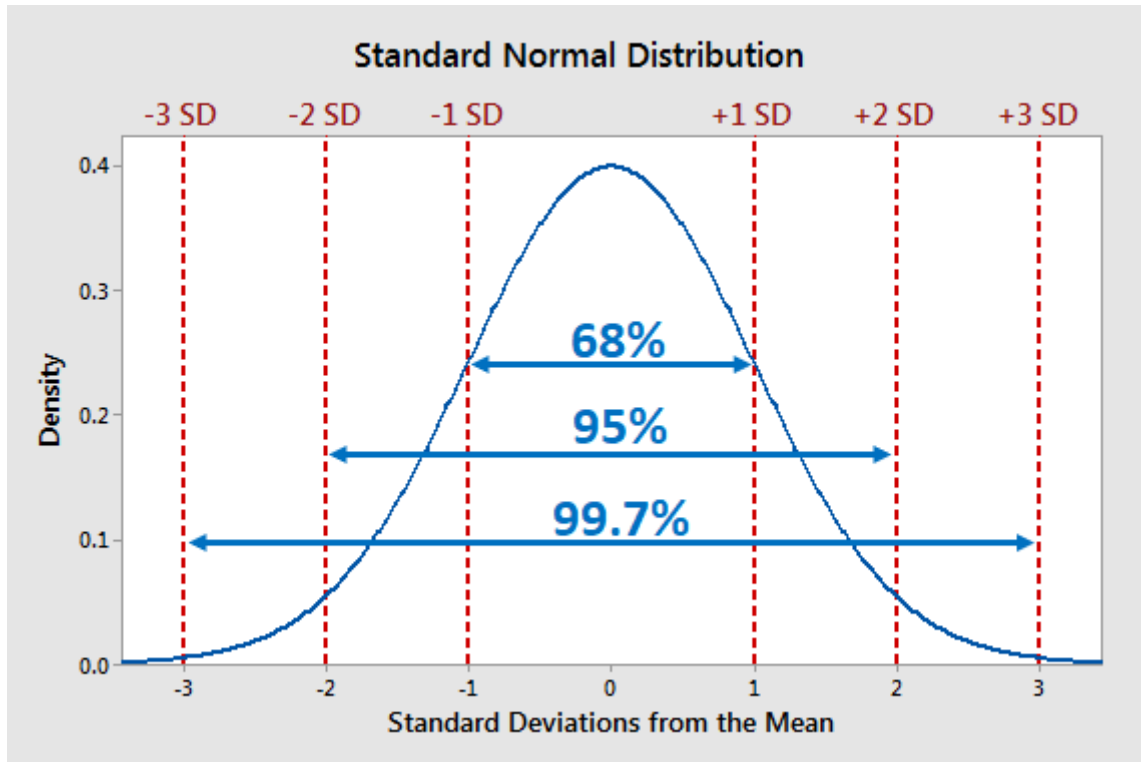
**Figure 26**  
**Network Monitoring interface.**

LocalAddress	LocalPort	ProcessId	ProcessName	RemoteAddress	RemoteHost	RemotePort	State	ServiceName	Name	Description	ExecutablePath	CommandLine	ThreadCount	VirtualSize	ParentProcessID	Handle	Caption
0.0.0.0	135	424		0.0.0.0		0	LISTENING	PfcpEptMapper	svchost.exe	svchost.exe			13	2203428304824	776	424	svchost.exe
0.0.0.0	445	4		0.0.0.0		0	LISTENING	System	System	System			203	10955392	0	4	System
0.0.0.0	1326	776		0.0.0.0		0	LISTENING	services.exe	services.exe	services.exe			10	2203389425904	696	776	services.exe
0.0.0.0	2179	2772		0.0.0.0		0	LISTENING	vmsas	vmsas.exe	vmsas.exe			12	2203478192128	776	2772	vmsas.exe
0.0.0.0	5040	7556		0.0.0.0		0	LISTENING	CDPSvc	svchost.exe	svchost.exe			13	2203455885312	776	7556	svchost.exe
0.0.0.0	7680	10872		0.0.0.0		0	LISTENING	DsSvc	svchost.exe	svchost.exe			9	2203446839648	776	10872	svchost.exe
0.0.0.0	49152	4		0.0.0.0		0	LISTENING	System	System	System			203	10955392	0	4	System
0.0.0.0	49153	4		0.0.0.0		0	LISTENING	System	System	System			203	10955392	0	4	System
0.0.0.0	49154	4		0.0.0.0		0	LISTENING	System	System	System			203	10955392	0	4	System
0.0.0.0	49155	4		0.0.0.0		0	LISTENING	System	System	System			203	10955392	0	4	System
0.0.0.0	49664	804		0.0.0.0		0	LISTENING	KeyIso	lsass.exe	lsass.exe			8	2203427913728	696	804	lsass.exe
0.0.0.0	49665	696		0.0.0.0		0	LISTENING	wscnt.exe	wscnt.exe	wscnt.exe			1	2203388395520	552	696	wscnt.exe
0.0.0.0	49666	1784		0.0.0.0		0	LISTENING	EventLog	svchost.exe	svchost.exe			6	2203430469532	776	1784	svchost.exe
0.0.0.0	49667	1560		0.0.0.0		0	LISTENING	Schedule	svchost.exe	svchost.exe			6	2203422982144	776	1560	svchost.exe
0.0.0.0	49668	2632		0.0.0.0		0	LISTENING	SessionSrv	svchost.exe	svchost.exe			6	2203401752676	776	2632	svchost.exe
0.0.0.0	49669	3912		0.0.0.0		0	LISTENING	Spooler	spoolsv.exe	spoolsv.exe			8	2203472674816	776	3912	spoolsv.exe
127.0.0.1	1325	4644		0.0.0.0		0	LISTENING	MSSQLSSEX	edvrsvr.exe	edvrsvr.exe			66	22145750088	776	4644	edvrsvr.exe
127.0.0.1	354	4116		0.0.0.0		0	LISTENING	Borgovr Service	mDNSResponder	mDNSResponder			2	33626964	776	4116	mDNSResponder
127.0.0.1	7693	16544	DESKTOP-CGCA	127.0.0.1	7693	ESTABLISHED		Reflex.exe	Reflex.exe	C:\Program Files\...	C:\Program Files\...		71	2218182007958	16276	16544	Reflex.exe
127.0.0.1	7695	16544	DESKTOP-CGCA	127.0.0.1	7698	ESTABLISHED		Reflex.exe	Reflex.exe	C:\Program Files\...	C:\Program Files\...		71	2218182007958	16276	16544	Reflex.exe
127.0.0.1	7670	12632	DESKTOP-CGCA	127.0.0.1	7671	ESTABLISHED		Reflex.exe	Reflex.exe	C:\Program Files\...	C:\Program Files\...		5	2203548599652	16544	12632	Reflex.exe
127.0.0.1	7671	12632	DESKTOP-CGCA	127.0.0.1	7670	ESTABLISHED		Reflex.exe	Reflex.exe	C:\Program Files\...	C:\Program Files\...		5	2203548599652	16544	12632	Reflex.exe
127.0.0.1	9468	10796		0.0.0.0		0	LISTENING	devervr.exe	devervr.exe	C:\Program Files\...	C:\Program Files\...		54	826584832	10964	10796	devervr.exe
127.0.0.1	9469	7544		0.0.0.0		0	LISTENING	Microsoft Ais Sh...	Microsoft Ais Sh...	C:\Windows\Mic...	C:\Windows\M...		11	330027008	10796	7544	Microsoft Ais Sh...
127.0.0.1	9469	7544	DESKTOP-CGCA	127.0.0.1	9468	ESTABLISHED		Microsoft Ais Sh...	Microsoft Ais Sh...	C:\Windows\Mic...	C:\Windows\M...		11	330027008	10796	7544	Microsoft Ais Sh...
127.0.0.1	9840	17932		0.0.0.0		0	LISTENING	devervr.exe	devervr.exe	C:\Program Files\...	C:\Program Files\...		42	844902400	10964	17932	devervr.exe
127.0.0.1	9840	17932	DESKTOP-CGCA	127.0.0.1	9842	CLOSE_WAIT		devervr.exe	devervr.exe	C:\Program Files\...	C:\Program Files\...		42	844902400	10964	17932	devervr.exe
127.0.0.1	9841	17968		0.0.0.0		0	LISTENING	Microsoft Ais Sh...	Microsoft Ais Sh...	C:\Windows\Mic...	C:\Windows\M...		11	318956072	17932	17968	Microsoft Ais Sh...
127.0.0.1	9842	17968	DESKTOP-CGCA	127.0.0.1	9840	FIN_WAIT2		Microsoft Ais Sh...	Microsoft Ais Sh...	C:\Windows\Mic...	C:\Windows\M...		11	318956072	17932	17968	Microsoft Ais Sh...
127.0.0.1	9843	0	DESKTOP-CGCA	127.0.0.1	9841	TIME_WAIT		System Idle Proc...	System Idle Proc...				4	8192	0	0	System Idle Proc...
127.0.0.1	9844	0	DESKTOP-CGCA	127.0.0.1	9841	TIME_WAIT		System Idle Proc...	System Idle Proc...				4	8192	0	0	System Idle Proc...
127.0.0.1	9848	10796	DESKTOP-CGCA	127.0.0.1	9469	ESTABLISHED		devervr.exe	devervr.exe	C:\Program Files\...	C:\Program Files\...		54	826584832	10964	10796	devervr.exe
172.17.176.1	139	4		0.0.0.0		0	LISTENING	System	System	System			203	10955392	0	4	System
192.168.6.121	139	4		0.0.0.0		0	LISTENING	System	System	System			203	10955392	0	4	System
192.168.6.121	7844	4496		13.64.180.106		443	ESTABLISHED	WpnService	svchost.exe	svchost.exe			7	220343457920	776	4496	svchost.exe
192.168.6.121	8170	17172		194.82.12.179	a104-82-12-179	443	CLOSE_WAIT	SearchApp.exe	SearchApp.exe	C:\Windows\Sys...	C:\Windows\Sy...		32	2236728142848	936	17172	SearchApp.exe
192.168.6.121	8903	13684		50.96.64.194		443	ESTABLISHED	OUTLOOK.EXE	OUTLOOK.EXE	C:\Program Files\...	C:\Program Files\...		48	1891291648	10964	13684	OUTLOOK.EXE
192.168.6.121	8902	13684		50.96.64.194		443	ESTABLISHED	OUTLOOK.EXE	OUTLOOK.EXE	C:\Program Files\...	C:\Program Files\...		48	1891291648	10964	13684	OUTLOOK.EXE
192.168.6.121	9207	13968		52.109.2.238		443	ESTABLISHED	WINWORD.EXE	WINWORD.EXE	C:\Program Files\...	C:\Program Files\...		47	1242079232	10964	13968	WINWORD.EXE

**Calculation.**

The identification of abnormalities within a dataset is a critical task in the detection of system intrusions. The proposed solution is proactive in nature, as it has the ability to identify and address any issues before they result in any detrimental impact on the system. The utilization of standard deviation will be employed in this investigation for the purpose of identifying abnormalities. This study proposed employing a pristine dataset for the computation of the standard deviations of the system. Subsequently, the upper and lower bounds of the pristine dataset were utilized to ascertain the upper and lower limits of the system's standard deviation. A data point is said to be within the normal range if it falls between the upper and lower bounds defined by the standard deviation. Any data point that falls outside of these established limitations is considered to be anomalous.

Figure 27

*Diagram of the 68-95-99.7% Rule***Equation 1***Standard Deviation Calculation*

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

According to the empirical rule, in the case of data that follows a normal distribution, about 99.7% of the data points will be within a range of three standard deviations from the mean (Ross, 2009). This rule is sometimes referred to as the "three-sigma rule" and the "68-95-99.7 rule." If the distribution is approximately normal, it may be observed that 68% of the data points are within one standard deviation of the mean, 95% lie within two standard deviations, and 99.7% lie within three standard deviations. The identified observation is an exceptional finding that challenges established beliefs. An outlier refers to a data point that

deviates significantly from the remaining observations within a dataset. The empirical rule is elucidated in the subsequent algorithm.

$$\mu \pm m\sigma$$

where  $\mu$  represents the mean,  $\sigma$  represents the standard deviation, and  $m$  represents the number of standard deviations.

68% of the data lies within -1 and +1 standard deviation of the mean.  $\mu \pm 1\sigma$

95% of the data lies between -2 and +2 standard deviations from the mean.  $\mu \pm 2\sigma$

99.7% of the data lies between -3 and +3 standard deviations from the mean.  $\mu \pm 3\sigma$

Inputting the mean and standard deviation into the empirical rule calculator as show below will generate the intervals automatically.

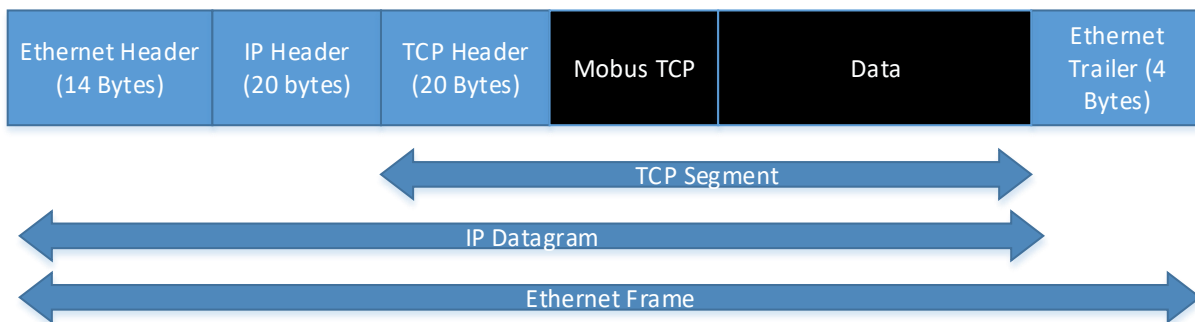
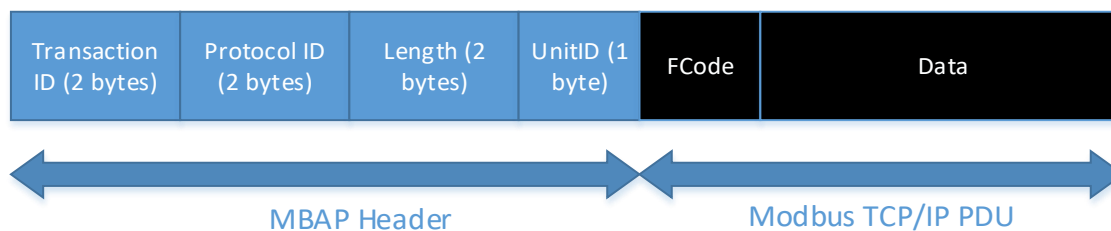
## **Equation 2**

### ***Empirical rule***

$$z = \frac{x - \mu}{\sigma}$$

Ethernet enables the TCP/IP protocol stack to incorporate the Modbus TCP protocol, which was formerly implemented via serial connections. This integration involves the process of unwrapping the various layers until the Modbus TCP header is accessed. The image below illustrates the Modbus TCP header and its corresponding data. The composition of the Modbus message consists of two main components: the Modbus Application Header (MBAP) and the Protocol Data Unit (PDU) (Modbus, 2012). The main aim of this phase is to conduct an analysis of the PCAP header with the goal of identifying anonymous connections. Additionally, it involves extracting the Modbus TCP for the purpose of studying the intended usage of the package. Furthermore, it entails quantifying the connections associated with normal-behavior traffic and comparing this count to the present non-baseline event.



**Figure 28*****Modbus TCP inside Ethernet Frame*****Figure 29*****MBAP Header and Modbus TCP/IP PDU*****Step 5 – Implementation and experimentation**

ARP poisoning refers to a type of man-in-the-middle attack that has the capability to disrupt, alter, or intercept the flow of network communication. The aforementioned approach is commonly utilized for the purpose of initiating additional offensive actions, such as session hijacking or denial-of-service attacks. There exist several procedures for the identification of ARP poisoning. Examining the packet capture (PCAP) file is a prevalent approach for identifying instances of ARP poisoning attacks. Packet capture is a networking methodology that entails the interception of data packets as they traverse a network. PCAP files refer to data files that are created by computer programs. The aforementioned files consist of network packet data and are employed for the purpose of studying network properties. A PCAP can be created by employing a network analyzer or packet capture program such as Wireshark or TCPDump. The PCAP file contains both TCP/IP and UDP broadcasts. The provided information can be utilized for the purpose of identifying ARP poisoning, as the header of the TCP package encompasses the MAC address and IP address of both the source and target of such package. If the packet capture (PCAP) file includes two separate Internet Protocol (IP) addresses that are associated with the same Media Access Control (MAC) address, it suggests

the presence of an Address Resolution Protocol (ARP) poisoning attack on the network. The act of counting packets is a supplementary technique that can enhance the efficacy of PCAP analysis in the identification of network assaults. IP addresses that exhibit a high volume of connection requests within a short timeframe may elicit suspicion. The identification of these abnormalities will facilitate the early detection of a Distributed Denial of Service (DDoS) assault prior to its escalation to maximum intensity. The frequency of connections for each IP address is computed in order to identify IP addresses that exhibit an excessive number of connections. The results are presented in both table and graph formats below. In order to ascertain the number of excessive connections, the baseline idea will be employed, utilizing the pristine file from this experiment as the baseline. The clean file will be analyzed in order to calculate the tally of regular network traffic. The forthcoming network package TCP transactions will be systematically quantified and juxtaposed against the established baseline. Any departure from the expected distribution of data will be regarded as an anomaly, hence suggesting the occurrence of a Distributed Denial of Service (DDoS) assault.

This paper proposes the implementation of a monitoring system for TCP packets, followed by a comparative analysis of the packets, in order to identify and mitigate attacks in their early stages, hence preventing their further propagation. The experimental procedure to be conducted in this phase will commence with the transfer of files from the initial phase of the framework. The subsequent illustration showcases the graphical user interface (GUI) designed for the purpose of uploading the network traffic file.

**Figure 30***Interface to upload network traffic information.*

The screenshot shows a software application window titled "The ABCD Risk Management Modeling". The interface includes a menu bar with options: "Pcap Analyzing", "Display", "Component", "Model", "Models And Components", "Component Relationships", "Categories", and "Component Groups". The main area contains a form with the following fields and controls:

- ID :** A text input field.
- Name :** A text input field.
- Description :** A large text area with a vertical scrollbar.
- File Path :** A text input field with a "Browse" button and an "Analyzing?" checkbox.
- Program Path :** A text input field with a "Browse" button.
- Text :** A large text area with a vertical scrollbar.

At the bottom of the form are four buttons: "Save", "Add", "Search", and "Clear". Below the form is a table with the following columns: "ID", "Name", "Description", "Analyze", "FilePath", and "Progr". The table is currently empty.

The Kaitai Struct library facilitates the interpretation of PCAP files. Kaitai Struct is a declarative programming language utilized for the purpose of describing binary data structures found in files or memory, including binary file formats, network stream packet formats, and other similar formats. (*Kaitai Struct: declarative binary format parsing language*, 2015). The module will encompass automatically produced code for a parser that is capable of reading the specified data structure from a file or stream, hence facilitating API access to the data. The PCAP's content is thereafter subjected to analysis by the model within the framework. The specified byte length is used to extract several pieces of information, including packet number, time, source IP, destination IP, source port, destination port, protocol, message length, source MAC address, destination MAC address, and protocol ID.

This process is illustrated in Figures 31 and 32. For each network packet included within the PCAP, the following information will be collected.

**Figure 31***Information to be captured by the system.*

```

dt.Columns.Add("PacketNo", typeof(string));
dt.Columns.Add("Time", typeof(string));
dt.Columns.Add("SourceIP", typeof(string));
dt.Columns.Add("DestinationIP", typeof(string));
dt.Columns.Add("SourcePort", typeof(string));
dt.Columns.Add("DestinationPort", typeof(string));
dt.Columns.Add("Protocol", typeof(string));
dt.Columns.Add("Length", typeof(string));
dt.Columns.Add("SourceMacAddress", typeof(string));
dt.Columns.Add("DestinationMacAddress", typeof(string));
dt.Columns.Add("OrigLen", typeof(string));
dt.Columns.Add("TsSec", typeof(string));
dt.Columns.Add("TsUsec", typeof(string));
dt.Columns.Add("EtherType", typeof(string));
dt.Columns.Add("Data", System.Type.GetType("System.Byte[]"));
dt.Columns.Add("FileName", typeof(string));
dt.Columns.Add("Transaction_ID", typeof(int));
dt.Columns.Add("Protocol_ID", typeof(int));
dt.Columns.Add("Len", typeof(int));
dt.Columns.Add("Unit_ID", typeof(int));
dt.Columns.Add("Function_Code", typeof(int));
dt.Columns.Add("Reference_Number", typeof(int));
dt.Columns.Add("Modbus_PDU", System.Type.GetType("System.Byte[]"));
dt.Columns.Add("AckNum", typeof(string));
dt.Columns.Add("SeqNum", typeof(string));
dt.Columns.Add("WindowSize", typeof(string));
dt.Columns.Add("URG", typeof(bool));
dt.Columns.Add("ACK", typeof(bool));
dt.Columns.Add("PSH", typeof(bool));
dt.Columns.Add("RST", typeof(bool));
dt.Columns.Add("SYN", typeof(bool));
dt.Columns.Add("FIN", typeof(bool));

```

The code below will be utilized to parse the data.

**Figure 32**

*C# code to parse data.*

```

TcpSegment ts = (TcpSegment)ipv4.Body.Body;
DstPort = ts.DstPort.ToString();
SrcPort = ts.SrcPort.ToString();
AckNum = ts.AckNum.ToString();
SeqNum = ts.SeqNum.ToString();
WindowSize = ts.WindowSize.ToString();
row["Length"] = ts.Body.Length;

byte[] tcp = ipv4.M_RawBody;
try
{
    var bits = new BitArray(tcp);
    URG = bits[106];
    ACK = bits[107];
    PSH = bits[108];
    RST = bits[109];
    SYN = bits[110];
    FIN = bits[111];

    if (ts.Body.Length >= 9)
    {
        byte[] bytes = ts.Body;
        byte[] Transaction_ID = new byte[2];
        byte[] Protocol_ID = new byte[2];
        byte[] Length = new byte[2];
        byte[] Unit_ID = new byte[1];
        byte[] Function_Code = new byte[1];
        Buffer.BlockCopy(bytes, 0, Transaction_ID, 0, 2);
        Buffer.BlockCopy(bytes, 2, Protocol_ID, 0, 2);
        Buffer.BlockCopy(bytes, 4, Length, 0, 2);
        Buffer.BlockCopy(bytes, 6, Unit_ID, 0, 1);
        if (BitConverter.IsLittleEndian)
        {
            Array.Reverse(Length);
            Array.Reverse(Transaction_ID);
            Array.Reverse(Protocol_ID);
        }
        int i1 = BitConverter.ToInt16(Length, 0);
        int Transaction = BitConverter.ToInt16(Transaction_ID, 0);
        int Protocol = BitConverter.ToInt16(Protocol_ID, 0);
        Buffer.BlockCopy(bytes, 7, Function_Code, 0, 1);
        byte[] Reference_Number = new byte[2];
        Buffer.BlockCopy(bytes, 8, Reference_Number, 0, 2);
        if (BitConverter.IsLittleEndian)
        {
            Array.Reverse(Reference_Number);
        }
        int reference = BitConverter.ToInt16(Reference_Number, 0);
        int Unit = Convert.ToInt32(Unit_ID[0]);
        int FCode = Convert.ToInt32(Function_Code[0]);
        byte[] Modbus_PDU = new byte[i1 - 4];
        Buffer.BlockCopy(bytes, 10, Modbus_PDU, 0, (i1 - 4));
    }
}

```

Subsequently, the collected data will be imported into the designated database and saved within the table as depicted in the following.

**Figure 33**

*PCAP data stored in a database table.*

PacketNo	Time	SourceIP	Destination	SourcePort	DestinationPort	Protocol	Length	SourceMac	DestinationMac	OrigLen	TsSec	TsUsec	EtherType	Data	FileName	Transaction
17415	1	172.27.224.50	172.27.224.50	502	53762	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	654024	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17416	1	172.27.224.50	172.27.224.50	502	53760	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	654076	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17417	1	172.27.224.50	172.27.224.50	502	53760	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	654239	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17418	1	172.27.224.50	172.27.224.50	502	53758	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	654290	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17419	1	172.27.224.50	172.27.224.50	53758	502	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	654443	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17420	1	172.27.224.50	172.27.224.50	502	53756	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	654493	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17421	1	172.27.224.50	172.27.224.50	53756	502	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	654707	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17422	1	172.27.224.50	172.27.224.50	53770	502	Tcp	12	00-80-F4-09...	00-80-F4-09...	66	1526987267	654756	Ipv4	<Binary dat...	C:\Users\M\Downl...	1161
17423	1	172.27.224.50	172.27.224.50	502	53754	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	661476	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17424	1	172.27.224.50	172.27.224.50	53754	502	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	663191	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17425	1	172.27.224.50	172.27.224.50	502	53752	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	663267	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17426	1	172.27.224.50	172.27.224.50	53752	502	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	663448	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17427	1	172.27.224.50	172.27.224.50	502	53750	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	663510	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17428	1	172.27.224.50	172.27.224.50	502	53750	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	663768	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17429	1	172.27.224.50	172.27.224.50	502	53748	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	663828	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17430	1	172.27.224.50	172.27.224.50	53748	502	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	664000	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17431	1	172.27.224.50	172.27.224.50	502	53746	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	664063	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17432	1	172.27.224.50	172.27.224.50	53746	502	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	664297	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17433	1	172.27.224.50	172.27.224.50	502	53744	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	664459	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17434	1	172.27.224.50	172.27.224.50	53744	502	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	664599	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17435	1	172.27.224.50	172.27.224.50	502	53742	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	664722	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17436	1	172.27.224.50	172.27.224.50	53742	502	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	664782	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17437	1	172.27.224.50	172.27.224.50	53778	502	Tcp	12	00-80-F4-09...	00-80-F4-09...	66	1526987267	666127	Ipv4	<Binary dat...	C:\Users\M\Downl...	1160
17438	1	172.27.224.50	172.27.224.50	502	53740	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	673219	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17439	1	172.27.224.50	172.27.224.50	53740	502	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	673308	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17440	1	172.27.224.50	172.27.224.50	502	53738	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	673471	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17441	1	172.27.224.50	172.27.224.50	53738	502	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	673528	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17442	1	172.27.224.50	172.27.224.50	502	53736	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	673920	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17443	1	172.27.224.50	172.27.224.50	53736	502	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	673975	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17444	1	172.27.224.50	172.27.224.50	502	53734	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	674152	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL
17445	1	172.27.224.50	172.27.224.50	53734	502	Tcp	0	00-80-F4-09...	00-80-F4-09...	60	1526987267	674307	Ipv4	<Binary dat...	C:\Users\M\Downl...	NULL

The stored procedure is a database object that contains a set of SQL statements and procedural logic, which may be executed as a single unit. The AnalyzePCAPFile function will be employed to analyze the PCAP data. The entire source code of the stored method may be seen in Appendix B. Presented below are three snippets of the SQL script extracted from the stored process. The script employs a method of anomaly detection by utilizing pre-existing IP addresses and MAC addresses. Both the pristine dataset and the testing dataset were subjected to data collection of IP and MAC addresses. The subsequent step involves doing a comparison between these two datasets in order to detect any aberrant patterns or behavior.

Figure 34

**SQL Script to detect ARP poison.**

```

/*****
1) Detect anomaly using known IPs and MAC addresses
--The assessment phase of the framework and the existence of a pristine dataset both provide recognizable data.
--Comparison of testing dataset data to known data in order to detect anomalous activities
*****/
IF (OBJECT_ID('tempdb..#unidentifyConnection') IS NOT NULL) DROP TABLE #unidentifyConnection;
with A as (
  select distinct MacAddress-SourceMacAddress,IP-SourceIP
  from [FileInformation]
  where filename in (select top 1 OrgFileName from #tmp where FileType='Clean')
  UNION
  select distinct DestinationMacAddress, DestinationIP
  from [FileInformation]
  where filename in (select top 1 OrgFileName from #tmp where FileType='Clean')
), B AS (
  select distinct MacAddress-SourceMacAddress,IP-SourceIP
  from [FileInformation]
  where filename in (select top 1 OrgFileName from #tmp where FileType<>'Clean')
  UNION
  select distinct DestinationMacAddress, DestinationIP
  from [FileInformation]
  where filename in (select top 1 OrgFileName from #tmp where FileType<>'Clean')
), C AS
(
  select B.MacAddress,B.IP
  from A RIGHT JOIN B on A.MacAddress=B.MacAddress and isnull(A.IP,'')=isnull(B.IP,'')
  where A.IP is null
), D AS
(
  select C.MacAddress,C.IP, TsSec=((cast(TsSec as bigint)-(select min(cast(TsSec as bigint)) from [dbo].[FileInformation] where filename=A.FileName))/@Interval)
  from [FileInformation] A INNER JOIN C on (A.DestinationMacAddress=C.MacAddress and isnull(A.DestinationIP,'')=isnull(C.IP,'')) or (A.SourceMacAddress=C.MacAddress and isnull(A.SourceIP,'')=isnull(C.IP,''))
  where filename in (select top 1 OrgFileName from #tmp where FileType<>'Clean')
)
select MacAddress,IP, TsSec=MIN(TsSec)
INTO #unidentifyConnection
FROM D
group by MacAddress,IP;

update #tmp
set MacAddressIP='MAC:'+A.MacAddress+' ==> IP:'+A.IP
from #unidentifyConnection A
WHERE #tmp.TsSec=A.TsSec and FileType<>'Clean';

```

Pulling source and destination MAC and IP addresses from a pristine file.

Pulling source and destination MAC and IP addresses from a file to be inspected.

Compare the MAC and IP addresses from both files.

The anticipated result of conducting a comparison between IP and MAC addresses will align with the following description.

Figure 35

**Example of ARP poison output formatted as a table.**

	FileType	FileName	TsSec	NumberOfTransaction	OrgFileName	MacAddressIP
1	mitm	eth2dump-mitm-change-5m-0.5h_1.pcap	60	110	C:\Users\IM\Downloads\captures1_...	MAC:00-0C-29-E6-14-0D ==> IP:172.27.224.70

The script uses the statistical measure of standard deviation in order to identify and detect irregularities. The tally of transactions is recorded for both the original dataset and the dataset used for testing purposes. The mean transaction count from the pristine dataset is utilized to compute the standard deviation of the testing dataset. This calculation is performed



in order to ascertain the number of standard deviations that the count corresponds to. Any departure from the empirical rule is considered an aberration.

**Figure 36**

**SQL Script to detect DDoS Attack**

```

/*****
2) Detect anomaly using standard deviation
--Calculate SQL standard deviation
--Calculating the Mean or Average ----> Mean = Sum of each individual/Total number of items
--Calculating the Statistical Variance ----> Variance = ((OriginalValue - Mean)^2 + (OriginalValue - Mean)^2 + ....)/(Total number of items - 1)
--Calculating Standard Deviation ----> Standard Deviation = Square root (Variance)
*****/

SELECT *,STD_Check_Min=case when FileType<>'Clean' and NumberOfTransaction <(@Manual_std * (@MinNormalNumberOfSTD-1)) then '0' else '1' end,
STD_Check_Max=case when FileType<>'Clean' and NumberOfTransaction >(@Manual_std * (@MaxNormalNumberOfSTD+1)) then '0' else '1' end,
DetectUnIdentityConnection=case when FileType<>'Clean' and MacAddressIP is not null then '0' else '1' end
FROM #tmp A
order by TsSec;

SELECT @BuiltIn_std=STDEV(NumberOfTransaction),@Manual_std=Sqrt(Sum(Power(((cast(NumberOfTransaction as float)- A.mean)),2) / A.ct))
FROM #tmp;
(SELECT avg(NumberOfTransaction) as mean, COUNT(NumberOfTransaction) as ct FROM #tmp where FileType='Clean') as A
where FileType='Clean';

select @MaxNormalNumberOfSTD=ceiling(max(NumberOfTransaction)/@BuiltIn_std),@MinNormalNumberOfSTD=Floor(min(NumberOfTransaction)/@BuiltIn_std)
from #tmp
where FileType='Clean';

```

The anticipated outcomes of data acquisition and comparison exhibit similarities to the illustrated data table.

**Figure 37**

**Example of DDoS attack output formatted as a table.**

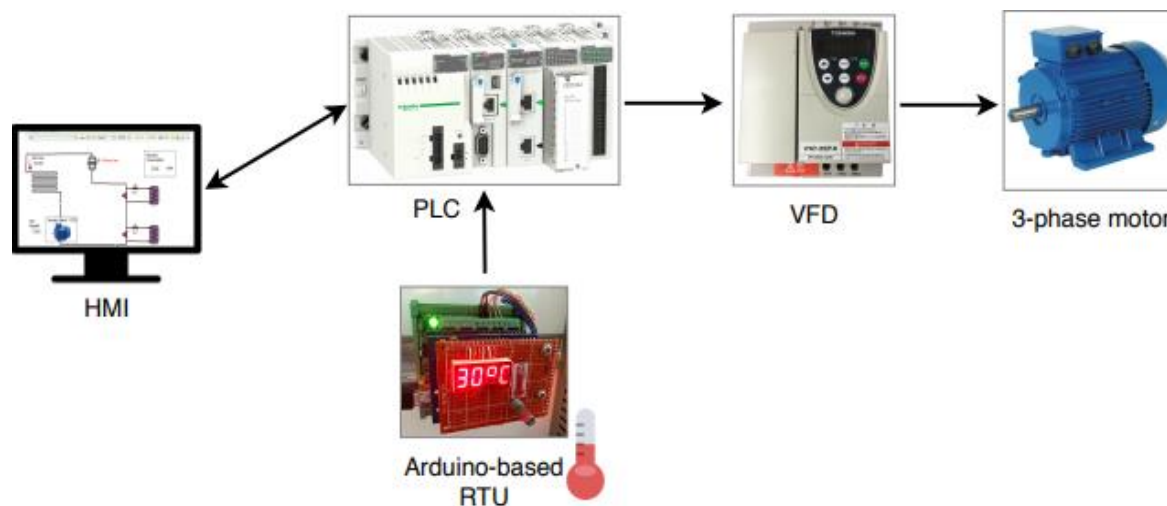
File Type	File Name	TsSec	NumberOfTransaction	OrgFileName	STD_Check_Min	STD_Check_Max	
708	QueryFlooding	eth2dump-modbusQueryFlooding1m-0.5h_1.pcap	353	35	C:\Users\IM\Downloads\captures1_v2\captures1_v2\m...	1	1
709	Clean	eth2dump-clean-0.5h_1.pcap	354	105	C:\Users\IM\Downloads\captures1_v2\captures1_v2\cle...	1	1
710	QueryFlooding	eth2dump-modbusQueryFlooding1m-0.5h_1.pcap	354	57	C:\Users\IM\Downloads\captures1_v2\captures1_v2\m...	1	1
711	Clean	eth2dump-clean-0.5h_1.pcap	355	99	C:\Users\IM\Downloads\captures1_v2\captures1_v2\cle...	1	1
712	QueryFlooding	eth2dump-modbusQueryFlooding1m-0.5h_1.pcap	355	50	C:\Users\IM\Downloads\captures1_v2\captures1_v2\m...	1	1
713	Clean	eth2dump-clean-0.5h_1.pcap	356	107	C:\Users\IM\Downloads\captures1_v2\captures1_v2\cle...	1	1
714	QueryFlooding	eth2dump-modbusQueryFlooding1m-0.5h_1.pcap	356	36	C:\Users\IM\Downloads\captures1_v2\captures1_v2\m...	1	1
715	Clean	eth2dump-clean-0.5h_1.pcap	357	84	C:\Users\IM\Downloads\captures1_v2\captures1_v2\cle...	1	1
716	QueryFlooding	eth2dump-modbusQueryFlooding1m-0.5h_1.pcap	357	56	C:\Users\IM\Downloads\captures1_v2\captures1_v2\m...	1	1
717	QueryFlooding	eth2dump-modbusQueryFlooding1m-0.5h_1.pcap	358	51	C:\Users\IM\Downloads\captures1_v2\captures1_v2\m...	1	1
718	Clean	eth2dump-clean-0.5h_1.pcap	358	105	C:\Users\IM\Downloads\captures1_v2\captures1_v2\cle...	1	1
719	Clean	eth2dump-clean-0.5h_1.pcap	359	98	C:\Users\IM\Downloads\captures1_v2\captures1_v2\cle...	1	1
720	QueryFlooding	eth2dump-modbusQueryFlooding1m-0.5h_1.pcap	359	56	C:\Users\IM\Downloads\captures1_v2\captures1_v2\m...	1	1

To evaluate the proposed approach, a machine learning model is constructed utilizing the publicly available dataset named "Cyber-security Modbus ICS dataset." This dataset was curated by Ivo Frazo, Pedro Abreu, Tiago Cruz, and Helder Arajo, (Frazão et al., 2018). The datasets utilized in this study were obtained through the utilization of MODBUS/TCP equipment within a small-scale process automation setting. The purpose of this research was to investigate the application of machine learning methodologies in enhancing cybersecurity measures within industrial control systems. The experimental setup comprises an electric motor that emulates the functionality of a liquid pump, alongside a PLC. Furthermore, the PLC establishes a connection with the HMI responsible for system control, as seen in Figure 31. The dataset comprises several files encompassing Modbus/TCP communication,

encompassing both routine operations and distributed denial-of-service (DDoS) assaults. There are several captures contained in the dataset. The initial data package, referred to as capture1, comprises the collected traces for the following scenarios in sequential order: nominal state, ARP-based attacks, Modbus inquiry flooding, ICMP flooding, and TCP SYN flooding. The obtained traces for Modbus inquiry flooding, ICMP flooding, and TCP SYN flooding are contained inside the second and third data packages, namely capture2 and capture3. The aforementioned datasets will serve as the test dataset for assessing the efficacy of the ABCD framework in identifying potential risks from intercepted network traffic.

**Figure 38**

*Test Dataset's component Structure.*



(Frazão et al., 2018)

### ***Experiment Results***

The model for analytical purposes included many types of cyber-security assaults on Modbus ICS dataset SCADA systems. The attacks encompass TCP Syn Flood DDoS, ping flood DDoS, Modbus query, and MITM. The picture below displays the names of datasets together with their corresponding package counts.

**Figure 39*****Experiment datasets.***

	FileName	Packet_Count
1	C:\Users\IM\Downloads\captures1_v2\captures1_v2\mitm\eth2dump-mitm-change-5m-0.5h_1.pcap	35430
2	C:\Users\IM\Downloads\captures1_v2\captures1_v2\pingFloodDDoS\eth2dump-pingFloodDDoS1m-0.5h_1.pcap	29692
3	C:\Users\IM\Downloads\captures1_v2\captures1_v2\clean\eth2dump-clean-0.5h_1.pcap	35430
4	C:\Users\IM\Downloads\captures1_v2\captures1_v2\modbusQueryFlooding\eth2dump-modbusQueryFlooding1m-0.5h_1.pcap	59213
5	C:\Users\IM\Downloads\captures1_v2\captures1_v2\modbusQuery2Flooding\eth2dump-modbusQuery2Flooding1m-0.5h_1.pcap	73563
6	C:\Users\IM\Downloads\captures1_v2\tcpSYNFloodDDoS\eth2dump-tcpSYNFloodDDoS1m-0.5h_1.pcap	45271

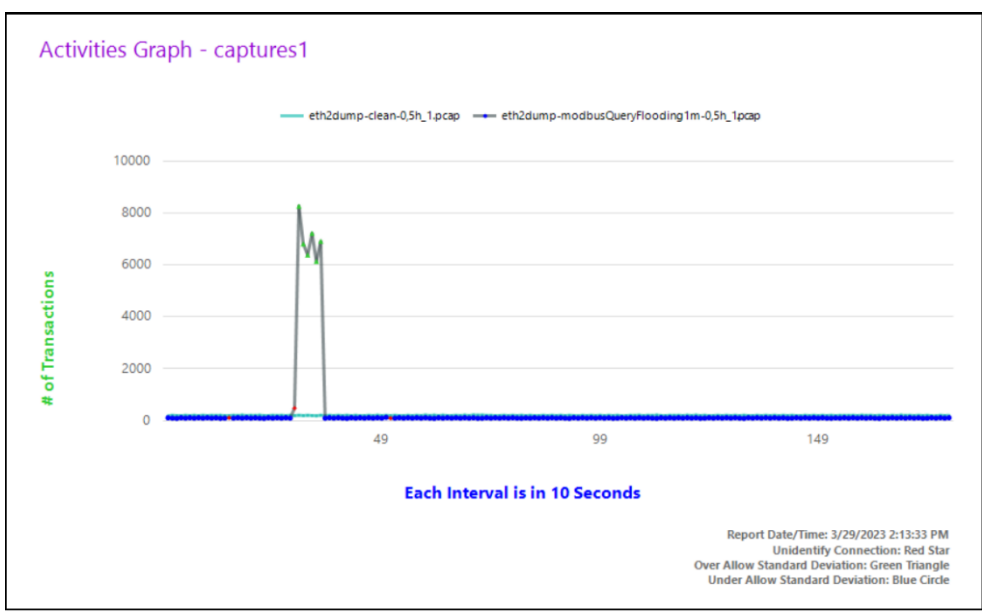
During the earliest stages of its attack, malware often induces a significant increase in network activity. In order to identify and detect malicious software, it is necessary to establish a reference point that represents typical and expected behavior. Once a thorough understanding of typical phenomena, such as traffic patterns, flow, and behavior, has been acquired, these foundational measures may be utilized to create alerts or warnings that warrant additional examination. The use of baselines significantly enhances the probability of security analysts detecting instances of traffic spikes. One of the datasets in question exhibits a state of purity, whereas the other datasets are compromised by the presence of malicious traffic, often associated with various forms of attacks. The unaltered file will serve as a reference point for our experiment. The present inquiry will utilize the suggested reasoning and formula in order to detect malevolent conduct inside the datasets.

**Modbus Query Flood.**

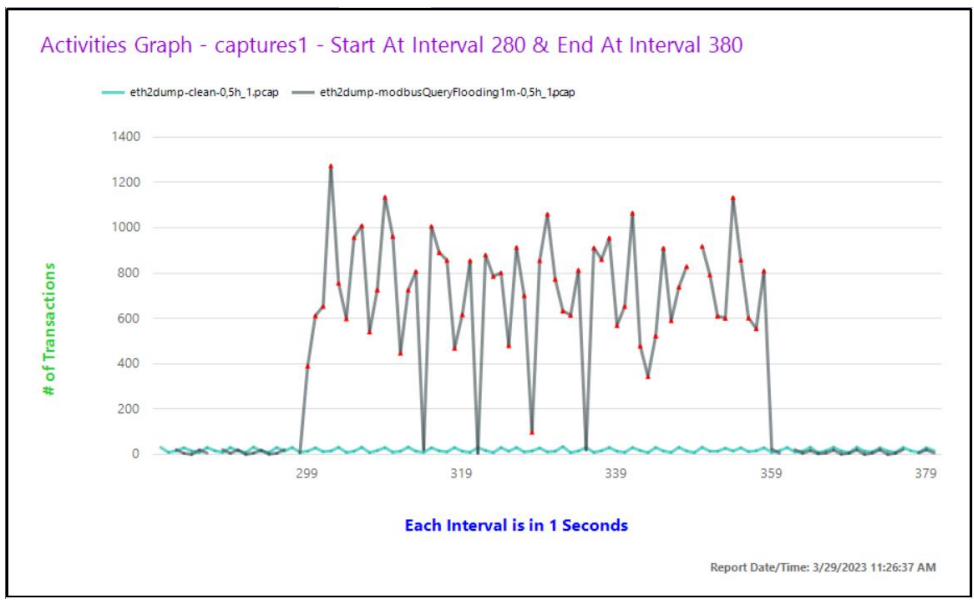
This study aims to do a comparative analysis between the query flood dataset and the pristine dataset. According to Figure 40, there is a notable rise in the number of transactions seen within the time interval of 250 to 400 seconds. Furthermore, an unidentifiable link is shown by a red dot, manifesting itself prior to the occurrence of the assault and coinciding with a surge in network activity. Figure 34 provides a comprehensive examination of the inundation assault. The presence of the initial red dot signifies the occurrence of ARP

poisoning within the system, while the succeeding red dot signifies the initiation of Distributed Denial of Service (DDoS) assaults.

**Figure 40**  
*Clean Dataset vs Query Flood Attack Dataset*



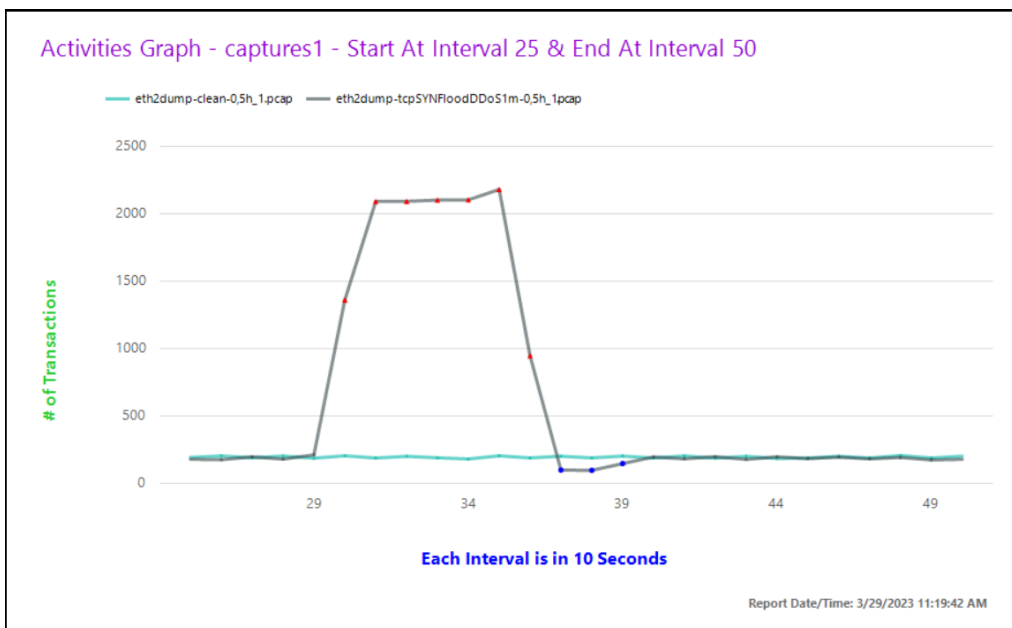
**Figure 41**  
*Clean vs Query Flood Attack Dataset between 280 to 380 seconds*



The dataset under consideration for analysis is the TCP SYN Flood dataset. The occurrence of the DDoS assault may be observed during the time interval of 280 to 380 seconds, as depicted in figures 35 and 36. A SYN deluge, sometimes referred to as a half-open assault, is a type of network-layer attack when a server is inundated with an overwhelming number of connection requests that are not acknowledged upon receipt. A significant quantity of open TCP connections might deplete the server's resources, resulting in the suppression of genuine traffic and rendering the establishment of new lawful connections unfeasible, so posing a challenge to the server's operational capacity. Following the culmination of the assault, a discernible decline in activity is observed, suggesting that a period of time will be required for the server to recommence its uninterrupted data processing operations.

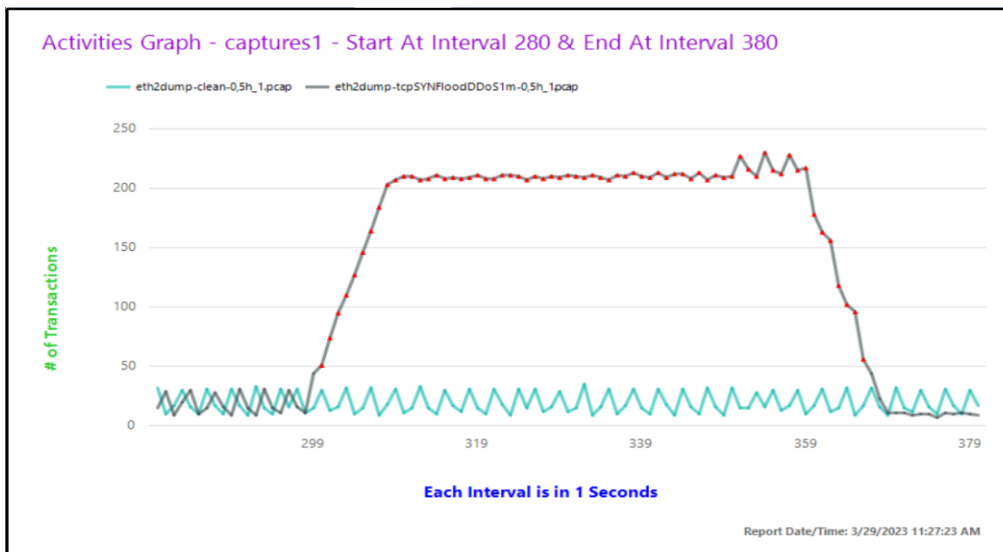
**Figure 42**

***Clean vs TCP SYN Flood Attack Dataset***



**Figure 43**

*Clean vs TCP SYN Flood Attack Dataset between 280 and 380 seconds*

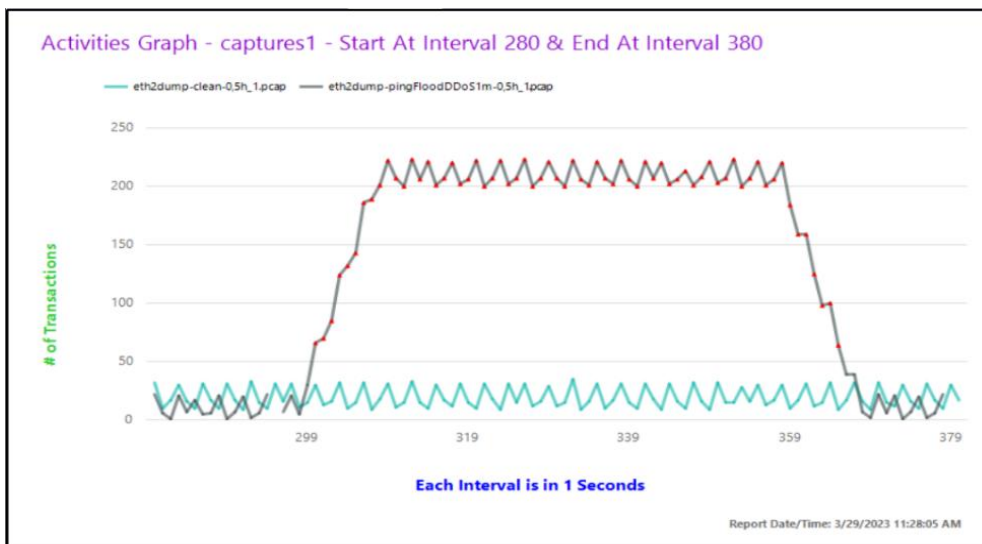


### **Ping Flood Attack.**

The dataset under consideration for analysis is the ping Flood assault dataset. As seen in Figure 44, the duration of the Distributed Denial of Service (DDoS) assault ranges from 280 to 380 seconds. The occurrence of the intrusion is apparent by the visual indication of the intruder's presence, symbolized by a red dot.

**Figure 44**

*Clean vs Ping Flood Attack Dataset between 280 and 380 seconds*

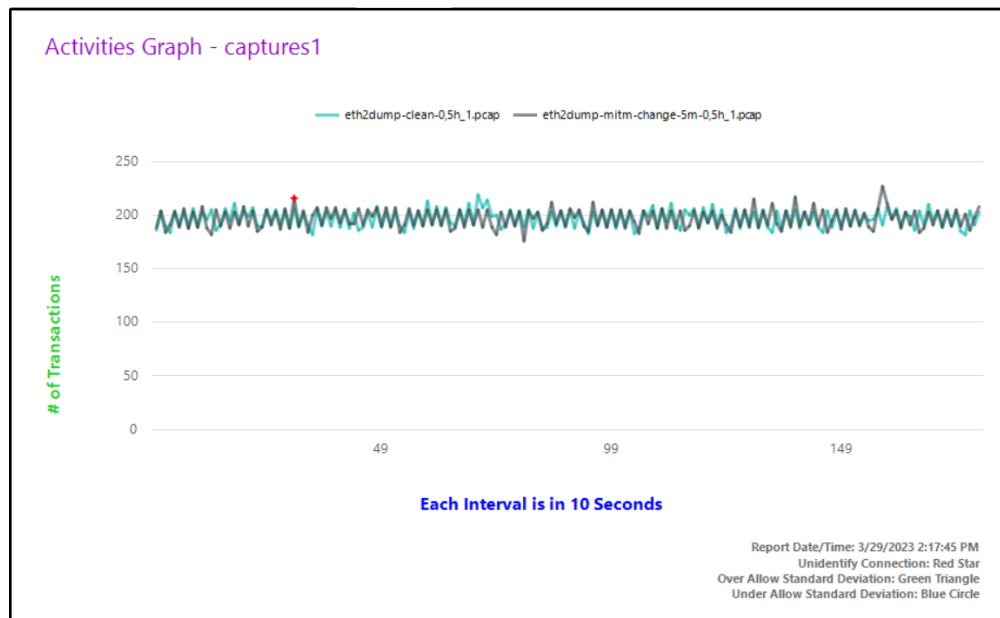


### **MITM Attack.**

The dataset under investigation is the MITM attack dataset. Figure 37 illustrates the absence of definitive proof on the presence of the Man-in-the-Middle (MITM) assault. The presence of a red dot on the graph indicates the occurrence of an unidentified connection being established with the system at a certain point in time. This singular piece of information is the sole insight that can be derived from the graph.

**Figure 45**

***Clean Dataset vs MITM Attack Dataset***



The term "unidentified connection" pertains to a connection that has been made using a combination of MAC and IP addresses, which has not been recorded in the database during phase 1 and does not appear in the clean dataset. As seen in Figure 38, in the context of this assault, the activation of an alert can be achieved by means of an unattributed connection, followed by the use of supplementary TCP packet data to conduct further investigation into the matter. Figure 38 illustrates the comprehensive header information pertaining to TCP and Modbus exchanges. The inclusion of TCP transactions with acknowledgment numbers, sequence numbers, and timestamps, along with Modbus information such as unit ID and function code, will provide the administrator with the means to discern the attacker's approach and intentions.

Figure 46

## TCP Packet Information

PacketNo	SourceIP	DestinationIP	Length	Transaction_ID	Protocol_ID	Len	Unit_ID	Function_code	Reference_Number	AckNum	SeqNum	NextAckNum	TsSec	TsUsec
1	172.27.224.70	172.27.224.250	0	NULL	NULL	NULL	NULL	NULL	NULL	2985000945	4032493011	NULL	1535046048	376131
2	172.27.224.70	172.27.224.250	12	0	0	6	1	3	0	2985000945	4032493011	4032493023	1535046048	397071
3	172.27.224.70	172.27.224.250	12	0	0	6	1	3	0	2985000945	4032493011	4032493023	1535046048	470440
4	172.27.224.250	172.27.224.70	31	0	0	25	1	3	5632	4032493023	2985000945	2985000976	1535046048	473558
5	172.27.224.70	172.27.224.250	0	NULL	NULL	NULL	NULL	NULL	NULL	2985000976	4032493023	NULL	1535046048	688103
6	172.27.224.70	172.27.224.250	12	0	0	6	1	3	0	2985000976	4032493023	4032493025	1535046048	782418
7	172.27.224.250	172.27.224.70	31	0	0	25	1	3	5632	4032493025	2985000976	2985001007	1535046048	794216
8	172.27.224.251	172.27.224.250	0	NULL	NULL	NULL	NULL	NULL	NULL	1954923156	534429382	NULL	1535046048	881953
9	172.27.224.250	172.27.224.251	0	NULL	NULL	NULL	NULL	NULL	NULL	534429383	1954923156	NULL	1535046048	892213
10	172.27.224.250	172.27.224.251	0	NULL	NULL	NULL	NULL	NULL	NULL	534429383	1954923156	NULL	1535046048	893047
11	172.27.224.251	172.27.224.250	0	NULL	NULL	NULL	NULL	NULL	NULL	1954923157	534429383	NULL	1535046048	893049
12	172.27.224.250	172.27.224.251	0	NULL	NULL	NULL	NULL	NULL	NULL	534429383	1954923157	NULL	1535046048	893225
13	172.27.224.250	172.27.224.251	0	NULL	NULL	NULL	NULL	NULL	NULL	0	1954923157	NULL	1535046048	902229
14	172.27.224.70	172.27.224.250	0	NULL	NULL	NULL	NULL	NULL	NULL	2985001007	4032493025	NULL	1535046048	96
15	172.27.224.70	172.27.224.250	12	0	0	6	1	3	0	2985001007	4032493025	4032493027	1535046048	94421
16	172.27.224.250	172.27.224.70	31	0	0	25	1	3	5632	4032493027	2985001007	2985001038	1535046048	104185
17	172.27.224.70	172.27.224.250	0	NULL	NULL	NULL	NULL	NULL	NULL	2985001038	4032493047	NULL	1535046048	312105
18	172.27.224.70	172.27.224.250	12	0	0	6	1	3	0	2985001038	4032493047	4032493059	1535046048	406274
19	172.27.224.250	172.27.224.70	31	0	0	25	1	3	5632	4032493059	2985001038	2985001069	1535046048	413477
20	172.27.224.70	172.27.224.250	0	NULL	NULL	NULL	NULL	NULL	NULL	2985001069	4032493059	NULL	1535046048	624094
21	172.27.224.70	172.27.224.250	12	0	0	6	1	3	0	2985001069	4032493059	4032493071	1535046048	718344

The exploitation of fragmented organizations and system management by malware necessitates the implementation of measures that enhance threat detection visibility for network administrators. By providing network administrators with the ability to interpret aberrant activity and its associated repercussions, these measures can effectively mitigate the impact of malware attacks.

### Discussions

Comprehending the underlying behavior of malware is of utmost importance in order to ascertain the extent of an assault and effectively halt its advancement. The act of monitoring the movement of malevolent network traffic and other forms of hostile communication can be essential in ascertaining the point of entry for malware into a network. Ethernet has emerged as the prevailing norm for networking in industrial and corporate environments. Consequently, the ability to automatically scrutinize ethernet packets for the identification of malicious software is of utmost importance. This is due to the fact that human examination of network packets is excessively time-consuming. Malicious software compels computer systems to exhibit aberrant behavior, thereby necessitating a comprehensive understanding of the behavioral signs associated with such malware. Once a baseline for typical network traffic has been established, the subsequent phase involves gaining an understanding of malware behavior. An inquiry will be initiated in response to any network traffic or activity that deviates from the established patterns or behaviors. This stage of the architecture entails adopting a temporal perspective on network activity in order to detect malware on the network by utilizing baselines and data on malware behavior. Analysts



possess the capability to observe the progression of network traffic over a period of time in order to detect deviations from normal patterns. This ability is particularly crucial for ascertaining the occurrence of a security breach and assessing the extent of the attack's progression. Such analysis is facilitated by a comprehensive understanding of the system's baseline characteristics.

While packet capture proves to be a valuable monitoring approach, it is important to include supplementary security measures. As seen in the preceding section, the absence of distinct signs characterizes a Man-in-the-Middle (MITM) assault, with the exception of a possible Address Resolution Protocol (ARP) poison attack, which may be identified by an unregistered connection. The deployment of a packet analyzer at the network interface will result in a decrease in the quantity of gathered visibility. The potential failure of the software to capture data may impede the system's ability to detect the beginning of a malware epidemic. Incorporating the examination of PCAP files into the network security framework is imperative; nonetheless, it is crucial to acknowledge that it should not be the only means of safeguarding the network. Phase 4 of the framework offers further safeguards for the SCADA system.

#### **4.4 Defending Phase**

The primary focus of this phase is the mitigation of risks. During this stage, the framework offers a systematic approach to address security threats such as denial of service and man-in-the-middle attacks. The SCADA system is vulnerable to a multitude of risks and hazards. The present architecture is a first endeavor aimed at facilitating the enhancement of security in SCADA systems. The framework may be modified to focus on certain components of the SCADA system or expanded to address the evolving and extensive cyber security threat landscape.

##### ***Experimental Set Up***

###### **Step 1 - Compile theoretical and practical principles learned.**

In contemporary times, it is exceedingly rare to encounter a cyber environment that is devoid of risks. (Hoppa, 2023). The level of complexity exhibited by cyberattacks targeting SCADA systems is progressively escalating (Lee & Hong, 2020; Mijwil & Aljanabi, 2023).

The development of SCADA systems did not prioritize cyber security considerations. Consequently, adapting traditional intrusion detection techniques from IT to fulfill the specific needs of SCADA poses a challenging issue (Zhu & Sastry, 2010). The integration of IoT technology into the SCADA systems has given rise to a new generation of SCADA systems. However, this integration also introduces additional vulnerabilities to the system. One such weakness is the potential for IoT malware, which poses a significant challenge because to its unstoppable nature (Maier et al., 2014). Modbus/TCP has gained significant adoption due to its ease of use and widespread availability (Abdulwahid et al., 2023). The Modbus protocol has emerged as the prevailing standard for industrial control systems, as indicated by the widespread adoption and support from a majority of manufacturers in the automation sector. The security of Modbus protocol is inadequate because it lacks the capability to survive intentional cyber-attacks that commonly target typical IT networks (Rahman et al., 2022). Theoretical data suggests that the implementation of SCADA systems utilizing the Modbus protocol necessitates careful consideration of cyber security measures.

The Modbus RTU protocol is designed to operate with a singular master device, which is unable to function as a slave device. Frequently, it is employed to establish connections with field-level equipment, such as sensors, valves, actuators, and other similar devices, because to their inherent need for real-time system behavior. The Modbus RTU cable is limited to transmitting a single set of signals at any given time. Either the only master of the RTU is engaged in communication, or one of the subordinate units, referred to as slaves, is transmitting data. There was a lack of a standardized approach for transmitting data between two Modbus RTU masters, where each master operated on a separate network segment and was the sole master on that segment. The incorporation of Modbus TCP has significantly simplified operational processes. Many more Modbus TCP devices may be controlled by a single controller via a single Ethernet cable. Modbus TCP enables network designers to leverage multiple masters inside their network infrastructure. The connection between a device functioning as a slave and a Modbus TCP master has been disrupted. The Modbus TCP device has the capability to be accessed and manipulated by an indefinite number of master devices. In a multi-controller system, it is a consistent feature that all controllers will possess unrestricted access to device data. Devices that utilize Modbus TCP protocol are limited to executing a singular transaction at a time and do not retain any data from preceding

transactions. The implementation of the Modbus protocol using TCP/IP occurs at the application layer. Consequently, it is imperative to take into account the buffering concern and the queuing of frames according to the First Input, First Output (FIFO) principle. Multiple masters have the capability to concurrently write to a shared register or coil within a Modbus device, and it is possible for these masters to assign different values to the register or coil. The discrepancy in motor speed settings between Modbus TCP Master 1 and Modbus TCP Master 2, with one setting the speed to 200 rpm and the other to 2000 rpm, would not have any impact. The Modbus TCP device possesses the capability to manage several TCP connections, enabling it to effectively evaluate incoming messages. By leveraging this functionality, the device is able to utilize the most recent write message in order to ascertain the appropriate motor speed. It is possible for slaves to have several masters, however they are only able to obey the commands of one master at a given moment. In the event that two masters concurrently attempt to poll a slave, it is possible that one of them may be required to retransmit the command, resulting in a subsequent delay. In the absence of a prioritization mechanism, the process lacks determinism and is hence vulnerable to potential delays. In the event that the specified time limitations are not adhered to, there exists the potential for the system to have malfunctions. Modbus TCP/IP is a viable solution for enabling communication between two gateways, such as a HMI and a PLC, a gateway and an HMI, or an Input/Output (I/O) device that does not necessitate Real-Time System (RTS) prerequisites (Figuroa-Lorenzo et al., 2019). Modbus TCP is vulnerable to attacks because to its inherent limitations in resisting plain text communications and its absence of integrated authentication mechanisms. The vital infrastructure and operations of SCADA systems need the implementation of risk management, control, and mitigation measures. The monitoring, regulation, and mitigation of risk are imperative due to the critical infrastructure and operational significance of SCADA, the widespread use of the Modbus TCP protocol, and the lack of adequate cybersecurity measures.

Understanding and analyzing the gathered traffic is crucial for comprehending the communication loop being utilized. This information has the potential to identify vulnerabilities or attack vectors, rendering it advantageous for a potential attacker. The next section will examine several methodologies for initiating attacks against Scada systems. The research conducted by Morris and Gao titled "Industrial Control System Cyber Attacks"

examines a total of 17 cyber assaults that target industrial control systems utilizing the MODBUS communication standard. The study categorizes assaults into four distinct classifications: reconnaissance, response and measurement injection, command injection, and denial of service. This discussion aims to offer an in-depth analysis of the diverse array of dangers associated with industrial control systems (Morris & Gao, 2013).

Reconnaissance attacks involve the gathering of information pertaining to control system networks. This includes activities such as mapping the network architecture and identifying various device properties, such as the manufacturer, model number, supported network protocols, system address, and system memory map. The prevalent reconnaissance attacks against MODBUS servers encompass address scanning, function code scanning, device identification attacks, and points scanning. The process of address scanning is utilized to identify Internet-connected ICS servers. The function code scan is a process that identifies network operations that may be performed on a designated server, based on the functionality provided by its function code. The device identification attack allows an assailant to obtain various details about a device, including as its vendor name, product code, major and minor revision numbers, and other relevant information. The utilization of the points scan technique allows the assailant to systematically create a comprehensive memory map of the targeted device. The integration of the results obtained from the address scan, function code scan, device identification attack, and points scan may be used to provide a distinctive identifier for MODBUS servers that are prevalent within a certain business, application scenario, or manufacturer. The aforementioned signature may be employed for the purpose of creating maps of identified systems based on factors such as organization, use case, or vendor. Furthermore, these signatures may be utilized to create a comprehensive database including vulnerabilities and exploits pertaining to each of the aforementioned categories.

There are three distinct types of response injection attacks. Firstly, it is important to note that injection attacks might originate from the manipulation of a PLC or RTU. These network endpoints function as servers that respond to client requests. Furthermore, response injection attacks have the capability to intercept network packets and manipulate their contents while being transmitted from the server to the client. Finally, it is possible to generate and disseminate response injections using a network device operated by a third party. In this particular scenario, there might be several reactions to a client's inquiry, and the

incorrect answer may be prioritized due to the exploitation of a race situation or a secondary attack, such as a denial-of-service attack, which hinders the legitimate server from providing a response. Polling techniques are commonly utilized inside industrial control systems for the purpose of ongoing monitoring of the operational state of a distant process. Polling involves the process of transmitting a query from the client to the server, which is then followed by the sending of a response packet from the server back to the client. The utilization of state information serves several purposes in the context of human-machine interaction, process monitoring, and data storage in historians. Additionally, it plays a crucial role in feedback control loops, where it facilitates the measurement of process parameters and the execution of control actions based on the current state of the process. Numerous network protocols utilized in industrial control systems exhibit a deficiency in terms of authentication capabilities, hence failing to provide a means of verifying the source of packets. This enables malicious actors to gather, modify, and transmit packets containing sensor reading values. Moreover, it is common for industrial control system protocols to exclusively accept the initial answer packet to a query while dismissing subsequent responses as invalid. This functionality facilitates the generation of response packets and allows for the exploitation of timing attacks to inject these responses into a network at the precise moment when the client anticipates receiving them.

NMRI (Naive Malicious Response Injection) assaults can be characterized as rudimentary or unsophisticated in nature. NMRI attacks are predicated upon the capacity to introduce response packets into the network, although they are deficient in terms of information pertaining to the observed or controlled process. Non-Malicious Remote Intrusion attempts might potentially transmit payloads that are deemed illegitimate. For example, an assailant may have executed a sequence of reconnaissance assaults in order to get system addresses, function codes, and memory mapping. However, they may still lack specific details about the monitored process or accurate data contents for each point that has been identified on a server. In the given context, the assailant has the possibility to execute a response injection assault by employing a payload that comprises only of zero values, negative numbers, excessively big numbers, or other forms of possibly inaccurate data. Alternately, NMRI attacks may rely on limited process information. For example, a potential assailant may possess knowledge on the intricacies of processes, including their limitations and the permissible material associated with each point on a server. However, they may lack the

requisite skills or resources to execute more sophisticated forms of attacks. An assailant has the potential to activate an alarm, as an example.

Command injection attacks include the insertion of unauthorized configuration and control commands into the control system of a given system. Control systems are overseen by human operators who occasionally intervene in supervisory control activities. Cybercriminals may endeavor to introduce counterfeit supervisory control activities into the network infrastructure of a control system. In general, remote terminals and intelligent electronic equipment are configured to autonomously monitor and regulate the physical processes occurring at a distant place. This programming encompasses ladder logic, C code, and registers that store essential control parameters, including the upper and lower thresholds that govern process control operations. Command injection methods may be employed by hackers to alter ladder logic, C code, and remote terminal register settings. The potential ramifications of malicious command injections encompass the disturbance of process control, interruption of device communications, unauthorized modification of device settings, and unauthorized adjustment of process set points. There are three distinct kinds of command injection attacks, namely Malicious State Command Injection (MSCI), Malicious Parameter Command Injection (MPCI), and Malicious Function Code Injection (MFCI).

The MSCI attacks include the transmission of malicious commands to remote field equipment, resulting in the process control system transitioning from a safe condition to a dangerous state in an irregular manner. In general, actuators that are affixed to physical systems, such as switches or valves, are linked to a digital or analog output that is in turn connected to a RTU or IED. The utilization of a digital point in a register to represent each output facilitates the establishment of a link with the cyber system. Modifying the state of one or multiple bits inside a register of this nature results in an instantaneous impact on the corresponding physical actuator. The ON/OFF mechanism of a pump, for example, can be modified by assigning a value to a specific bit in a register located on a RTU. The modifiability of these registers is facilitated by the utilization of network protocol write commands. The MODBUS protocol, for example, provides support for instructions such as write coil and write register. If an opponent possesses knowledge about the implementation details of a device, including its memory map, they possess the capability to create a command that may alter the states of actuators. The modification of recorded settings

pertaining to system control mode, pump state, and solenoid status may be achieved through the utilization of MODBUS instructions. In order to commence the assault, a write register instruction is employed, specifically targeting address 0xABCD through the utilization of MODBUS function code 03. This instruction serves the purpose of configuring the control mode to manual. The activation of the pump is achieved by issuing a write register instruction to the memory address 0xABCD. In the context of the gas pipeline control system, it is considered a critical condition when the pressure within the pipe exceeds 60 PSI. In the event that the pressure is above this predetermined level, there is a possibility of detrimental effects on the various components of the system. When the system is switched to manual mode and the pump is activated, it results in an increase in pressure to a critical level. The ability of an operator to effectively monitor a system through a HMI is crucial in detecting and responding to increasing pressure levels by implementing appropriate remedial measures. Furthermore, the increasing gas pressure has the potential to trigger a process alarm, therefore notifying an operator.

The objective of Denial of Service (DOS) attacks on industrial control systems is to render the entire system inoperable by impeding the correct functioning of a specific component within the cyber-physical system. Consequently, Denial of Service (DoS) attacks have the potential to be directed on either the digital or tangible infrastructure. Distributed Denial of Service (DDoS) attacks are specifically designed to disrupt the normal functioning of a computer system by either targeting its communication connections or disabling programs that run on system endpoints. These endpoints are responsible for crucial system operations such as logging data and regulating communications. Denial-of-Service (DoS) attacks against the physical system encompass a spectrum of actions, ranging from the manual manipulation of valves and switches to the deliberate destruction of physical components that hinder the system's functionality. Traffic jamming refers to a form of denial-of-service attack when a network endpoint is overwhelmed by a substantial influx of traffic. The perpetrators endeavor to overpower the endpoint by providing transmissions at a pace that exceeds its processing capacity or by sending packets specifically tailored to trigger software faults, resulting in exceptions that lead to the network stack, running application, or operating system of the targeted device experiencing a crash.

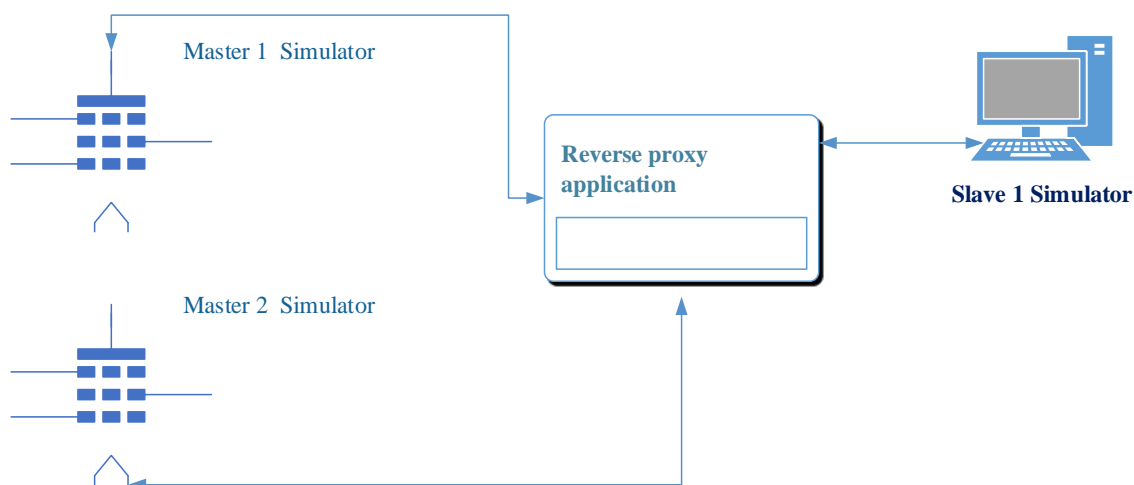
## **Step 2 - Design Principles.**

In contrast to the recommendation proposed in Phase 2 of the framework, which advocated for the implementation of offensive security measures to fortify the SCADA network, the idea put out in Phase 4 of the framework was to employ defensive security strategies with the aim of mitigating possible cyber threats. The subsequent stage of the structure involves using the cognitive abilities possessed by attackers to exploit the organization, hence leading to the implementation of security measures or an extra layer of safeguarding for the system. The concept of offensive security involves the utilization of tools, methods, and strategies akin to those employed by real attackers in their attempts to breach an organization's security. However, rather than causing harm, security experts apply these approaches to enhance the security measures of the business. The fourth stage of the architectural process involves implementing proactive measures to identify, prevent, and reduce existing dangers. Defensive operations have the capacity to effectively halt an adversary's attack, strategically prolong the duration of engagement, mitigate the necessity for more military personnel, and establish a favorable context for launching a counteroffensive, so enabling the defenders to regain control and overcome the opposing party. After the identification of a vulnerability through the preventive, detective, and reactive procedures, the subsequent stage of safeguarding, known as defensive security, is enacted.

### **Step 3 - Formulate design principles.**

Regarding the Modbus TCP protocol, communication can occur in two distinct manners. The execution of software that transmits action requests is the duty of the Modbus master, which assumes a role like to that of a supervisor for the slave devices. The proposed framework proposes the utilization of this theory as a means of safeguarding the SCADA system by impeding the progress of attacks or rendering the master component inoperable, hence enabling alternative masters to assume control and sustain the operation. The utilization of Modbus TCP/IP, which allows for many masters, is a potential means of enhancing the security of SCADA systems. In contrast, the architecture proposes the utilization of a proxy server as a means to protect the slave against specific predicted threats, such as Distributed Denial of Service (DDoS). The design principles employed for this particular phase are illustrated in figure 40, as presented below.



**Figure 47*****Multiple master and reverse proxy application process flow***

In the context of Modbus TCP, the stateless nature of the connections allows many masters to concurrently write distinct values to a shared register or coil within a Modbus device. Due to the capability of Modbus TCP to accommodate an unlimited number of masters, it may be leveraged as a means to acquire more time during the execution of an attack. Slaves may have several masters, but they may only obey one at a time. One such configuration involves utilizing a server as a Modbus TCP slave, with three clients operating as masters. The Modbus TCP master has the capability to periodically request information from the Modbus TCP slave using a variety of commands, including read and write operations. Due to the inherent limitation of slaves being able to reply to only a single master at any given moment, the transmission of information is restricted to a singular master. It is possible for other master's degrees to receive a return status of 3, which corresponds to the MBCHNL\_RESP\_STATUS\_CANCELLED. The response code MBCHNL\_RESP\_STATUS\_CANCELLED was placed in the stack since it incorporates a logical process to ascertain that the request is not a duplicate. As a result of constraints in available resources, the slave channel exhibited an inability to concurrently handle communications originating from all three masters. The delay in transmitting the request to the subordinate was a result of the expeditious actions of the individuals in positions of authority. In this particular scenario, it is possible that the default interval may be

insufficiently lengthy. The identification of assaults on SCADA networks will greatly enhance their defensive capabilities. By manipulating the rate at which the hacked and exploited master operates, it becomes possible to effectively render it inactive and halt the ongoing assaults.

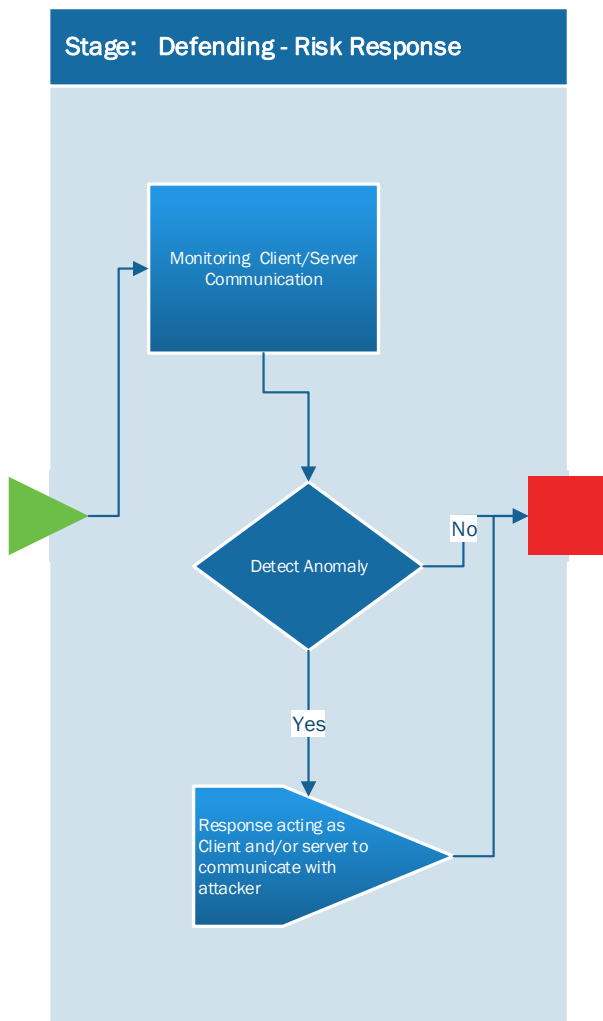
It is advisable to employ the concept of a proxy and reverse proxy server in order to enhance the security of the devices linked to a SCADA network. A conventional proxy server is designed to safeguard the primary entity, whereas a reverse proxy is intended to provide protection for subordinate entities. This is contrary to the typical functionality exhibited by a conventional proxy server. A reverse proxy refers to a specific kind of proxy server that accepts requests from a primary server, relays those requests to a secondary server for processing, and afterwards transmits the processed outcomes back to the original primary server, giving the impression that the proxy server itself has executed the operation. The reverse proxy server manages all interactions of the master, resulting in the master being unaware of which slave is responsible for executing its request. In the context of TCP connections, the TCP reverse proxy server assumes the role of an intermediary entity. The network provides a multitude of benefits. The use of this technology has the potential to provide network security by mitigating malicious assaults, reducing latency, and enhancing overall network performance. Moreover, it provides functionalities such as caching, load balancing, and traffic control. The proxy server serves as an intermediary entity that facilitates communication between the client and the server, allowing the client to establish a connection with the server without the need for a direct connection. Moreover, the proxy server enhances security measures by effectively screening out potentially harmful network traffic and imposing limitations on access to designated websites. The proxy possesses the capacity to interpret and comprehend the data transmitted between the servers and the slaves. It is conceivable for the request being submitted or the information being received to undergo alterations. The purpose of this tool is to analyze the data being sent between two Transmission Control Protocol (TCP) nodes. The configuration of the system can include the specification of a specific interface, a local TCP port for listening, and a designated destination address for establishing a connection. The connection between the client and the Modbus device is facilitated by the proxy. The implementation serves as an additional protective measure for the SCADA network. By integrating the ideas of proxy and reverse

proxy, together with network communication monitoring, communication sharing, and client-server interaction, the resultant system will function as a Modbus data processing center.

### ***Experiment Implementations***

#### **Step 4 - Construction of the Framework.**

The three main objectives of this phase are monitoring the communication between the master and the slaves, identifying any irregularities, and then responding to any attacks. The utilization of several master models for the purpose of enabling uninterrupted operation has rendered the achievement of these objectives viable. Furthermore, it is advisable to employ a reverse proxy server that fulfills the dual role of safeguarding the slaves and functioning as a firewall. The reverse proxy application will actively monitor network traffic and respond accordingly in the event of any anomalous activity. The identification of atypical events is an additional element of the framework, which was addressed in the preceding phase. The ABCD risk assessment approach employs a diverse range of methodologies with the aim of ensuring system security. Modifying the polling rate and assuming the role of an intermediary through the utilization of a proxy software are two instances of such actions. The inclusion of the proxy application in the SCADA system is vital when deemed necessary. The implementation of this approach would yield the establishment of a supplementary stratum of security for the crucial subordinate element, alongside the deployment of a gateway mechanism that effectively sieves out undesired network traffic.

**Figure 48*****Defending Phase Process Flow*****Step 5 – Implementation and experimentation.**

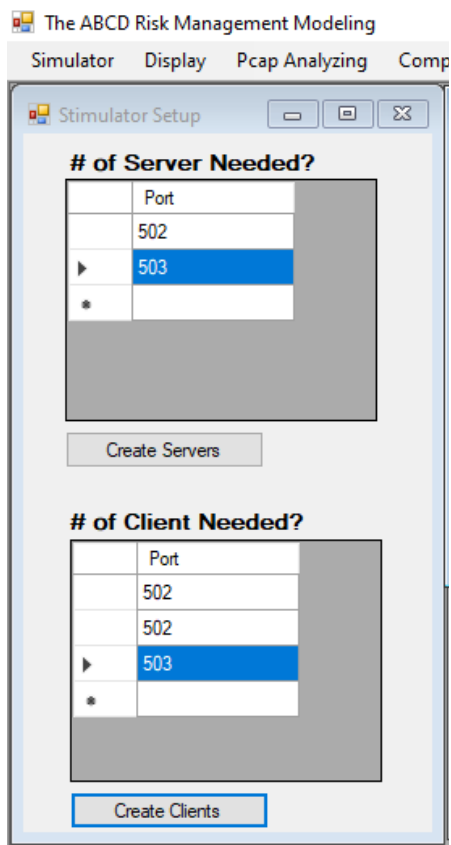
The testbed's master and slave components were implemented in the C# programming language, utilizing the EasyModbusTCP library and interface. The network traffic was recorded by employing the Npcap and Wireshark tools. Yang Luo was responsible for the development of the NDIS 6 Light-Weight Filter at the Google Summer of Code in both 2013 and 2015. Npcap is a software improvement that use WinPcap as its foundation. Npcap introduces several more functionalities to the Wireshark software, encompassing the capability to collect loopback data. The Npcap software, equipped with the Loopback Packet Capture and Injection functionality, enables the interception of loopback packet communications occurring between services inside the same computer by using the Windows

Filtering Platform. Npcap has an interface named NPF\_Loopback and Adapter for the purpose of loopback capture. This adapter facilitates the ability of Wireshark users to record loopback communication in a manner consistent with other adapters that lack support for loopback traffic.

Utilizing the interface displayed below, it is necessary to provide the model with the desired number of server and client instances for setup. The figure presented below depicts the arrangement of two server units and three client devices.

**Figure 49**

***Server and Client Configure***

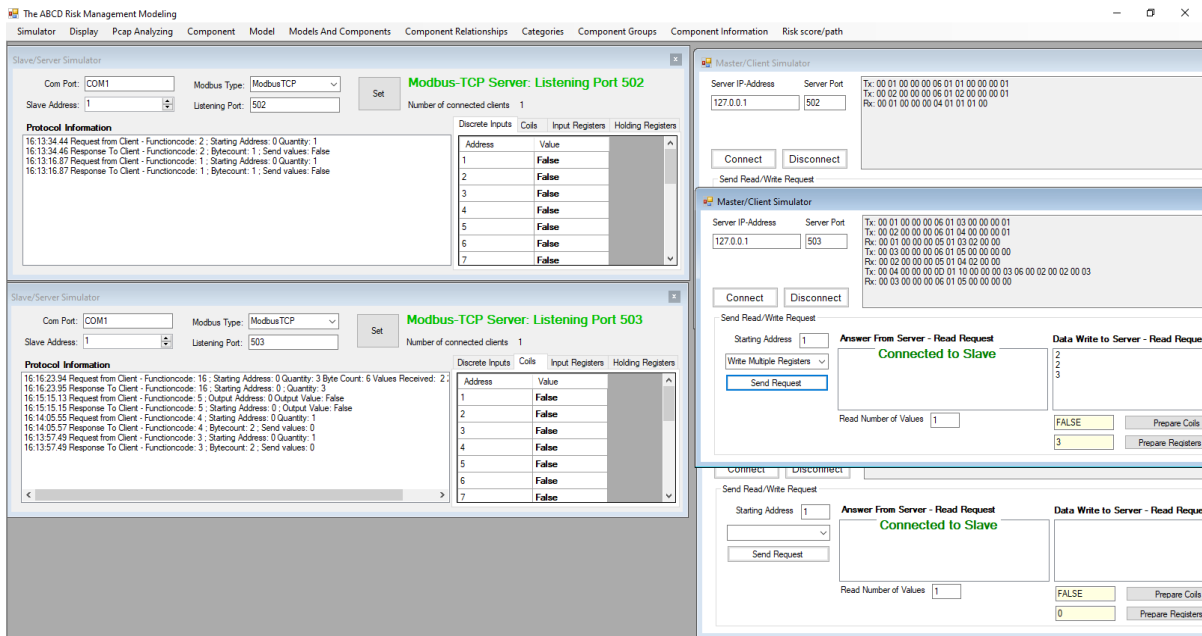


The following are the results obtained from the establishment of two servers and three clients. As seen in the diagram provided, the subsequent step involves initiating the process of assessing and manipulating data in the coils and registers. This is also an opportune

opportunity to initiate the use of Wireshark and commence the process of capturing the communication between these instances.

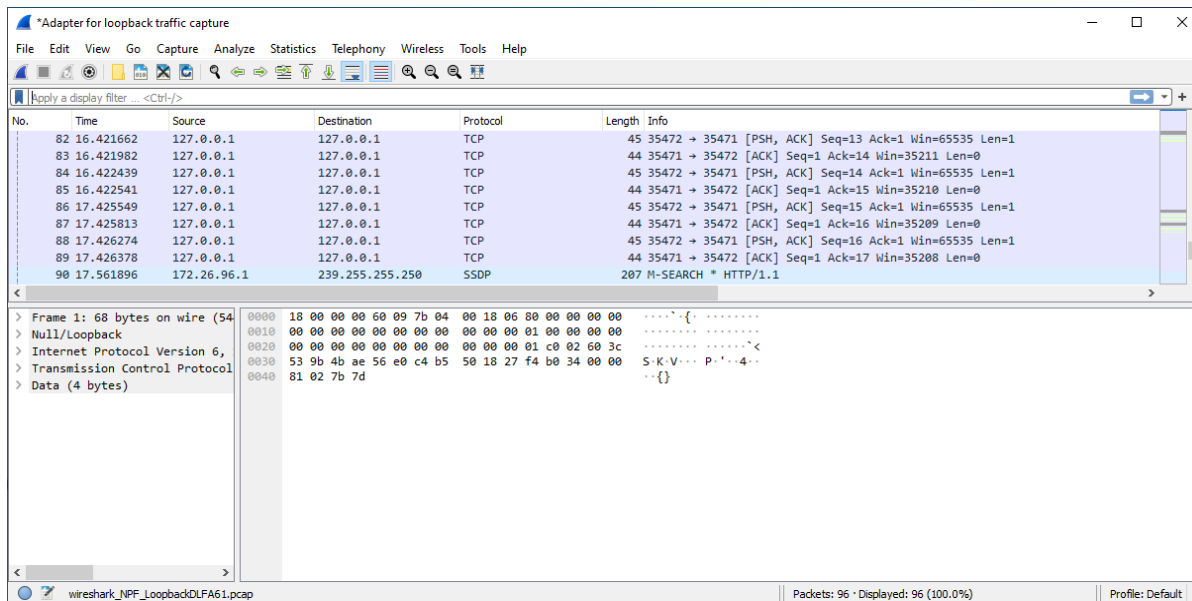
**Figure 50**

*Example of Client/Server Interaction*



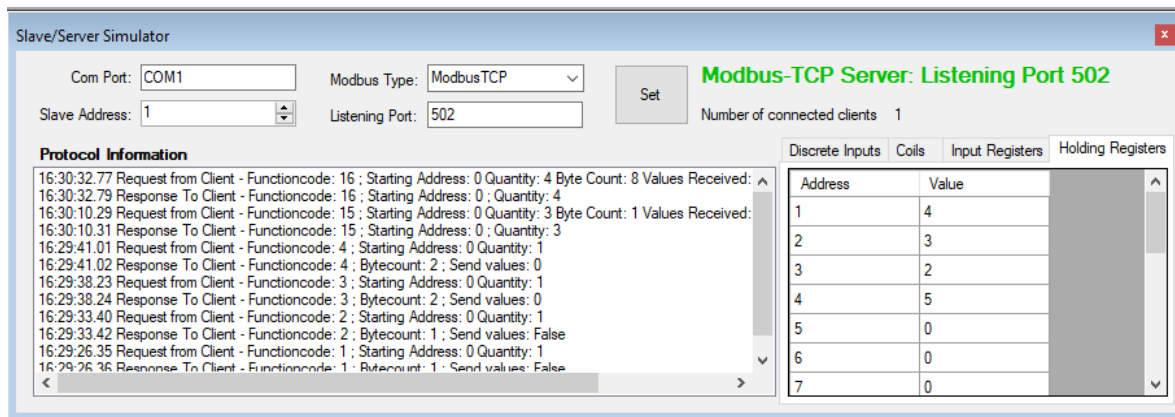
In order to collect local loopback traffic, Wireshark necessitates the utilization of the npcap packet capture library. The program is pre-installed in more recent iterations of Wireshark. Nevertheless, earlier iterations of WinPcap do not include the capability to record loopback traffic. The study utilizes Wireshark, specifically version 4.0.6. During the installation process, a dialogue box will appear, providing the option to install npcap. To begin the process of data collection using Wireshark, the user should first access the main interface and proceed to double-click on the Adapter for Loopback Traffic Capture. The capability to collect loopback traffic has been included in Wireshark. It is imperative to retain the captured traffic subsequent to its cessation within the Wireshark application.

Figure 51

*Example of traffic capture*

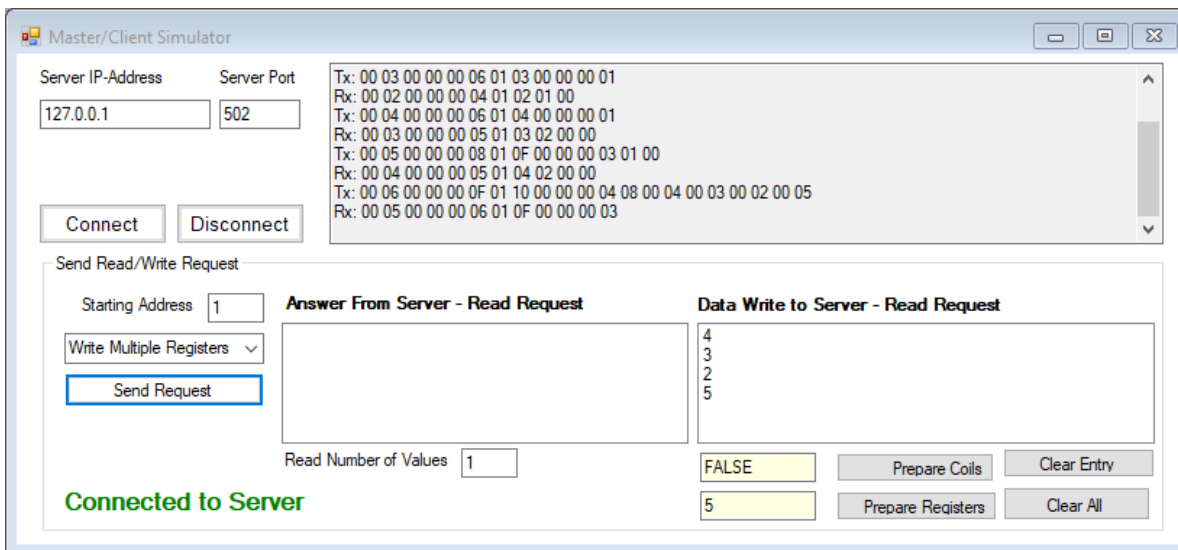
Modbus TCP encompasses four separate tables that act as repositories for data on the Slave device, which is commonly referred to as the server. There are two tables that hold discrete values, commonly referred to as coils, and two tables that store numeric values, commonly referred to as registers. Every coil and register is furnished with both a read-only table and a read-write table. Coils are registers with a capacity of 1 bit, enabling both reading and writing operations to control discrete outputs. Discrete inputs refer to 1-bit input registers that possess the exclusive capability of being read. The input registers are registers with a width of 16 bits, and they are only designed for reading purposes. The range of Modbus discrete input addresses spans from 0 to 65,535.

Figure 52

*Illustration of Slave communication within the framework's Model*

The utilization of the Client Simulator enables the reading or writing of individual or numerous registers and coils to the server, as depicted in the accompanying example.

Figure 53

*Illustration of Master communication within the framework's Model*

The aim of this phase is to develop a reverse proxy that will be responsible for managing the modbus request and response. A distinct port will be assigned to each individual client for configuration purposes. As a result, the poll messages sent by the other client are hidden from both clients. As an illustration, Client A initiates the transmission of a poll to the proxy server, designating the IP address of a subordinate computer. Once the query reaches the proxy server, it undergoes a comparison process with the list of authorized addresses. In the event

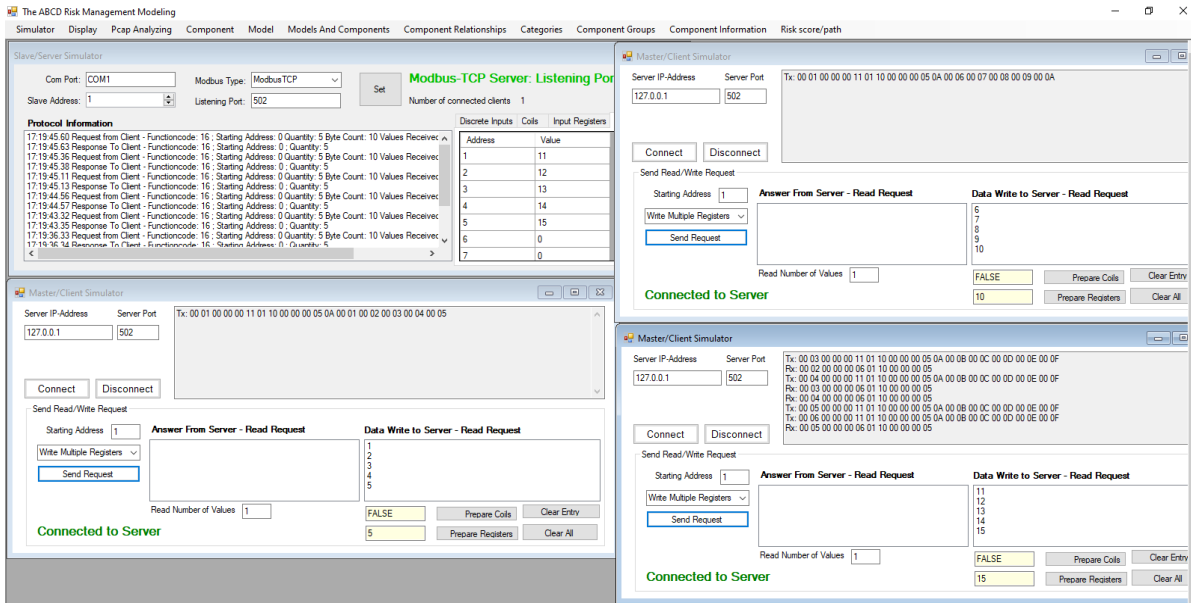


that an address cannot be found, a response in the form of an exception is issued. In the event of a match being identified, the proxy server will ascertain the specific port to which the slave has been set. Subsequently, the poll message is conveyed to the slave port. The proxy server obtains a response from the subordinate entity through the process of polling. Subsequently, the proxy server proceeds to transmit the response back to the original requester. Furthermore, the proxy server is responsible for parsing the answer and storing the retrieved information in a cache block, which may be utilized in future instances. The duration of storage for this information can be modified. In the event that both Client A and Client B make identical requests for information that has not yet reached its expiration, the proxy server will provide a response by retrieving the data from its transitory storage. In the event that one of the clients expresses a need for further information or if the expiration date of the temporary information has lapsed, the process of matching and relaying is reiterated. The voice of a slave is expressed. The polling entity, referred to as Client B, gets the respondent's feedback and proceeds to store the data temporarily, facilitating subsequent access by Client A.

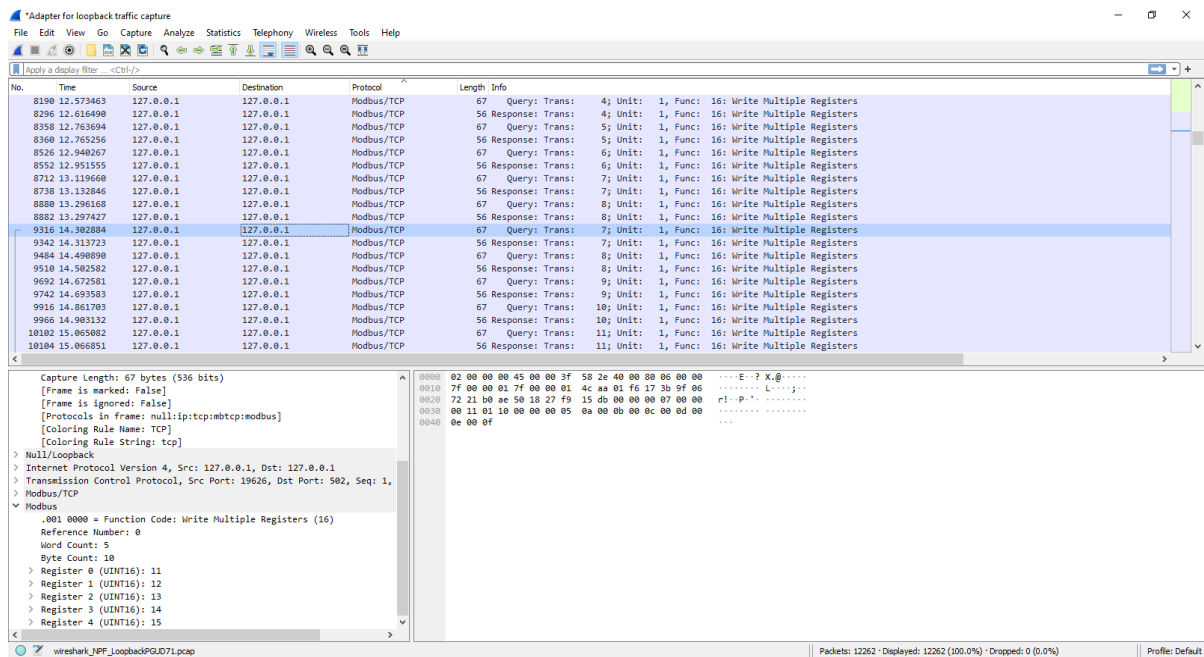
### ***Experiment Results***

The purpose of this phase within the framework is to facilitate the identification and protection of the system at the Modbus communication level. The experimental design included of two distinct portions, with the initial section implementing a replay approach in order to decelerate the assault on the compromised system. The aforementioned concept bears resemblance to the notion of a Distributed Denial of Service (DDoS) attack, since both aim to induce a delay that renders the request null and void. The second segment utilizes the proxy idea as a means to obfuscate the identity of components during encounters with potential attackers, prolong attacks through automated responses to read requests, and establish network visibility as a unified data point. The Figure 45 below exemplifies the concept that several clients can concurrently update the server, and the server will respond to all requests in a way that adheres to the principle of first-come, first-served. Furthermore, the server has the capability to accept and analyze a packet that was first captured using Wireshark and subsequently replayed using the Colasoft Package player application.

**Figure 54**  
*Several clients concurrently update the server*



**Figure 55**  
*Communication intercepted from many clients and servers transmission*

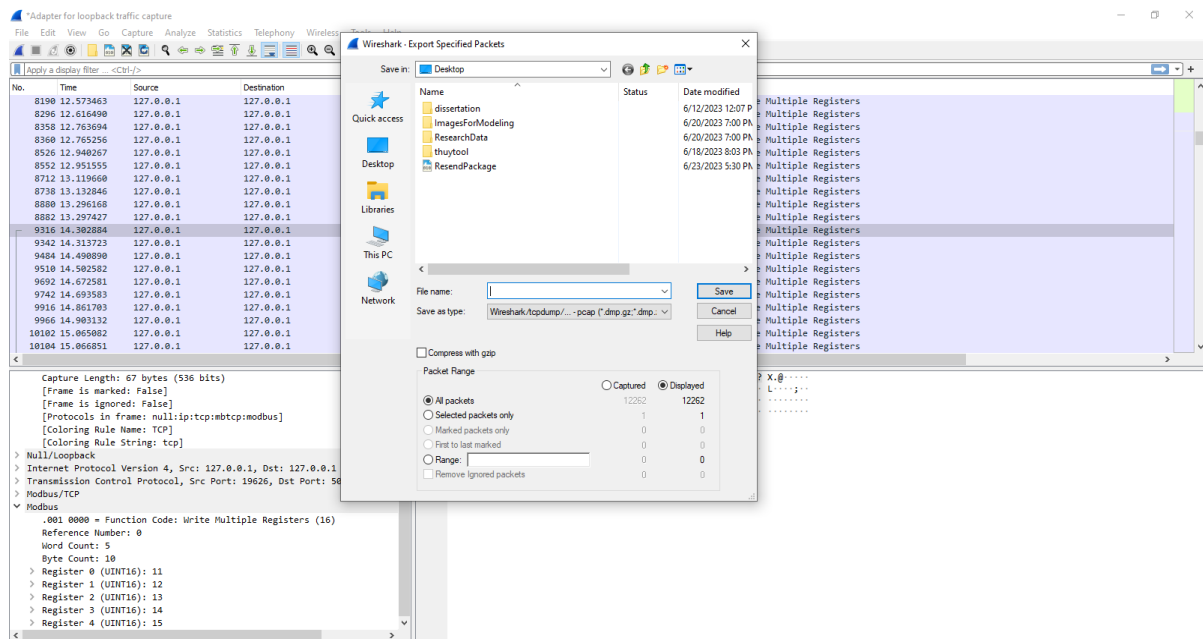


Several diverse applications possess the capability to execute the playback of the PCAP. Several programs in this area include Scapy, tcpreplay, and SharpPCap, among others.

The study project incorporates the utilization of the ColaSoft package player as an integral component of the inquiry. The ColaSoft software bundle exhibits the capability to evaluate network data in real-time, as well as the ability to examine packets that have been captured by other programs like Wireshark, Omnipeek, and similar tools. In this experiment, the software tool Wireshark will be employed to collect and analyze data. Subsequently, the recorded data will be replayed using the ColaSoft package player.

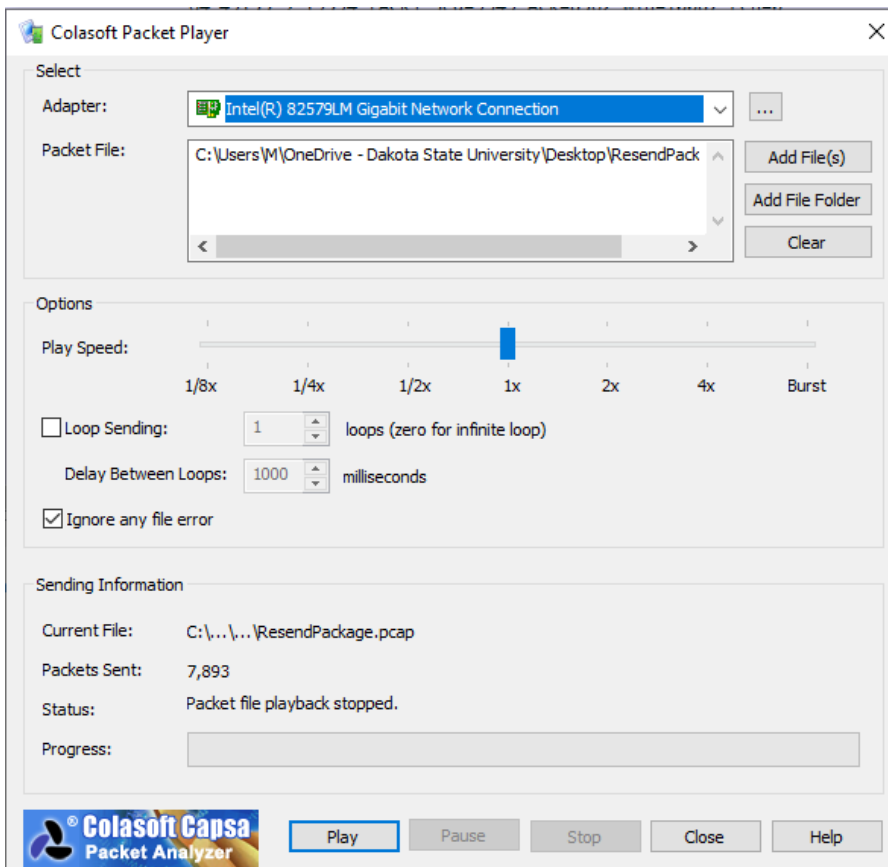
**Figure 56**

*TCP packets were captured and saved to drive*



**Figure 57**

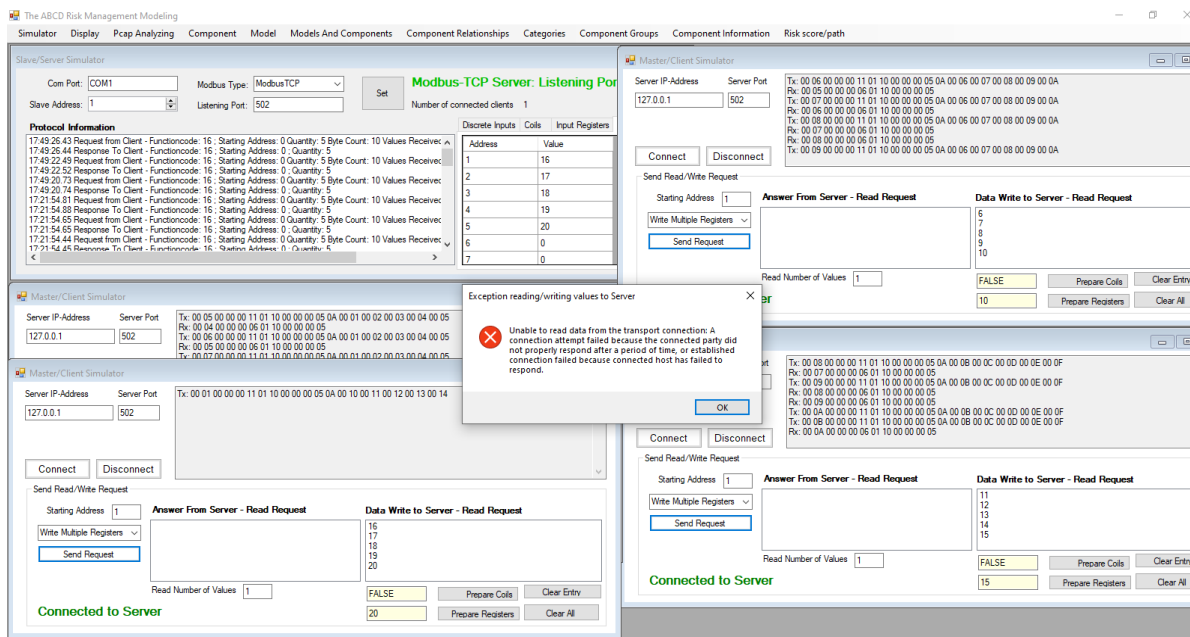
*Colasoft software is used to stimulate the playback attack*



The server may encounter the aforementioned issue, as depicted in the error popup seen in the Figure 49, when an excessive number of transactions are submitted simultaneously. At now, there exists no method to autonomously detect baud rates using any means. It is imperative that both the servers and the clients connecting to the bus employ identical baud rates. The protocol does not specify a specific baud rate at any given instance. During its initial development, Modbus facilitated data transfer at baud rates of 4800, 9600, and 19200 kilobits per second (kbps). Manufacturers have modified their products to accommodate the high-speed serial port, capitalizing on the capability of contemporary computers to support baud rates of up to 115,200 Kbps. This study encompasses the prototype of devices and the subsequent evaluation of their performance in order to ascertain the failure rate of the system and the system's capacity in terms of workload. The experiment was carried out in several settings, when multiple instances of system failures were detected. Nevertheless, the exact moment of system breakdown remains uncertain.

Figure 58

*The playback attack was detected, and error was sent back to the client.*

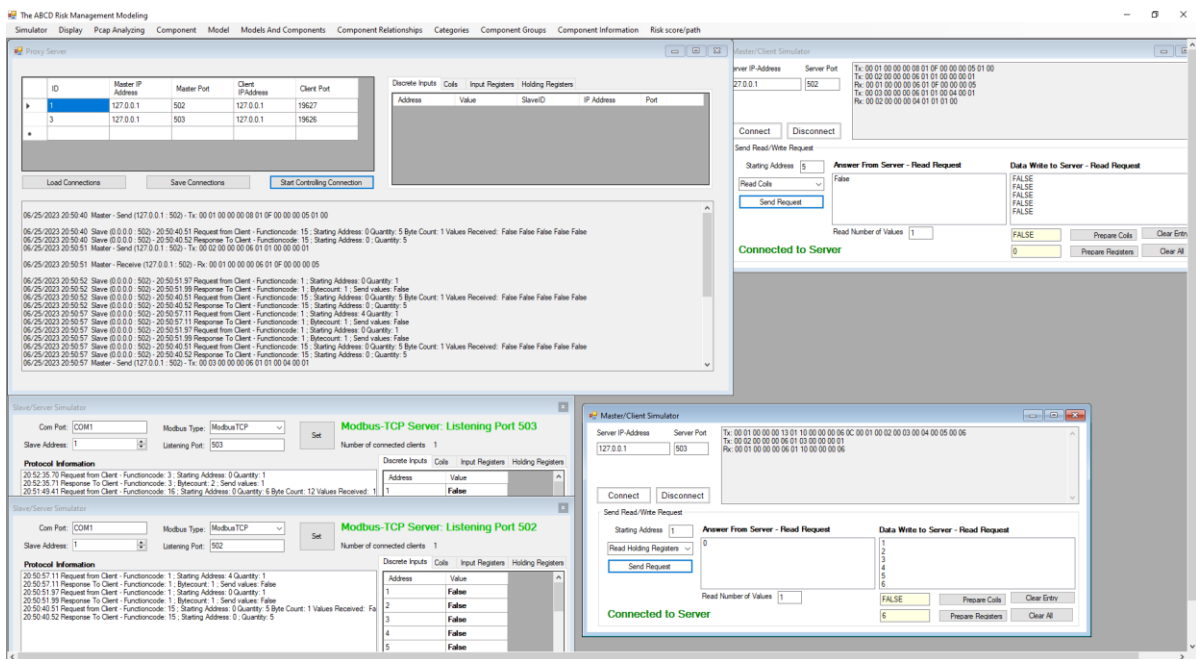


As mentioned in the last part of this chapter, the utilization of a proxy server can effectively reduce and perhaps postpone assaults on Scada systems. The following Figures 50 illustrate the utilization of a proxy server for the purpose of managing several master and client connections. The utilization of a proxy server enables the monitoring of data requests and answers originating from several servers and subordinate systems. As proposed in Phase 3 of the framework, the use of network communication data can be employed for the purpose of anomaly detection and protection against malware in Phase 4 of the framework. The aim is to utilize the strategies employed by the aggressor on themselves. A DDoS attacker employs a method of assault wherein they send orders to overwhelm either the master or slave system with data processing tasks, hence impeding their ability to carry out their intended functions. The proposal suggests that the proxy server will act as a database engine for storing the state of inductor and register tables. Upon receiving a reading request, the proxy server has the capability to retrieve the corresponding value from its database and subsequently fulfill the request, hence eliminating the necessity for the slave to do the task as originally intended. This notion is founded upon the principle of a man-in-the-middle assault. The perpetrator is situated at the focal point, endeavoring to react to and solicit the conduct of others. In this

particular case, the proxy server assumes the role of the central intermediary attacker in order to safeguard the system. The proxy server will effectively handle requests for Distributed Denial of Service (DDoS) assaults, therefore impeding the attack's progress and freeing up compromised systems to carry out their intended operations.

**Figure 59**

*The clients and servers remain operational despite the playback attack.*

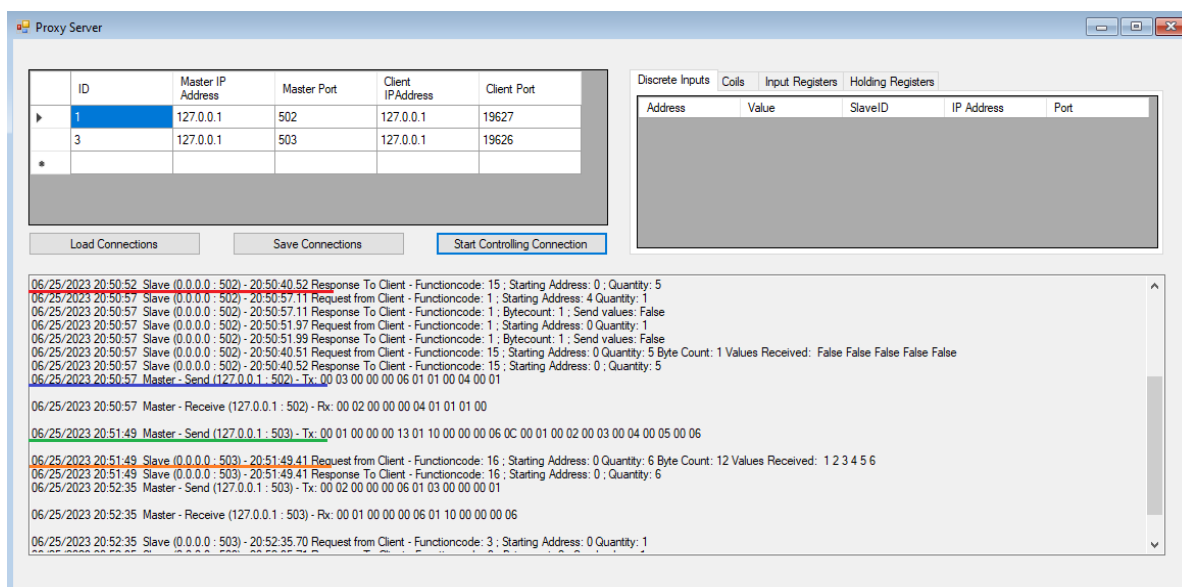


The diagrams presented below depict the locations from which the proxy server acquires data from various subordinate entities. The proxy interface is responsible for storing the information and current state of both masters and slaves. All modifications made to the coils and registers will be documented, therefore guaranteeing that the data accurately reflects the most up-to-date values. In the context of a read request, the proxy has the capability to provide its own values as a response. This eliminates the necessity for slaves to handle the read request, thereby contributing to the mitigation of a potential DDoS assault executed using read requests. The first master device that is linked to port 502 is visually emphasized through the utilization of a blue underlining. The second master, which is linked to port 503, is visually emphasized by the use of green underlining. The initial subordinate linked to port 502 is visually indicated by a red underline. The second slave, which is connected to port 503,

is visually emphasized by the use of an orange underline. By providing the names of the proxy and reverse proxy servers, it becomes possible to obscure the identities of the master and slave servers. This enables the slaves, which are linked to port 502, to handle the requests made by the master server connected to port 503. This procedure provides the chance to obscure the detectability of a component within a system, therefore augmenting the intricacy of the system and perplexing an adversary's efforts to undermine its security. The proposed approach involves employing a similar methodology to that of the offender, wherein the concealment of their identity is pursued, mirroring the efforts of attackers to obfuscate their presence following a successful system hack. Furthermore, this procedure creates supplementary prospects that may be employed to equilibrate the workload. Lastly, this procedure may be employed to consolidate the network and provide network admins with a comprehensive perspective of system operations.

**Figure 60**

***Processing module/Proxy Interface***



***Discussions***

Phase 3 of the framework encompasses a detailed elucidation of both the parsing of the model's output and the comprehensiveness of the PCAP data. The fourth step of the design encompasses the utilization of the model for the purpose of generating many instances of

clients and servers, facilitating communication to stimulate traffic, and obtaining network data. Adversaries use the vulnerabilities included in the TCP Modbus protocol to initiate attacks targeting the Modbus network. One effective strategy for safeguarding a network utilizing this technology is adopting a defensive posture that mirrors the tactics employed by potential attackers. The identity of the network will remain undisclosed to potential assailants due to the utilization of Modbus data traffic centers, which will function as both proxy and reverse proxy centers. This will enable the network to effectively protect itself against targeted attacks on SCADA systems. Moreover, Modbus data traffic centers include the capability to consolidate all network connections, resulting in enhanced network visibility and a single monitoring point for all traffic. As a result, the administrator will have enhanced visibility into their network, even if the network will become more intricate and obscured from potential adversaries. Data traffic centers equipped with monitoring capabilities that utilize the Modbus protocol engage in the analysis of packet information and traffic speed to detect and mitigate potential assaults, while also ensuring sufficient capacity for operational requirements. This facilitates the unimpeded movement of traffic and ensures the continued functionality of the network.



## CHAPTER 5 - CONCLUSIONS

The integration of IT and OT networks and IoT devices in recent years has exposed modern SCADA systems to new threats and vulnerabilities. Numerous analyses of malware that target industrial corporations have identified these weaknesses. In addition, the continued success of cyberattacks and the escalating criticism of how cybersecurity risk is communicated condemn the current state of affairs. To address these issues, our research compiled critical infrastructure insights from theory and practice and developed a set of evidence-based design principles, a design science perspective on secure critical infrastructures, using an offensive security approach. The four-stage ABCD risk assessment framework is the one this study recommends. First, a comprehensive overview of the SCADA system's architecture, devices, activity log, and transactional data is compiled. The second phase proposed protocols for risk management in SCADA, emphasizing the security measures provided by the protocols, their expansions, and their global dissemination. In addition, the research evaluated new risk management strategies for the SCADA industry by analyzing the numerous attacks. To achieve this, the model of the framework categorized the test datasets based on their functionality and outlined the numerous obstacles and requirements to consider during the third phase. Lastly, the fourth phase will attempt to defend against or delay the attacks by employing the same techniques that the perpetrator used to attack the SCADA system. The ABCD is a risk-manageable framework that uses a consistent and methodical process to identify, assess, and monitor risks. Phase 1's brainstorming interfaces enable the identification of relevant SCADA system-related risks. Once a risk has been identified, there must be a method for assessing the risk's impact on the system, which is what phase 2 of the framework provides using the two criteria of probability and impact. Risk probability describes the likelihood that a risk will be realized, whereas risk impact describes the severity of the risk's influence on the system if it is realized. These criteria are evaluated using a quantitative evaluation. Treating the risk at the component level and then the attack path that exposes the risk. Every risk will require an appropriate, realizable, and cost-effective response, as well as continuous process monitoring. By monitoring the activity and response to hazards, phases 3 and 4 of the framework aim to achieve this objective.

As demonstrated in the preceding chapter, the existing data on SCADA systems is incomplete and does not provide an evaluation of the vulnerabilities present in SCADA designs, problems with their administration, or the efficacy of the established security measures. The ABCD risk management framework in experiment chapter of this research, will significantly enhance the overall security of the SCADA system. This approach provides a holistic perspective that analyzes the security of SCADA systems. By enhancing the visibility of SCADA components and assessing the risk score of these components based on attack paths, the security of SCADA systems can be enhanced. The framework provides a methodology for organizing SCADA components, which successfully identifies vulnerabilities during the initial evaluation phase. Identify risks by developing a thorough understanding of the organization to successfully manage cybersecurity hazards related to systems, persons, resources, data, and competencies. The duties associated with the process of hazard identification are vital for efficiently managing risk, as it is impracticable to address or reduce issues that are unknown or not apparent. To ensure successful prioritization and alignment with risk management strategy and business demands, an organization must have a thorough understanding of the business environment, the resources that support critical functions, and the corresponding cybersecurity threats. Utilizing a thorough evaluation technique to ensure system security helps in directing the formulation and implementation of appropriate actions to guarantee the supply of key services. The safeguard feature of the framework enables the ability to limit or contain the impact of a potential cybersecurity issue, as demonstrated in the experiment conducted during the second phase. The third stage of the framework offers techniques for detecting and evaluating possible risks. Identifying potential threats will enable the development and implementation of appropriate procedures to detect the initiation of a cybersecurity incident. The Detect Function enables the swift identification of cybersecurity incidents. The fourth phase of the framework offers a systematic approach to identifying and reducing risks. This entails the creation and execution of appropriate measures to tackle a recognized cybersecurity incident. The Respond Function helps to mitigate the potential repercussions of a cybersecurity incident. The experimentation conducted involved the utilization of diverse SCADA systems from different generations. The framework employed in these experiments successfully identified vulnerabilities, detected abnormal activities, and effectively mitigated and reduced cyber risks to the system.

The nomenclature ABCD was formed by taking the first letter of each stage name inside the framework. Specifically, A denotes assessing, B denotes blocking, C denotes catching, and D denotes defending. An alternative interpretation of the nomenclature of the framework pertains to its inception or commencement. When individuals see the acronym ABCD, it is commonly associated with the initial quartet of letters in the English alphabet. The framework was developed with the explicit purpose of serving as a first step and a fundamental basis for further development. The architecture of SCADA systems is specifically built to accommodate development and growth, as these systems continuously evolve in response to technological advancements. The initial stage of the framework involves the systematic gathering of data and input, which subsequently evolves into a more intricate process over time. The framework operates holistically to proactively address, diminish, or alleviate cyber danger and bolster the overall security of SCADA systems, whether through a straightforward or intricate approach. The study has successfully incorporated risk assessment services into a realistic SCADA scenario. It has also proved the effectiveness of the ABCD approach in evaluating cyber-risks associated with complex systems, including SCADA infrastructure. Furthermore, the study has shown that the ABCD technique can effectively guide risk mitigation efforts.

As stated in the justification portion of the evaluation, this framework now only functions with Modbus TCP. In addition, the inclusion of the processing module in phase four of the framework will be a valuable addition to the SCADA architecture. However, there may be an unintended downside in terms of speed, as the module will process the data before delivering it to the target client or server. Adding a processing module for query instructions would be highly beneficial, as it would enhance response speed by processing data at the module level rather than at the target server level. However, the update command will experience a slight decrease in performance since the module must first process the request before forwarding it to the destination server. Modbus TCP typically transmits at the same speed as the Ethernet connection. The majority of Modbus TCP devices operate at a speed of 100 MB, whereas a small number of older devices operate at a speed of 10 MB. At such high velocities, the limiting factor typically lies in the processing speed of the connection's endpoints to handle the messages. This factor ensures that the addition of the process module should not significantly impact the overall performance of the communication. In addition,

adding a processing module to the network before a key part of the SCADA system will be helpful because it will extend the attack path that leads to the component. This module will also function as a safeguard, providing an additional layer of security for the important component that we are seeking to secure. However, incorporating an extra component into the system will introduce another element of the network, potentially creating a new pathway for attacks and serving as an attack vector that can target the SCADA system. Since the processing module doesn't operate with an open protocol, we can implement additional measures to strengthen its protection. The framework can be used for both static and dynamic purposes. During static mode, communications are uploaded, whereas in dynamic mode, communications are examined in real-time. When integrating cybersecurity safeguards into the SCADA system, it is essential to meticulously assess the balance between the advantages and the potential impact on performance and security. When a component is important, it is more beneficial to accept a slight decrease in performance and security complexity in order to guarantee the safety of the component. Using the framework in static mode will reduce cyber security threats without affecting system performance or security.

Due to the essential nature of the SCADA system, the ABCD risk management framework can only be developed and validated in a simulation environment. In the future, this study can be expanded to be implemented in a production setting. Furthermore, although further effort is needed to enable the framework sandbox to completely replicate a comprehensive SCADA network, As mentioned earlier in this study, SCADA systems can support various protocols, including Modbus, Meter-Bus (M-BUS), Simple Network Management Protocol (SNMP), Distributed Network Protocol 3 (DNP3), and Building Automation Controls Network (BACnet), depending on the specific technological needs. This research exclusively concentrates on Modbus protocols; however, it has the potential to be extended to incorporate other protocols. Further investigation can be undertaken to explore supplementary SCADA communication methods and the incorporation of incident management protocols to assess and control the amalgamation of physical and cyber vulnerabilities on such infrastructure. By combining stimulation and production, the hybrid method makes it possible to do research and training on SCADA network security without stopping the production SCADA system from normal operation. An issue that arises with SCADA infrastructure is the lack of standardized models to verify this emulation method and

subsequently provide security remedies. Although there exist some models, none of them incorporate the integration of computer networks and SCADA topologies. Utilizing the SCADA sandbox in an experimental research setting is the subsequent rational progression. The generated data sets has the capacity to become valuable resources for the research community, which presently faces a scarcity of publicly accessible sources. Furthermore, by extending the implementation of the sandbox to encompass additional SCADA settings, such as those found in the oil and gas, water supply, or industrial sectors, researchers would have the opportunity to investigate and resolve challenges that are unique to those particular systems. The framework is designed with the purpose of serving as a fundamental basis for future expansion, as implied by its name. The framework provides comprehensive coverage of all critical security domains and facilitates the creation of customized policies specific to each SCADA operation. Nevertheless, the prospects of the future are dependent on two key factors: the accessibility of technology and the level of public concern. The implementation of a defense-in-depth strategy and the adoption of proactive measures are crucial for bolstering the security of SCADA control systems, hence ensuring the resilience of control systems and critical infrastructure. It is expected that the first set of principles will undergo refinement as research progresses, further research findings and practitioner literature are incorporated, a broader spectrum of infrastructure specialists are contacted, and the design process is iterated. This study holds the potential to serve as a valuable resource for future scholars, who may utilize its findings as a foundation for further investigation and development.

## REFERENCES

- Abdulwahid, M. M., Abdullah, H. K., Ateah, W. M., & Ahmed, S. (2023). Implementation of Automated Water based Level Management Model by using SCADA system and PLC. *Journal of Energy Engineering and Thermodynamics (JEET) ISSN 2815-0945*, 3(03), 40-51.
- Abou el Kalam, A. (2021). Securing SCADA and critical industrial systems: From needs to security mechanisms. *International Journal of Critical Infrastructure Protection*, 32, 100394.
- Aken, J. E. v. (2004). Management research based on the paradigm of the design sciences: the quest for field-tested and grounded technological rules. *Journal of management studies*, 41(2), 219-246.
- Al-Dalky, R., Abduljaleel, O., Salah, K., Otrok, H., & Al-Qutayri, M. (2014). A Modbus traffic generator for evaluating the security of SCADA systems. 2014 9th International Symposium on Communication Systems, Networks & Digital Sign (CSNDSP),
- Astolfi, D. (2021). Perspectives on SCADA Data Analysis Methods for Multivariate Wind Turbine Power Curve Modeling. *Machines*, 9(5), 100.
- Baiardi, F., Telmon, C., & Sgandurra, D. (2009). Hierarchical, model-based risk management of critical infrastructures. *Reliability Engineering & System Safety*, 94(9), 1403-1415.
- Beggs, C., & Warren, M. (2009). Safeguarding Australia from cyber-terrorism: a proposed cyber-terrorism SCADA risk framework for industry adoption.
- Butun, I., Österberg, P., & Song, H. (2019). Security of the Internet of Things: Vulnerabilities, attacks, and countermeasures. *IEEE Communications Surveys & Tutorials*, 22(1), 616-644.
- Byres, E., & Lowe, J. (2004). The myths and facts behind cyber security risks for industrial control systems. Proceedings of the VDE Kongress,
- Byres, E. J., Franz, M., & Miller, D. (2004). The use of attack trees in assessing vulnerabilities in SCADA systems. Proceedings of the international infrastructure survivability workshop,
- Cappers, B. C., Meessen, P. N., Etalle, S., & Van Wijk, J. J. (2018). Eventpad: Rapid malware analysis and reverse engineering using visual analytics. 2018 IEEE Symposium on Visualization for Cyber Security (VizSec),
- Cherdantseva, Y., Burnap, P., Blyth, A., Eden, P., Jones, K., Soulsby, H., & Stoddart, K. (2016). A review of cyber security risk assessment methods for SCADA systems. *Computers & Security*, 56, 1-27.  
<https://doi.org/https://doi.org/10.1016/j.cose.2015.09.009>
- Chittester, C. G., & Haines, Y. Y. (2004). Risks of terrorism to information technology and to critical interdependent infrastructures. *Journal of Homeland Security and Emergency Management*, 1(4).
- Chochtoula, D., Ilias, A., Stamatiou, Y. C., & Makris, C. (2022). Integrating Elliptic Curve Cryptography with the Modbus TCP SCADA Communication Protocol. *Future Internet*, 14(8), 232.
- Choo, K.-K. R. (2011). The cyber threat landscape: Challenges and future research directions. *Computers & security*, 30(8), 719-731.

- Cifranic, N., Hallman, R. A., Romero-Mariona, J., Souza, B., Calton, T., & Coca, G. (2020). Decepti-SCADA: A cyber deception framework for active defense of networked critical infrastructures. *Internet of Things*, 12, 100320.
- Cárdenas, A. A., Amin, S., Lin, Z.-S., Huang, Y.-L., Huang, C.-Y., & Sastry, S. (2011). Attacks against process control systems: risk assessment, detection, and response. Proceedings of the 6th ACM symposium on information, computer and communications security,
- Duggan, D. (2005). *Penetration Testing of Industrial Control Systems*.  
[https://energy.sandia.gov/wp-content/gallery/uploads/sand\\_2005\\_2846p.pdf](https://energy.sandia.gov/wp-content/gallery/uploads/sand_2005_2846p.pdf)
- FernhillSoftware. (2012). *Modbus Protocol Overview with Examples*.  
<https://www.fernhillsoftware.com/help/drivers/modbus/modbus-protocol.html>
- Figuerola-Lorenzo, S., Añorga, J., & Arrizabalaga, S. (2019). A role-based access control model in modbus SCADA systems. A centralized model approach. *Sensors*, 19(20), 4455.
- Francia III, G. A., Thornton, D., & Dawson, J. (2012). Security best practices and risk assessment of SCADA and industrial control systems. Proceedings of the international conference on security and management (SAM),
- Frazão, I., Abreu, P. H., Cruz, T., Araújo, H., & Simões, P. (2018). Denial of service attacks: Detecting the frailties of machine learning algorithms in the classification process. International Conference on Critical Information Infrastructures Security,
- Gertman, D. I., Folkers, R., & Roberts, J. (2006). *Scenario-based approach to risk analysis in support of cyber security*.
- Guan, J., Graham, J. H., & Hieb, J. L. (2011). A digraph model for risk identification and mangement in SCADA systems. Proceedings of 2011 IEEE International Conference on Intelligence and Security Informatics,
- Henry, M. H., & Haimes, Y. Y. (2009). A comprehensive network security risk model for process control networks. *Risk Analysis: An International Journal*, 29(2), 223-248.
- Henry, M. H., Layer, R. M., Snow, K. Z., & Zaret, D. R. (2009). Evaluating the risk of cyber attacks on SCADA systems via Petri net analysis with application to hazardous liquid loading operations. 2009 IEEE Conference on Technologies for Homeland Security,
- Hewett, R., Rudrapattana, S., & Kijsanayothin, P. (2014). Cyber-security analysis of smart grid SCADA systems with game models. Proceedings of the 9th annual cyber and information security research conference,
- Holkovič, M., Ryšavý, O., & Dudek, J. (2019). Automating network security analysis at packet-level by using rule-based engine. Proceedings of the 6th Conference on the Engineering of Computer Based Systems,
- Hoppa, M. A. (2023). Understanding Cybersecurity Risks in Offshore Wind Farms.
- Howard, M. (2021). 2021 State of the Software Supply Chain: Open Source Security and Dependency Management Take Center Stage.
- Huq, N., Hilt, S., & Hellberg, N. (2017). US cities exposed: Industries and ICS. *A shodan-based security study of exposed systems and infrastructure in the US*.
- ISO. (2019). *IEC 31010:2019*. @isostandards. <https://www.iso.org/standard/72140.html>
- Jakaboczki, G., & Adamko, E. (2015). Vulnerabilities Of Modbus Rtu Protocol—A Case Study. *Nnals Of The Oradea University, Fascicle Of Management And Technological Engineering*, 1.
- Kaitai Struct: declarative binary format parsing language*. (2015). GitHub. <https://kaitai.io/>

- Kim, D.-S., & Tran-Dang, H. (2019). Implementing Modbus and CAN Bus Protocol Conversion Interface. *Industrial Sensors and Controls in Communication Networks: From Wired Technologies to Cloud Computing and the Internet of Things*, 65-72.
- Kour, R., Thaduri, A., & Karim, R. (2020). Railway Defender Kill Chain to Predict and Detect Cyber-Attacks. *9*, 47-90. <https://doi.org/10.13052/jcsm2245-1439.912>
- Kriaa, S., Bouissou, M., & Piètre-Cambacédès, L. (2012). Modeling the Stuxnet attack with BDMP: Towards more formal risk assessments. 2012 7th International Conference on Risks and Security of Internet and Systems (CRiSIS),
- Kuchar, K., Holasova, E., Fujdiak, R., Blazek, P., & Misurec, J. (2022). Incident Detection System for Industrial Networks. In *Big Data Privacy and Security in Smart Cities* (pp. 83-102). Springer.
- Kwame, A. E., Martey, E. M., & Chris, A. G. (2017). Qualitative assessment of compiled, interpreted and hybrid programming languages. *Communications*, 7(7), 8-13.
- Lake, S. (2022). The cybersecurity industry is short 3.4 million workers—that's good news for cyber wages. *Fortune*.
- Lamba, A., Singh, S., Balvinder, S., Dutta, N., & Rela, S. (2017). Mitigating cyber security threats of industrial control systems (scada & dcs). 3rd International Conference on Emerging Technologies in Engineering, Biomedical, Medical and Science (ETEBMS—July 2017),
- Lee, J.-M., & Hong, S. (2020). Keeping host sanity for security of the SCADA systems. *IEEE Access*, 8, 62954-62968.
- LeMay, E., Unkenholz, W., Parks, D., Muehrcke, C., Keefe, K., & Sanders, W. H. (2010). Adversary-driven state-based system security evaluation. Proceedings of the 6th International Workshop on Security Measurements and Metrics,
- Letavay, V., Pluskal, J., & Ryšavý, O. (2019). Network Forensic Analysis for Lawful Enforcement on Steroids, Distributed and Scalable. Proceedings of the 6th Conference on the Engineering of Computer Based Systems,
- Leyva, F. R., Cuellar, J. A., Basilio, R. M., & Justo, E. E. (2004). Wireless system for electrical networks testing based on MODBUS protocol. 14th International Conference on Electronics, Communications and Computers, 2004. CONIELECOMP 2004.,
- Maier, D., Müller, T., & Protsenko, M. (2014). Divide-and-conquer: Why android malware cannot be stopped. 2014 Ninth International Conference on Availability, Reliability and Security,
- McQueen, M. A., Boyer, W. F., Flynn, M. A., & Beitel, G. A. (2006). Time-to-compromise model for cyber risk reduction estimation. In *Quality of protection* (pp. 49-64). Springer.
- Mijwil, M., & Aljanabi, M. (2023). Towards artificial intelligence-based cybersecurity: the practices and ChatGPT generated ways to combat cybercrime. *Iraqi Journal For Computer Science and Mathematics*, 4(1), 65-70.
- Miller, B., & Rowe, D. (2012). A survey SCADA of and critical infrastructure incidents. Proceedings of the 1st Annual conference on Research in information technology,
- Mirzaev, R. (2021). Unsupervised Progressive Anomaly Detection for Network Traffic.
- Modbus. (2012). *MODBUS Messaging Implementation Guide 1 0 b - Modbus\_Messaging\_Implementation\_Guide\_V1\_0b.pdf*. [https://modbus.org/docs/Modbus\\_Messaging\\_Implementation\\_Guide\\_V1\\_0b.pdf](https://modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf)



- Morris, T., & Gao, W. (2013). *Industrial Control System Cyber Attacks*.  
<https://doi.org/10.14236/ewic/ICSCSR2013.3>
- Mouton, J. (2001). *How to succeed in your master's and doctoral studies: A South African guide and resource book*. Van Schaik.
- NBCNews. (2022). A record 4.5 million people quit their jobs in November.
- NozomiNetworks. (2023). *OT/IoT Security Report - A Deep Look Into the ICS Threat Landscape*. <https://www.nozominetworks.com/downloads/Nozomi-Networks-OT-IoT-Security-Report-2022-2H.pdf>
- OWASP. (2018). *OWASP IoT Top 10 2018*. s. Retrieved 2/7/2022 from  
<https://owasp.org/www-pdf-archive/OWASP-IoT-Top-10-2018-final.pdf>
- Pal, S., & Jadidi, Z. (2021). Analysis of Security Issues and Countermeasures for the Industrial Internet of Things. *Applied Sciences*, 11(20), 9393.
- Panguluri, S., Nelson, T. D., & Wyman, R. P. (2017). Creating a cyber security culture for your water/waste water utility. *Cyber-Physical Security: Protecting Critical Infrastructure at the State and Local Level*, 133-159.
- Patel, S., & Zaveri, J. (2010). A risk-assessment model for cyber attacks on information systems. *J. Comput.*, 5(3), 352-359.
- Patel, S. C., Graham, J. H., & Ralston, P. A. (2008). Quantitatively assessing the vulnerability of critical information systems: A new method for evaluating security enhancements. *International Journal of Information Management*, 28(6), 483-491.
- PENNSYLVANIA, C. O. (2022). § 302.1208. *PLCs and SCADA*.  
<http://www.pacodeandbulletin.gov/Display/pacode?file=/secure/pacode/data/025/chapter302/s302.1208.html&d=reduce>
- Permann, M. R., & Rohde, K. (2005). *Cyber assessment methods for SCADA security*.
- Program, P.-P. A. E. (2017). *11 Supply Chain Risks of SCADA Industrial Control Systems in the Electricity Sector\_Risks and Mitigations.pdf*.  
[https://www.dni.gov/files/PE/Documents/11---Supply-Chain-Risks-of-SCADA-Industrial-Control-Systems-in-the-Electricity-Sector\\_Risks-and-Mitigations.pdf](https://www.dni.gov/files/PE/Documents/11---Supply-Chain-Risks-of-SCADA-Industrial-Control-Systems-in-the-Electricity-Sector_Risks-and-Mitigations.pdf)
- Rahman, A., Mustafa, G., Khan, A. Q., Abid, M., & Durad, M. H. (2022). Launch of denial of service attacks on the modbus/TCP protocol and development of its protection mechanisms. *International Journal of Critical Infrastructure Protection*, 39, 100568.
- Rajesh, L., & Satyanarayana, P. (2021). Detection and blocking of replay, false command, and false access injection commands in scada systems with modbus protocol. *Security and Communication Networks*, 2021.
- Ravindranath, R. S. (2009). *Smartgrid SCADA system security issues and counter measures* [California State University, Sacramento].
- Rawat, R., Mahor, V., Garg, B., Chouhan, M., Pachlasiya, K., & Telang, S. (2022). Modeling of cyber threat analysis and vulnerability in IoT-based healthcare systems during COVID. In *Lessons from COVID-19* (pp. 405-425). Elsevier.
- Richards, G. (2008). Hackers vs slackers-[control security]. *Engineering & Technology*, 3(19), 40-43.
- Romme, A. G. L., & Endenburg, G. (2006). Construction principles and design rules in the case of circular design. *Organization science*, 17(2), 287-297.
- Ross, S. (2009). *Probability and statistics for engineers and scientists* (Vol. 16). Elsevier, New Delhi.

- Roy, A., Kim, D. S., & Trivedi, K. S. (2010). Cyber security analysis using attack countermeasure trees. Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research,
- Sadeeq, M. A., Zeebaree, S. R., Qashi, R., Ahmed, S. H., & Jacksi, K. (2018). Internet of Things security: a survey. 2018 International Conference on Advanced Science and Engineering (ICOASE),
- Sajid, A., Abbas, H., & Saleem, K. (2016). Cloud-assisted IoT-based SCADA systems security: A review of the state of the art and future challenges. *IEEE Access*, 4, 1375-1384.
- Sanchez, G. (2019). Man-In-The-Middle Attack Against Modbus TCP Illustrated with Wireshark. *SANS Institute*.
- Schumacher, S., & McMillan, J. H. (1993). Research in education: A conceptual introduction. SecurityScoreCard. (2022). *Addressing the Trust Deficit in Critical Infrastructure*. SecurityScoreCard. <https://securityscorecard.com/all/addressing-the-trust-deficit>
- Shukla, P., Singh, S., Joshi, T., Kumar, S., Kelkar, S., Das, M. R., & Moudgalya, K. M. (2017). Design and development of a MODBUS automation system for industrial applications. 2017 6th International Conference on Computer Applications In Electrical Engineering-Recent Advances (CERA),
- Simon, H. (1996). The Sciences of Artificial, Cambridge MA and London. *Published online*.
- Singh, V. K., & Govindarasu, M. (2021). Cyber Kill Chain-Based Hybrid Intrusion Detection System for Smart Grid. In *Wide Area Power Systems Stability, Protection, and Security* (pp. 571-599). Springer.
- Song, J.-G., Lee, J.-W., Lee, C.-K., Kwon, K.-C., & Lee, D.-Y. (2012). A cyber security risk assessment for the design of I&C systems in nuclear power plants. *Nuclear engineering and technology*, 44(8), 919-928.
- Stouffer, K., Falco, J., & Scarfone, K. (2011). *NIST Special Publication 800-82, Guide to Industrial Control Systems (ICS) Security*.
- Swales, A. (1999). Open modbus/tcp specification. *Schneider Electric*, 29(3), 19.
- SynSaber. (2022). SynSaber Releases ICS Vulnerabilities & CVEs Report Covering Second Half of 2022.
- Taylor, C., Krings, A., & Alves-Foss, J. (2002). Risk analysis and probabilistic survivability assessment (RAPSA): An assessment approach for power substation hardening. Proc. ACM Workshop on Scientific Aspects of Cyber Terrorism,(SACT), Washington DC,
- Ten, C., Liu, C., & Manimaran, G. (2008). Vulnerability Assessment of Cybersecurity for SCADA Systems. *IEEE Transactions on Power Systems*, 23(4), 1836-1846. <https://doi.org/10.1109/TPWRS.2008.2002298>
- Ten, C.-W., Manimaran, G., & Liu, C.-C. (2010). Cybersecurity for critical infrastructures: Attack and defense modeling. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 40(4), 853-865.
- Trellix. (2023). The Threat Report: February 2023.
- Ujvarosi, A. (2016). Evolution of SCADA systems. *Bulletin of the Transilvania University of Brasov. Engineering Sciences. Series I*, 9(1), 63.
- Urias, V., Leeuwen, B. V., & Richardson, B. (2012, 29 Oct.-1 Nov. 2012). Supervisory Command and Data Acquisition (SCADA) system cyber security analysis using a live, virtual, and constructive (LVC) testbed. MILCOM 2012 - 2012 IEEE Military Communications Conference,

- Vaishnavi, V., & Kuechler, W. (2004). Design research in information systems.
- Viega, J., & Thompson, H. (2012). The state of embedded-device security (spoiler alert: It's bad). *IEEE Security & Privacy*, 10(5), 68-70.
- vom Brocke, J., Hevner, A., & Maedche, A. (2020). Introduction to design science research. *Design science research. Cases*, 1-13.
- Wang, J., Constante Flores, G., Moya, C., & Hong, J. (2020). A Semantic Analysis Framework for Protecting the Power Grid Against Monitoring-Control Attacks. *IET Cyber-Physical Systems: Theory & Applications*, 5. <https://doi.org/10.1049/iet-cps.2019.0026>
- Wang, Y., Zhang, T., & Ye, Q. (2021). Situation awareness framework for industrial control system based on cyber kill chain. MATEC Web of Conferences,
- Woo, P. S., & Kim, B. H. (2014). A study on quantitative methodology to assess cyber security risk of SCADA systems. *Advanced Materials Research*,
- Yan, J., Govindarasu, M., Liu, C.-C., & Vaidya, U. (2013). A PMU-based risk assessment framework for power control systems. 2013 IEEE Power & Energy Society General Meeting,
- Zhu, B., & Sastry, S. (2010). SCADA-specific intrusion detection/prevention systems: a survey and taxonomy. *Proceedings of the 1st workshop on secure control systems (SCS)*,
- Zografopoulos, I., Ospina, J., Liu, X., & Konstantinou, C. (2021). Cyber-physical energy systems security: Threat modeling, risk assessment, resources, metrics, and case studies. *IEEE Access*, 9, 29775-29818.

## APPENDIX A: USERS' MANUAL

**Component Groups:** The provided table categorizes various components into several groups, including field instruments, databases, human-machine interfaces (HMIs), field controllers, and network connection. The table comprises three columns, namely the component group name, a description of the component group, and an indicator of whether the component group is affiliated with an IT network or an OT network. network.

**Figure 61**

*An illustration of component groups*

ID	Name	Description	ITorOT
1	Field instrumentation	The array of monitors and transmitters on the factory floor that SCADA applications use	OT
2	Field controllers	RTUs and PLCs collect and compile data supplied by field instrumentation, preparing it for display and a...	OT
3	Human-machine interface	Master units that allow humans to supervise the SCADA data acquisition process	OT
4	Network connectivity	The SCADA system relies on maintaining integrated network connectivity throughout its operation	OT
5	Database or historian	Physically and digitally secure places to store data gathered, analyzed and processed by SCADA system	OT

**Component Types:** The provided table presents a comprehensive categorization and detailed analysis of the many sorts of components. A wide range of field instruments, such as sensors, actuators, and temperature monitors, are categorized into component categories. The table is comprised of many columns, which encompass the categorization group ID (making reference to the Component Groups database), the component type's name, and a concise description of the component type.

**Figure 62***An illustration of component types*

ID	ComponentGroupID	Name	Description
56	3	SCADA Master	The master station displays the acquired data and also allows the operator to perform remote control tasks.
57	3	Remote Terminal Unit (RTU)	A remote terminal unit (RTU) is a microprocessor-controlled electronic device that interfaces objects in the physica...
58	4	Wide Area Network (WAN)	Independent Ignition Gateways provide local control and SCADA functionality.
59	3	Operating Stations	
60	4	LAN	
61	2	Communication Server	
63	3	PC	
66	5	Historian	
70	1	Field Devices	
71	4	Other Control Center	
72	2	PMU	
73	2	PLC	
76	4	Corporate LAN	
77	4	Firewall	
78	4	AMI	
80	4	Port Server	
82	3	Application Servers	
83	4	Communication Links	
84	4	Redundant LAN	
86	3	HMI	Human Machine Interface, often known by the acronym HMI, refers to a dashboard or screen used to control mach...

**Components:** The table shown above comprises discrete components of a SCADA system, including two HMIs situated within a given facility, a sensor identified by its XYZ Mac address and RTX IP address, among other components. The table comprises many columns, encompassing the kind of component (according to the Component kind database), the name of the instrument, a concise description of the component, the icon picture associated with the component, its MAC address, and its IP address.

**Figure 63***An illustration of components*

ID	ComponentTypeID	Name	Description	Color	ImageDisplay	MACAddress	IPAddress
1	56	SCADA Master	The master station displays the acquired d...	#FF8080	<Binary data>	NULL	NULL
2	57	Remote Terminal Unit (RTU)	A remote terminal unit (RTU) is a micropr...	#004080	<Binary data>	NULL	NULL
3	58	Wide Area Network (WAN)	Independent Ignition Gateways provide Io...	Navy	<Binary data>	NULL	NULL
4	59	Operating Stations		#FF8000	<Binary data>	NULL	NULL
5	60	LAN		Lime	<Binary data>	NULL	NULL
6	61	Communication Server		#80FF00	<Binary data>	NULL	NULL
7	63	PC 1		Aqua	<Binary data>	00-0C-29-E6-14...	172.27.224.250
8	66	Historian		Silver	<Binary data>	NULL	NULL
9	70	Field Devices		Blue	<Binary data>	NULL	NULL
10	71	Other Control Center			<Binary data>	NULL	NULL
11	72	PMU			<Binary data>	NULL	NULL
12	73	PLC			<Binary data>	NULL	NULL
13	76	Corporate LAN		Blue	<Binary data>	NULL	NULL
14	77	Firewall		#FF8040	<Binary data>	NULL	NULL
15	78	AMI		Yellow	<Binary data>	NULL	NULL
16	80	Port Server		Gray	<Binary data>	NULL	NULL
17	82	Application Servers		#8080FF	<Binary data>	NULL	NULL
18	83	Communication Links			<Binary data>	NULL	NULL
19	84	Redundant LAN			<Binary data>	NULL	NULL
20	86	HMI	Human Machine Interface, often known by...	Blue	<Binary data>	NULL	NULL
21	63	PC 2		Aqua	NULL	00-80-F4-09-51...	172.27.224.50

**Information:** The table shown above serves as a tool for network administrators to effectively monitor and track pertinent information and actions inside the environment. The table comprises multiple columns encompassing the information type's name, a concise description of the information type, the system's analysis requirement for the information, the file's content, the specific information's path, and the specific program's path for a vital piece of information.

**Figure 64***A depiction of upload logs for a certain component.*

ID	Name	Description	Analyze	FileContent	FilePath	ProgramPath	Text
1	Window Log	Log Transaction from Jun	True	NULL	NULL	NULL	

**File Information:** If a file is designated for analysis in the aforementioned table, its data will undergo analysis and be placed in the file information table. As indicated in the research, our focus will be directed on the Modbus protocol. The table is equipped with many columns designed to store data extracted from the PCAP file.

**Figure 65**

*A demonstration of extracted PCAP data*

PacketNo	Time	SourceIP	DestinationIP	SourcePort	Destination...	Protocol	Length	SourceMacAddress	DestinationMacA...	OrigLen	TsSec	TsUsec	EtherType	Data	FileName	Trar
17414	1	172.27.224.250	172.27.224.50	502	53762	Tcp	0	00-80-F4-09-51-38	00-0C-29-E6-14-0D	60	1526987267	654024	Ipv4	<Binary dat...	C:\Users\M...	NUL
17415	1	172.27.224.250	172.27.224.250	53762	502	Tcp	0	00-0C-29-E6-14-0D	00-80-F4-09-51-38	60	1526987267	654076	Ipv4	<Binary dat...	C:\Users\M...	NUL
17416	1	172.27.224.250	172.27.224.50	502	53760	Tcp	0	00-80-F4-09-51-38	00-0C-29-E6-14-0D	60	1526987267	654239	Ipv4	<Binary dat...	C:\Users\M...	NUL
17417	1	172.27.224.50	172.27.224.250	53760	502	Tcp	0	00-0C-29-E6-14-0D	00-80-F4-09-51-38	60	1526987267	654290	Ipv4	<Binary dat...	C:\Users\M...	NUL
17418	1	172.27.224.250	172.27.224.50	502	53758	Tcp	0	00-80-F4-09-51-38	00-0C-29-E6-14-0D	60	1526987267	654443	Ipv4	<Binary dat...	C:\Users\M...	NUL
17419	1	172.27.224.50	172.27.224.250	53758	502	Tcp	0	00-0C-29-E6-14-0D	00-80-F4-09-51-38	60	1526987267	654493	Ipv4	<Binary dat...	C:\Users\M...	NUL
17420	1	172.27.224.250	172.27.224.50	502	53756	Tcp	0	00-80-F4-09-51-38	00-0C-29-E6-14-0D	60	1526987267	654707	Ipv4	<Binary dat...	C:\Users\M...	NUL
17421	1	172.27.224.50	172.27.224.250	53756	502	Tcp	0	00-0C-29-E6-14-0D	00-80-F4-09-51-38	60	1526987267	654756	Ipv4	<Binary dat...	C:\Users\M...	NUL
17422	1	172.27.224.50	172.27.224.250	53770	502	Tcp	12	00-0C-29-E6-14-0D	00-80-F4-09-51-38	66	1526987267	661476	Ipv4	<Binary dat...	C:\Users\M...	1161
17423	1	172.27.224.250	172.27.224.50	502	53754	Tcp	0	00-80-F4-09-51-38	00-0C-29-E6-14-0D	60	1526987267	663191	Ipv4	<Binary dat...	C:\Users\M...	NUL
17424	1	172.27.224.50	172.27.224.250	53754	502	Tcp	0	00-0C-29-E6-14-0D	00-80-F4-09-51-38	60	1526987267	663267	Ipv4	<Binary dat...	C:\Users\M...	NUL
17425	1	172.27.224.250	172.27.224.50	502	53752	Tcp	0	00-80-F4-09-51-38	00-0C-29-E6-14-0D	60	1526987267	663448	Ipv4	<Binary dat...	C:\Users\M...	NUL
17426	1	172.27.224.50	172.27.224.250	53752	502	Tcp	0	00-0C-29-E6-14-0D	00-80-F4-09-51-38	60	1526987267	663510	Ipv4	<Binary dat...	C:\Users\M...	NUL
17427	1	172.27.224.250	172.27.224.50	502	53750	Tcp	0	00-80-F4-09-51-38	00-0C-29-E6-14-0D	60	1526987267	663768	Ipv4	<Binary dat...	C:\Users\M...	NUL
17428	1	172.27.224.50	172.27.224.250	53750	502	Tcp	0	00-0C-29-E6-14-0D	00-80-F4-09-51-38	60	1526987267	663828	Ipv4	<Binary dat...	C:\Users\M...	NUL
17429	1	172.27.224.250	172.27.224.50	502	53748	Tcp	0	00-80-F4-09-51-38	00-0C-29-E6-14-0D	60	1526987267	664000	Ipv4	<Binary dat...	C:\Users\M...	NUL
17430	1	172.27.224.50	172.27.224.250	53748	502	Tcp	0	00-0C-29-E6-14-0D	00-80-F4-09-51-38	60	1526987267	664063	Ipv4	<Binary dat...	C:\Users\M...	NUL
17431	1	172.27.224.250	172.27.224.50	502	53746	Tcp	0	00-80-F4-09-51-38	00-0C-29-E6-14-0D	60	1526987267	664241	Ipv4	<Binary dat...	C:\Users\M...	NUL

**Component Information:** The table shown herein displays the correlation between the component and its corresponding information. The table consists of three columns. The three IDs are the connection identifier, the componentID which references the component table, and the informationID which references the information table.

**Figure 66**

*A depiction the correlation between components and information*

ID	ComponentID	InformationID
1	63	1

**Model:** The table shown above comprises a compilation of the designated nomenclatures for the gathered data. The term "facility" might also refer to a location where data collection is taking place. During the second part of the inquiry, the model serves as a representation of the generating aspect of the SCADA system. The table shown above comprises the nomenclature of the models alongside concise descriptions of each model.





**Risk categories:** The table contains the classification of the risk categories. The table comprises four fields, namely the category name, the description of the risk category, the level of risk, and the sublevel of the risk category.

**Figure 69**

*A depiction of the established risk categories for the experiment.*

	ID	Name	Description	Level	SubLevel
▶	1	T	Threat	3	1
	2	V	Vulnerability	1	1
	4	S	Abilities to successful...	2	1
	6	R	Replaceability	3	2

**Model Component Risk Relationship:** The presented table illustrates the association between the Model Component and Risk Category tables. The dataset comprises four columns. The first column serves as the identification for the record. The ID of the ModelComponent table is located in the second column. The third column of the table provides the identifier that references the RiskID database, while the fourth column represents the risk score associated with the component.

**Figure 70**

*A depiction of the correlation between component and risk relationship*

	ID	ModelComponentID	RiskID	RiskScore
▶	1	38	1	2
	2	39	1	2
	3	40	1	4
	4	41	1	1
	5	83	1	3
	6	84	1	2
	7	35	1	4
	8	42	1	4
	9	43	1	1
	10	44	1	0
	11	45	1	2
	12	46	1	1

**Display:** Individuals have the ability to modify the model according to their preferences and are provided with the choice to store their visual representation within the system for future use. The table presented above comprises the presentation that has been stored by the user. The table is comprised of four columns, namely: the identifier of the record, the modeID which references the model table, the user ID of the author of the display, and a short name assigned to the display.

**Figure 71**

*A depiction of the view that is contained in the database.*

	ID	ModelID	UserID	Name
▶	36	4	m	Save2
	35	3	m	Save1
	18	5	m	Save3
	15	6	m	Saved4

**User Setting:** The table shown herein encompasses the configurations for the aforementioned display. The table is comprised of eight columns, which encompass the table's identification, the displayID that references the display table, the componentID that references the component table, the coordinates of the table (containing the X coordinate, Y coordinate, width, and height), and the table's display state.

**Figure 72**

*An illustration of the position of the component.*

	ID	DisplayID	Componen...	X	Y	Width	Height	Stop
▶	361	36	84	734	253	368	129	False
	362	36	42	150	400	356	129	False
	364	36	44	449	34	355	129	False
	365	36	45	911	39	355	129	False
	366	36	46	23	31	355	129	False
	367	36	47	1002	563	381	129	False
	368	36	48	20	569	381	129	False
	369	36	49	517	572	381	129	False
	356	35	38	943	411	381	129	False
	357	35	39	39	403	381	129	False
	358	35	40	491	409	381	129	False
	359	35	41	478	222	459	129	True

The study's focus is limited to conventional SCADA designs, while there is potential for its extension to incorporate more complex SCADA systems.

**Figure 73**

*The database structure of the ABCD risk management framework.*

