FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Cybersecurity analysis of a SCADA system under current standards, client requisites, and penetration testing

Filipe Rocha

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Internal Supervisor: Professor Doctor Ricardo Santos Morla

External Supervisor: Eng. José Carlos Fonseca

February 18, 2019

Resumo

Sistemas de Supervisão e Aquisição de Dados (SCADA) são sistemas essenciais para monitorizar e controlar Infraestruturas Críticas (IC) de um país, tais como a rede elétrica, distribuição de gás, distribuição de água e serviços de transporte. Estes sistemas costumavam ser isolados e seguros, mas já não o são devido ao uso de redes de comunicação mais amplas e interconectadas que foram adotadas para obter benefícios como escalabilidade, confiabilidade, usabilidade e integração. Devido a esta mudança na arquitetura da rede e à importância crítica que os sistemas têm, os sistemas SCADA tornaram-se alvos desejáveis de ataques cibernéticos. Tal como em sistemas de Tecnologia da Informação (TI), normas e boas práticas foram desenvolvidas para fornecer orientação a desenvolvedores de sistemas SCADA de como melhorar a segurança dos seus sistemas contra ataques cibernéticos.

Com a ajuda da EFACEC, este trabalho fornece uma metodologia para ajudar a realizar uma análise de cibersegurança dum sistema SCADA sob normas atuais, requisitos dum cliente e testes de penetração. Temos como objetivo fornecer orientação, através de exemplos, de como avaliar e melhorar a segurança de sistemas SCADA. Para tal seguimos duas abordagens, uma teórica e uma prática. Na abordagem teórica, começamos por compilar uma lista das normas de segurança mais usadas e referenciadas em Sistemas de Controlo Industrial (SCI). De seguida, é feita uma análise de cada uma destas normas, onde fazemos um resumo geral da norma, e realçamos e explicamos os capítulos e/ou requisitos que são relevantes para sistemas SCADA. Depois, sugerimos e demonstramos uma abordagem de como fazer uma análise dos requisitos de cibersegurança de um cliente genérico, onde dividimos os requisitos em grupos temáticos e associamos a cada requisito recomendações e requisitos das normas. Por fim, propomos soluções para garantir a conformidade de cada requisito. Na abordagem prática, apresentamos uma metodologia para estabelecer um modelo de ameaças para ajudar a identificar pontos de acesso comuns, bens desejáveis e possíveis vetores de ataque que permitiriam aceder a tais bens. Finalmente, propomos uma metodologia de testes de penetração que permitirá validar os vetores de ataque do modelo de ameaças.

Ambas as abordagens foram testadas num protótipo do sistema SCADA da EFACEC conhecido como ScateX#, permitindo-nos identificar com sucesso os requisitos que não estavam a ser cumpridos. As soluções que foram propostas na abordagem teórica ajudaram a fornecer orientação para assegurar o cumprimento destes requisitos. A informação que nos foi fornecida na abordagem prática permitiu-nos identificar aonde o sistema tinha que ser fortificado primeiro.

Abstract

Supervisory Control and Data Acquisition (SCADA) systems are essential for monitoring and controlling a country's Critical Infrastructures (CI) such as electrical power grids, gas, water supply, and transportation services. These systems used to be mostly isolated and secure, but this is no longer true due to the use of wider and interconnected communication networks to reap benefits such as scalability, reliability, usability, and integration. This architectural change together with the critical importance of these systems made them desirable cyber-attack targets. Just as in other Information Technology (IT) systems, standards and best practices have been developed to provide guidance for SCADA developers to increase the security of their systems against cyber-attacks.

With the assistance of EFACEC, this work provides a methodology to assist in performing a cybersecurity analysis of a SCADA system under current standards, client requisites, and penetration testing. Our aim is to provide guidance by example on how to evaluate and improve the security of SCADA systems, following both a theoretical and practical approach. For the theoretical approach, we start by compiling a list of the security standards most commonly used and referenced in Industrial Control Systems (ICS). We then analyze each of these standards by performing a general overview of the standard, and by highlighting and explaining the chapters and/or requirements in the standard that are relevant to SCADA systems. After, we suggest and demonstrate an approach on how to perform an analysis of a generic client's cybersecurity requisites, by dividing the requisites into thematic groups and associating each requisite with standard recommendations and/or requirements. Lastly we propose solutions to assure the compliance of each requisite. For the practical approach, we present a methodology to establish a threat model to help identify common entry points, desirable assets on SCADA systems and possible attack vectors that could allow access to such assets. Finally, we propose a penetration testing methodology that will help validate the attack vectors of the threat model.

Both of these approaches were tested on a prototype of EFACEC's ScateX# SCADA system, allowing us to successfully identify the requisites that weren't being complied. The solutions proposed in the theoretical approach helped in providing guidance to assure the requisite's compliance. The information provided in the practical approach allowed us to identify where the system had to be fortified first.

Agradecimentos

I am extremely grateful to all that participated in this thesis:

- Firstly, to Professor Ricardo Morla for all support provided (and patience!);
- My supervisor at EFACEC Eng. José Carlos Fonseca which and ex-supervisor Eng. João Luís Pinto, helped create all the conditions necessary to carry ou this work;
- To all the collaborators at EFACEC for the constant accompaniment and help;
- To my dearest family, for the unconditional support and understanding shown.

  It is a great pleasure to thank everyone who helped me work on my dissertation.

Thank you all.

Filipe Rocha

"If you're not doing scans and penetration tests,
then just know that someone else is.
And they don't work for you."

George Grachis

Contents

List of Figures

List of Tables

Abbreviations and Symbols

AP	Access Point
BES	Bulk Electric System
CPNI	Centre for the Protection of National Infrastructure
CVE	Common Vulnerability Enumeration
DHCP	Dynamic Host Configuration Protocol
DMZ	Demilitarized Zones
DNP3	Distributed Network Protocol version 3
DNS	Domain Name System
ENISA	European Union Agency for Network and Information Security
FEP	Front-end Processor
FTP	File Transfer Protocol
HMI	Human Machine Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
ICS	Industrial Control System
IEC	International Electrotechnical Commission
IP	Internet Protocol
IPSec	Internet Protocol Security
ISA	International Society of Automation
ISO	International Organization for Standardization
IT	Information Technology
LAN	Local Area Network
MAC	Medium Access Control
MITM	Man-in-the-middle attack
NAT	Network Address Translation
NERC CIP	North American Electric Reliability Corporation Critical Infrastructure Protection
NIST	National Institute of Standards and Technology
NTP	Network Time Protocol
PLC	Programmable Logic Controllers
RBAC	Role-based access control
RTU	Remote Terminal Unit
SANS	System Administration, Networking and Security
SCADA	Supervisory Control and Data Acquisition
SSH	Secure Shell
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	Universal Datagram Protocol
VLAN	Virtual LAN
VM	Virtual Machine
VPN	Virtual Private Network
WAN	Wide Area Network

Chapter 1

Introduction

Supervisory Control And Data Acquisition (SCADA) systems have played a vital role during the last decades in monitoring and controlling large-scale industrial and critical infrastructures such as electrical power generation/distribution, water plants, transportation services, oil & gas refineries, and others. These systems have been in use for more than 40 years, and are expected to remain operational for decades after their deployment providing high availability [13]. When initially created, SCADA systems used to be strictly isolated systems that used proprietary protocols and software to operate. With the advance of computer technology, these systems have also become more advanced and complex, adopting the use of interconnected networks to reap benefits such as scalability, reliability, usability, and integration [14].

1.1 Context and Motivation

Although these systems benefited immensely with their architectural change that adopted internet-based techniques, it also consequently exposed them to threats and vulnerabilities they have never been exposed to before, and to a much greater extent than previously [15]. Additionally, when SCADA systems were first designed, they were not designed with cybersecurity in mind. They were designed to do what they are intended to do, monitor and control. Due to this, many of the most commonly used protocols in current SCADA systems still lack security features, making them vulnerable to cyber-threats. These two issues, combined with the critical importance of SCADA systems, made these systems highly desirable targets for cyber-attacks. Attacks on these systems may threaten the system's operation, safety and stability, causing large economic losses. In recent years, many malicious cyber-security incidents that specifically targeted SCADA systems were reported. Two of the most notorious cyber-attacks on SCADA systems that were extremely sophisticated and complex were the Stuxnet worm and Ukrainian power grid hack.

Just as in Information Technology (IT) systems, standards, guidelines and best practices were developed by recognized standard setting bodies that aim to improve the security of Industrial Control Systems (ICSs) such as SCADA. They do this by providing recommendations and/or requirements for SCADA developers to enhance the security of their systems against cyber-threats.

Due to the positive impact of these standards, SCADA system clients started requesting the systems compliance with these standards.

After learning about the security issues within SCADA systems, attempts were made by their developers and clients to address these issues. They then discovered that conventional security solutions are not always applicable to SCADA systems. This is due to performance and availability requirements being different for administrative IT systems and SCADA systems [15]. This concern also applies to penetration testing security solutions.

1.2 Objectives

Motivated by these facts, main goal of this work is to perform a cybersecurity analysis of a SCADA system under current standards, client requisites, and penetration testing. To do this, our initial objectives are to:

- Undertake a survey of the most common security standards that are used and referenced in ICS;
- Perform an overview of each standard to identify recommendations, requirements, and controls that apply to SCADA systems;
- Analyze a generic client's cybersecurity requisites to identify what standards they reference;
- Propose solutions that could assure the compliance of each requisite;
- Develop a methodology to establish a threat model to understand an attacker's source, potential targets and approaches;
- Suggest a penetration testing methodology to validate the threat model.

1.3 Thesis Outline

The structure of this thesis is organized as follows:

- Chapter 2: SCADA System Overview and attacks:

  Provides a general overview of SCADA systems and past incidents. It presents a typical architecture of a SCADA system, describes SCADA-specific communication protocols and explains the most impactful cyber-attacks that were performed on SCADA systems.

- Chapter 3: Related Work:

  Describes the related work researched relative to the analysis of cybersecurity standards, vulnerability assessment and penetration testing.

- Chapter 4: Theoretical Approach to Cybersecurity Analysis:

  Proposes a theoretical approach to cybersecurity analysis. This approach involves performing an overview of standards that are referenced in ICS, highlighting the chapters and/or requirements of each standard relevant to SCADA systems. The approach also suggests a method of performing an analysis of a generic client's cybersecurity requisites.

- Chapter 5: Practical Approach to Cybersecurity Analysis:

  Proposes a practical approach to cybersecurity analysis. In this approach, a methodology to establish a threat model is presented to identify common entry points, desirable assets and possible attack vectors that could allow access to such assets within SCADA systems. A penetration testing methodology is also presented to validate the attack vectors of the threat model.

- Chapter 6: Conclusions and Future Work:

  Provides final conclusions and suggestions of future work.

Chapter 2

SCADA System Overview and Attacks

Before explaining the function of SCADA systems, it's important to reference that Industrial Control Systems (ICS) and Supervisory Control and Data Acquisition (SCADA) systems are not the same thing. These two types of systems are commonly confused as equal, since documents of one almost always reference the other. SCADA systems are instead a subgroup of ICS. Although ICS have many subgroups, they are generally divided into two major subgroups [16]:

- Geographically independent systems, or SCADA systems.
- Geographically dependent systems, such as Distributed Control Systems (DCS).

Over the last decades, SCADA systems have become the most important technology for monitoring and controlling large-scale industrial and critical infrastructures [13]. These include industries such as electrical power grids, oil & gas refineries, water & waste control, and transportation services [17]. A SCADA system is essentially a computerized control system that allows supervisors and operators to be centrally located at a monitoring facility, while this facility monitors, controls and maintains all the components of a large scale system that is distributed throughout a large geographical area. This is achieved by using field controllers (not to be confused with field devices), such as Remote Terminal Units (RTU) and Programmable Logic Controllers (PLC). These controllers are remotely controlled devices that are used to monitor and manage field devices, such as sensors and actuators, and handle alarms [18].

In this chapter we will present a typical architecture of a SCADA system, including what devices are used and how they relate with one another. We will also explain which communication protocols are used, and how they work, when data is transferred from control devices to the SCADA central server. Lastly, we will describe the most impactful cyber attacks that were performed on SCADA systems.

2.1 Architecture

SCADA systems have existed for decades, way before the wide-spread use of Internet. They have been deployed since the 1960s and have since then evolved as technology changes. Over the years, SCADA architectures have undergone four main phases of development [17][19]:

- **Central architecture** - First generation SCADA architecture where mainframe systems with redundancy are in charge of all the functions. The communication between the SCADA mainframe and RTUs is done via Wide Area Networks (WANs), using vendor-proprietary equipment and protocols.
- **Distributed architecture** - In the second generation SCADA architecture, systems took on a distributed architecture where multiple computers in a Local Area Network (LAN) shared the computing load together. Different computers performed specific functions and roles. Communication protocols are similar to the first generation SCADA systems.
- **Network architecture** - The third generation SCADA architecture is similar to the distributed architecture. The major difference in this generation is that the SCADA functionalities are distributed across WAN and not just LAN. Relative to the first and second architectures, the major improvement of this generation was the adoption of WAN protocols, such as the Internet Protocol (IP), for communication. This architecture is the most utilized today.
- **IoT architecture** - The fourth generation is quite new and still undergoing tests. This architecture uses IoT technology and commercial cloud services, making it easier to maintain and integrate. The advantage of this upgrade are increased data accessibility, cost efficiency, flexibility, optimization, availability and scalability [20].

When discussing the architecture of a SCADA system, there is no *de facto* way that they can be setup. Each architecture will vary, depending on three major factors: the SCADA's industry, the SCADA's developer and the SCADA system's client. It all comes down to what a client wants/requests in his system, what the developer is able to deliver and what industry the system is being developed for.

By researching typical SCADA architectures that are used in various studies/papers [13] [21] [22] [23] [14] [24], suggested by standard-setting organizations [25] [26] [27], used in open-source SCADAs [28] [29] [30], and used by SCADA developers [31] [32] [33], we attempt to create a generic SCADA architecture by joining common features of each architecture. The resulting architecture can be seen in Figure 2.1.

One of the most common industries/markets where SCADA systems are used is in power distribution, due to it being an essential requirement of an entire county. This is reflected on the architecture that was designed. Many of the studies and papers that are published that refer to SCADA systems, refer specifically to its use in the energy sector. Additionally, it's one of the main markets that SCADA developers explore, since it's one of the most sought after. For the sake of this work, the primary focus will be on SCADA's usage in power distribution.

As mentioned before, there is no strict way to project a SCADA architecture. It can have multiple devices and multiple networks. Although not recommended, a device can be located on any of the networks. For our architecture, we follow the device placement recommended by the National Institute of Standards and Technology (NIST), the International Society of Automation (ISA) and System Administration, Networking and Security (SANS), in order to be more practical. The SANS document mentioned uses many other standards developed by standard setting

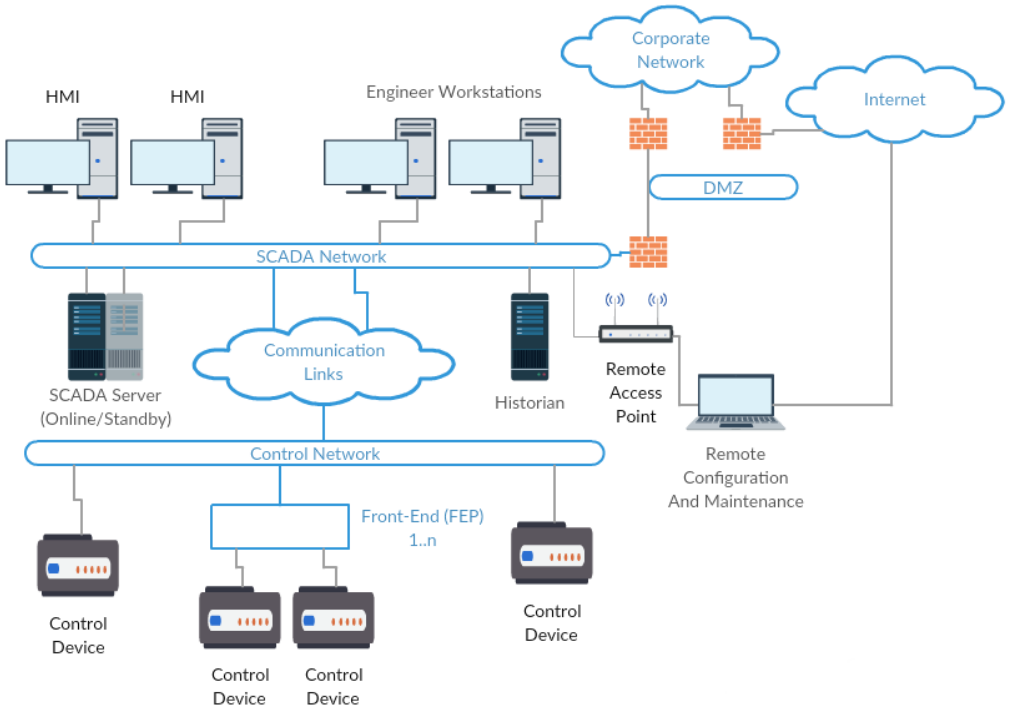


Figure 2.1: Generic SCADA Architecture

organizations as a basis for its information, such as the NERC CIP standards, ISA-99, NIST SP 800-82 and others [27].

According to the sources mentioned, a SCADA system will generally consist of five (or six for SANS) levels. All of these levels are associated with a specific network in the SCADA system that represent specific functions. Every device in the SCADA system can and should therefore be placed in a specific level/network depending on its function. These levels range from 0 to 4 (or 5) and they represent the following:

- **Process Control Network (Level 0)** - Network that contains all of the field devices such as sensors, motors, actuators, and handle alarms;
- **Control Devices (Level 1)** - Network that contains control devices such as RTUs, PLCs, Intelligent Electronic Devices (IEDs), and others;
- **Supervisory Control (Level 2)** - Network that contains the main servers for operating the control system such as SCADA servers, Engineering Workstations, HMIs and Historians;
- **Operations Support (Level 3)** - Network that contains devices related to managing the operations environment such as Test Systems, Operations Analysis Systems, and others;
- **Plant Network (Level 4)** (Only present in SANS) - Network with IT shared services such as Print Servers, Local File Servers, and others;
- **Enterprise Business Network (Level 4/5)** - Corporate network that contains Email Servers, Internal Web Servers, and others.

For the sake of this work, we will be focusing on the two networks/levels that are always represented in a SCADA system's architecture, Level 1 and Level 2. The networks mentioned are the Control network and the SCADA network respectively. As mentioned above, there can be more networks in the system, such as demilitarized zones (DMZ), or enterprise networks. The SCADA network is the centralized monitoring facility network that is composed of machines that monitor, process data and send control instructions, such as:

- **SCADA server**;
- **HMIs**;
- **Engineer Workstations**;
- **Historian**;
- **Distributed Management System (DMS)** (Optional) (Mostly used in electric distribution);
- **Inter-Control Center Communications Protocol (ICCP)** (Gateways to remote control centers) (Replaces Remote access point).

Most of the servers on the SCADA network are critical servers that need to be maintained available at all times. Due to this, most SCADA systems have "duplicated" servers that represent a redundancy feature that most SCADA systems have. This feature consists of having standby servers that can be maintained in one of three standby states: hot (synchronized), warm (periodically synchronized) or cold (not synchronized).

The control network consists of all the control devices such as RTUs, PLCs and Intelligent Electronic Devices (IEDs), that receive the information gathered by field devices. Additionally, the network can contain Front-end processors (FEPs) that are gateway machines commonly used in industrial settings [21]. These machines are capable of establishing communication with multiple control devices, allowing the network to communicate with a larger amount of control devices.

2.2 SCADA-specific Communication Protocols

In summary, SCADA systems are responsible for monitoring and controlling critical infrastructures by gathering information from field controllers, transferring it to the control server, and displaying the information on a HMI. These transfers of data are done over communication channels that either use industry standard protocols, or proprietary protocols.

SCADA-specific communication protocols refer to standard protocols used in data transfer between field controllers and the central server. Due to the nature of this data transfer, most protocols operate on a master/slave basis. The most commonly used SCADA communication protocols are IEC 60870-5-101/104, DNP3, and Modbus. Both the IEC and DNP3 protocols are more functional than Modbus and are generally used for higher data volumes. The IEC protocol is mostly used in European countries whereas DNP3 is widely used in North America [34].

2.2.1 IEC 60870-5-104

IEC 60870 is a group of standards created by the International Electrotechnical Commission (IEC) for telecontrol (SCADA) in electrical engineering and power system automation applications [1]. The group consists of six main parts plus a number of companion standards. Part 5 (IEC 60870-5) in particular, known as Transmission protocols, provides a communication profile for the transmission of SCADA telemetry control and information [35].

When IEC 60870, or IEC 870 for short, is discussed in the context of SCADA, it normally refers to the companion standard IEC 60870-5-101, released in 1995, due to it being the first document that detailed the complete SCADA transmission protocol, allowing it to be used in production [35]. The IEC 60870-5-104 standard was subsequently released, in 2000, as an extension of IEC 60870-5-101, since 101 was originally designed for serial communications. IEC 60870-5-104 allowed the same serial frames of 101 to be transmitted over TCP/IP [2].

Figure 2.2 shows the data, or payload, structure of an IEC 60870-5-104 packet. This payload is often referred to as the Application Protocol Data Unit (APDU), and it consists of two parts, the Application Protocol Control Information (APCI) and Application Service Data Unit (ASDU) [2].

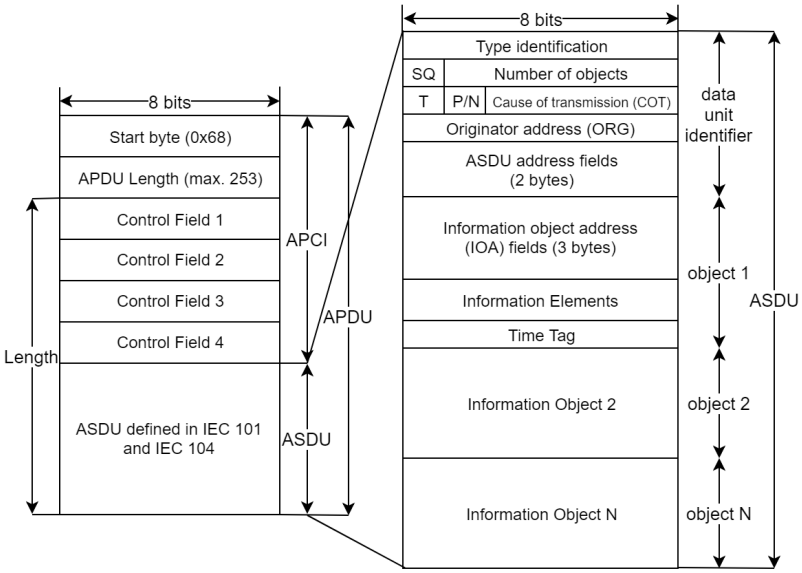
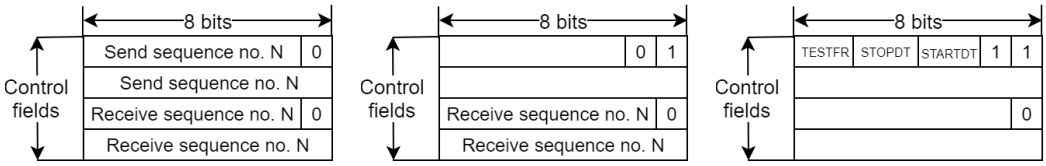


Figure 2.2: IEC 60870-5-104 APDU Frame Format [1] [2]

The APCI is essentially used as a communication start and stop mechanism for the ASDU. It generally has a length of 6 bytes, that includes a start byte with value 0x68 followed by a 8-bit length field (length of the APDU) and four 8-bit control fields. The APDU's frame format/type is determined by the last two bits of the APCI's first control field as seen on Figure 2.3, and can be defined as [1]:

- I-format (X0) - **Information** transfer format;
- S-format (01) - Numbered **supervisory** functions;
- U-format (11) - **Unnumbered** control functions.



(a) I-format (b) S-format (c) U-format

Figure 2.3: APCI Control Field Information [1]

The ASDU (Figure 2.2), that is only incorporated in the I-format, contains two main sections, the data unit identifier and the data payload of one or more information objects [3]. The data unit identifier in particular defines the specific type and amount of data, provides addressing to determine the identity of data, and includes additional information such as cause of transmission (COT) [1]. One of the fields present in the identifier is the Type identification field. This field is responsible for defining the type of data by referring to 8-bit code types. The types that are currently defined by IEC are shown on Figure 2.4 [3]. Another field is the Number of objects that indicates the amount of information objects contained in the payload, and can vary from 0 to 127. The field labeled "Cause of transmission" indicates the reason why the payload was transmitted, and is used to control the routing of messages. Its values vary from 1-47 for standard definitions and 48-63 for special use. Lastly, the ASDU address field, or common address for ASDU, is associated with all objects contained within the ASDU. It is normally interpreted as a station address [1]. Additional information of IEC 104 ASDU types and COT values can be found in *Matoušek's* analysis of IEC 104 protocol [1].

CODE TYPE RANGE	**GROUP**
1-21, 30-40	Process information in monitor direction
45-51	Process information in control direction
70	System information in monitor direction
100-106	System information in control direction
110-113	Parameter in control direction
120-126	File Transfer

Figure 2.4: Code Type Groups [3]

After the data unit identifier, each information object will start with an Information object address (IOA) followed by the actual information. This address is used as a destination address in control and as a source address in monitor direction [1].

IEC 60870-5-104 is generally assigned, by default, to the TCP port number 2404 [1] [3].

2.2.2 DNP3

The Distributed Network Protocol Version 3 (DNP3) is a communication protocol standard that defines communications between master stations, RTUs and other intelligent electronic devices. The protocol was designed specifically for SCADA applications, and was originally created as a proprietary protocol by Harris Controls Division to be used solely in the electrical utility industry. It was then later made available for public use as an open protocol standard, when its ownership was transferred to the DNP3 User Group, making it an accepted standard the electric, oil & gas, waste/water, and security industries [35].

DNP3 is primarily used within SCADA so control centers can communicate with remote substations, or outstations in the case of serial communication. It's typically configured in a master-slave configuration, where the DNP3 master would be the control center, and the slaves would be the various RTUs inside a substation [11]. An example of a configuration with one master and multiple slaves (multi-drop) can be seen on Figure 2.5.

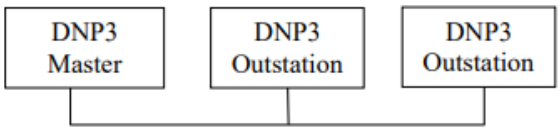


Figure 2.5: DNP3 Master/Slave Architecture [3]

In terms of architecture, DNP3 is a four-layer subset of the Open Systems Interconnection (OSI) model. These layers include the application, data link, physical, and pseudo-transport layers [3]. Figure 2.6 shows the DNP3 packet architecture. The first section of the packet is the Data Link layer frame. This frame starts off with the two start bytes (0x0564) that indicate where the frame begins. The following byte is the Length field that specifies the number of bytes of the remainder of the frame excluding the Cyclic Redundancy Check (CRC). Next is the Control field (1 byte) that contains information about the packets contents. Both the Destination and Source fields that follow are 2 byte addresses that identify the DNP3 device receiver and sender respectively [11]. By using this 2 byte addressing scheme, there are over 65500 available addresses in which every DNP3 device is required to have a unique address for sending and receiving messages to and from each other. Various CRC fields appear along the packet due to the data payload being divided into blocks. Each block contains a pair of CRC bytes for every 16 data bytes with the exception of the last block. Following the Data Link layer is the pseudo-transport layer. This layer is responsible for breaking long application layer messages into smaller packets that are suitably sized for the link layer to transmit, and, when receiving, to reassemble the frames into longer application layer sized messages [3].

Finally, we have the Application layer. This layer contains the instructions for the devices, such as confirmation, reads, writes, selects, restarts, responses, and more. The layer starts off with an Application Header that contains a control byte, followed by a function code that provides the instruction, and ends with internal indications only if the instruction is a response [11]. Table 2.1

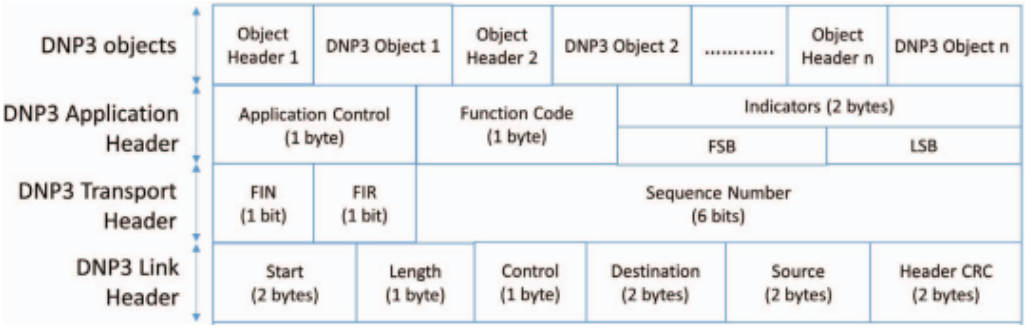


Figure 2.6: DNP3 Message Architecture [4]

lists the most commonly used function codes when performing a vulnerability assessment and the most enticing ones for attacking the DNP3 protocol. A list of all possible function codes can be fount in Appendix A.

Table 2.1: DNP3 Function Codes [11]

Function Code	**Function Code Description**
0x00	Confirm
0x01	Read
0x02	Write
0x03	Select
0x04	Operate
0x05	Direct Operate
0x0d	Cold Restart
0x0e	Warm Restart
0x12	Stop Application
0x1b	Delete File
0x81	Response
0x82	Unsolicited Response

Like other traditional SCADA protocols, DNP3 was originally designed for serial communications. As in IEC 104, the protocol was extended to allow the use of TCP/IP as a transport mechanism. This extension was done by simply encapsulating the entire DNP3 frame with TCP/IP headers, maintaining the original architecture [4]. DNP3 is assigned, by default, to the TCP port number 20000 [3] [11].

2.2.3 Modbus

Modbus is a serial-based protocol that was developed in 1979 by Modicon (now Schneider Electric) to be used in industrial automation systems and with their PLCs [6]. It's one of the most commonly used protocols in ICS systems due to it being a simple and robust protocol that is open to use without requiring royalties. As in other SCADA communication protocols, Modbus has since been altered to work on Ethernet networks. This was achieved by encapsulating the

serial-based protocol inside of TCP headers. There are many iterations of Modbus, which include Modbus RTU, Modbus+, Modbus TCP/IP, Modbus over TCP/IP, that is similar to Modbus TCP/IP, but it has checksums within the payload of the packet, and other less common implementations [11]. In this thesis we will focus on Modbus TCP/IP that is the encapsulated version of Modbus RTU.

Devices that communicate with one another via Modbus establish this communication using a master-slave (client-server) technique in which only one device (master/client) can initiate transactions, or queries. The other devices (slave/server) simply respond by supplying the requested data to the master, or by performing the action requested in the query [6]. In the case of SCADA systems, Modbus is used to establish communication between the control center (master/client) and RTUs (slave/server). An example of this client-server model on an Ethernet TCP/IP network can be seen in Figure 2.7. In this model, there are four possible types of messages that represent the following [5]:

- **Modbus Request** - message sent by the client to initiate a transaction.
- **Modbus Indication** - request message received on the server side.
- **Modbus Response** - response message sent by the server.
- **Modbus Confirmation** - response message received on the client side.

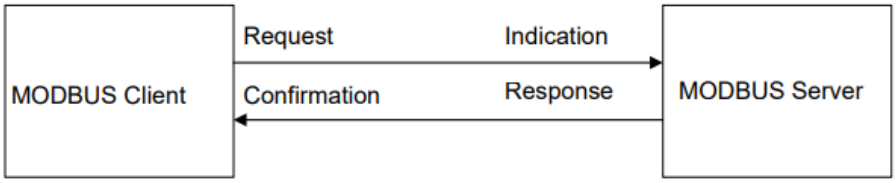


Figure 2.7: Modbus TCP/IP Client/Server model [5]

In terms of architecture, a basic Modbus packet frame is shown in Figure 2.8. This packet frame can be split into two sections: the Application Data Unit (ADU) and the Protocol Data Unit (PDU) which is enclosed by the ADU. The ADU includes an address field, the PDU and an error checking method. The PDU consists of a function code and a data field [11].

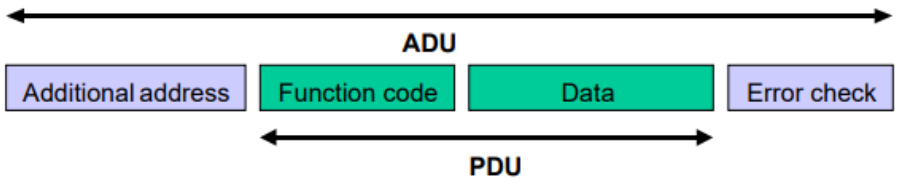


Figure 2.8: General Modbus frame [5]

The Modbus packet frame that is used in Modbus TCP/IP differs slightly from the basic frame. As shown in Figure 2.9, it's also composed of an ADU and PDU, however, the ADU in this packet frame consists of a Modbus Application (MBAP) header and the PDU, dropping the error checking

method. The MBAP header is a 7-byte header that contains a Transaction ID, Protocol ID, Length, and a Unit ID. The Transaction ID field is 2 bytes set by the master to identify each transaction. The Protocol ID field is the 2 bytes that are used to identify the protocol, and for Modbus are always set to 0x0000. The Length field also consists of 2 bytes and indicates the number of following bytes until the end of the ADU, including the Unit ID and data fields. The Unit ID consists of 1 byte and is used to identify a remote slave located on a non TCP/IP network (for bridging Ethernet to a serial sub-network) [5].

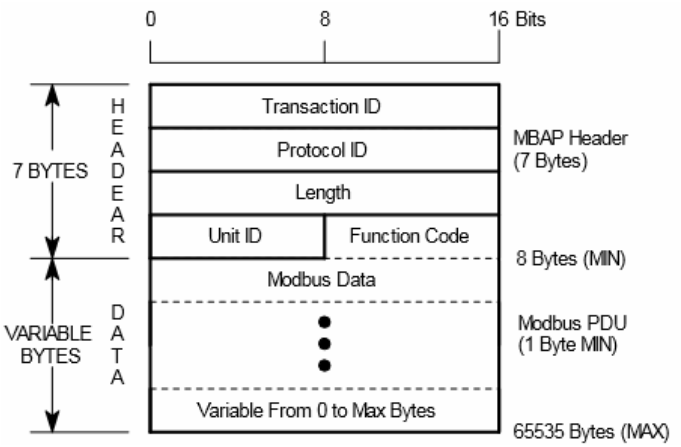


Figure 2.9: Modbus TCP/IP Application Data Unit (ADU) [6]

The PDU consists of a function code and the actual data of the protocol. The function code field consists of 1 byte that tells the slave what kind of action to take. Function codes can be categorized in three ways: public, user-defined and reserved. Valid function codes range from 1-255 in decimal, but not all codes will apply to a module. Of these 255 function codes, some are reserved for future use, others, such as 65-72 and 100-110, are allocated for user-defined services [6]. Some public function codes that most devices will support are shown on Table 2.2.

Table 2.2: Modbus Public Function Codes [11]

Function Code - Function Description (Decimal)	
1 Read Coils	14 Read Device Identification
2 Read Discrete Inputs	15 Write Multiple Coils
3 Read Multiple Holding Registers	16 Write Multiple Holding Registers
4 Read Input Register	17 Report Slave ID
5 Write Single Coil	20 Read File Record
6 Write Single Holding Register	21 Write File Record
7 Read Exception Status	22 Mask Write Register
8 Diagnostic	23 Read/Write Multiple Registers
11 Get Com Event Counter	24 Read FIFO Queue
12 Get Com Event Log	43 Read Device Identification

Like the other protocols mentioned before, Modbus is assigned a default TCP/IP port when

it is configured. Modbus TCP/IP packets are transferred across Ethernet networks over TCP port number 502 [5] [6] [11].

2.3 Cyber-attacks on SCADA Systems

As mentioned before, SCADA systems have existed for many decades, existing even before the widespread use of Internet. Their main focus in terms of IT-security used to be mostly consisted of protecting the physical access to the computers of the system. Since their adaptation to interconnected communication networks, SCADA systems have become more susceptible to a wider variety of cyber-attacks.

Although cybersecurity should be one of the highest priority tasks in todays SCADA systems, there is still much work to be done in this area. There are many examples of past incidents, where unauthorized users attempt to gain access to SCADA systems by have exploiting vulnerabilities in the systems. Once access is gained, unauthorized users can control these systems, potentially leading to catastrophes.

In this section we discuss some of the most well-known cyber-attacks on SCADA systems, explaining how they were conducted and what consequences they brought.

2.3.1 Stuxnet Worm

Before discussing this attack, it's important to reference the difference between a computer virus and a computer worm. A computer virus is a type of malicious software that generally requires a host program or human help in order to propagate itself and infect other targets. Computer worms, on the other hand, are a stand-alone type of malicious software that can run independently and are capable of spreading itself over a computer network without the need of assistance. They spread by either exploiting a vulnerability on the targeted system or use some kind of social engineering to trick users into executing them [36].

Stuxnet (name is derived from keywords in its code) is a malicious computer worm that was first uncovered in 2010. It was used to specifically target highly specialized industrial control systems in critical high-security infrastructures [37]. The worm was one of, if not the first malwares every designed to attack control systems and was the first attack of its kind that make SCADA vulnerabilities an important topic [14].

The Stuxnet worm was a unique malicious code in the sense that it performs a sophisticated multiphase/multilayered attack. The worm was designed to specifically spread to three different targeted systems [7] [37]:

1. Microsoft Windows machines.
2. Siemens PCS 7, WinCC and STEP7 industrial software applications that also run on Windows.
3. Siemens S7 PLCs that were attached both to the previous machines and to specific variable-frequency drives.

In Figure 2.10, Kushner [7] details the various stages of the Stuxnet worm's attack, showing how the worm quietly spread and caused damage to the targeted PLCs.

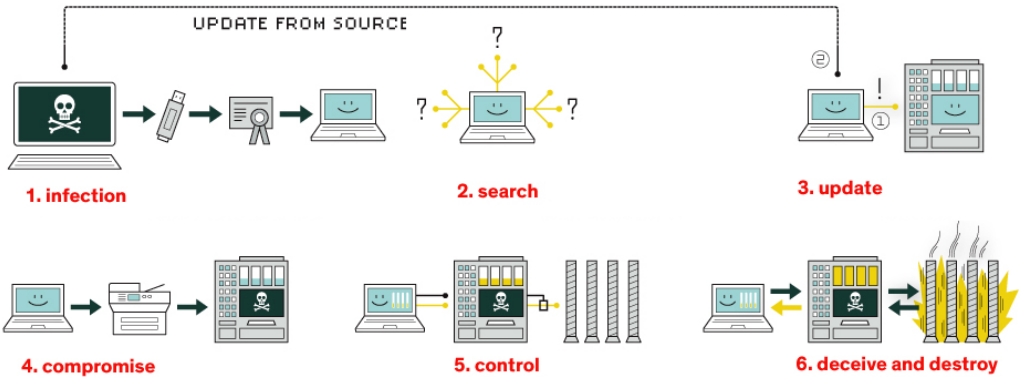


Figure 2.10: Stuxnet Worm Attack Phases [7]

Phase one consists of the initial infection. The Stuxnet worm was originally put on a (or various) USB flash drive(s) by its attackers. The worm entered the targeted systems via a USB flash drive that was plugged into a computer by someone with malicious intent [37]. Once one machine is infected with the worm, it will than proceed to infect all the machines on the network that are running Microsoft Windows. In the second phase of the attack, the worm will identify its second target, by verifying which Windows machines have Siemens industrial software installed. If the system isn't a target, the worm will lay dormant and only infect other machines if necessary. When the worm successfully identifies its second target, it will proceed to the third phase, attempting to access the Internet to download a more recent version of itself. Either updated or not, the worm will continue to the fourth phase of the attack by compromising the Siemens PLCs (third target) that were attached to the previous target and to specific variable-frequency drives. This is done by exploiting "zero day" vulnerabilities that are essentially software weaknesses that haven't been identified by security experts. Phase five and six are generally done simultaneously as it's essentially a Man-in-the-Middle (MITM) attack. The worm begins by gathering information about the operations of the targeted system. It will then use the information gathered to take control of the centrifuges connected to the variable-frequency drives (slave) attached to the PLC (master) by modify the PLC's requests, making them spin themselves to failure. The worm will simultaneously provide false feedback to the monitoring systems, deceiving the operators into thinking that everything is running as normal [7]. The worm also had a un-install mechanism, allowing it to remove itself. Although it was discovered beforehand, the worm was programmed to uninstall itself on 24-June-2012 [37].

Additionally, when it was discovered by a Belarusian malware-detection company in 2010, they found that the malware was signed by one of two digital certificates that were stolen from two different issuing companies, making the software appear to come from a reputable company [7]. After various security firms began reverse engineering the code, they found that the worm required specific slave variable-frequency drives to be attached to the Siemens S7 PLCs. The worm would

only launch its final phases on PLC systems that were connected to variable-frequency drives specifically sold by Vacon (Finnish vendor) and Fararo Paya (Iran). To have a more specialized target, the worm monitored the frequency of the attached motors, and only attacked systems that spin between a specific range, showing just how sophisticated to worm was [37].

Although the authors of Stuxnet have never been officially identified, the size and sophistication of the worm indicates that it could have only been created by a nation-state with and advanced cybersecurity center. It was responsible for destroying nearly 1,000 of Iran's 6,000 nuclear centrifuges, causing billions of dollars in damage [7].

Since this attack, multiple malicious worms related to Stuxnet have been detected. In 2011, Hungarian researchers uncovered a worm named Duqu that was designed to steal information about industrial control systems. In 2012, Kaspersky Lab detected a malware called Flame that supposedly destroyed files from oil-company computers in Iran. When analyzing the malicious code, they found traces of a file named Flame (hence the name) that was also present in early iterations of Stuxnet. They later realized that Flame was a precursor of Stuxnet that somehow had gone undetected. The same year Kaspersky Lab found Gauss. Gauss is a worm that also infected computers via USB flash drives. The worm targeted Lebanese bank credentials, stealing files and gathering passwords [7].

2.3.2 Ukraine Power Outage

In 2015, three different energy distribution companies in Ukraine experienced coordinated cyber-attacks that were executed within 30 minutes of each other, resulting in various power outages. The attack lasted several hours, affecting approximately 225,000 customers in different areas that lost power due to these attacks. The companies where forced to move to manual operations in response [38]. This attack was the first confirmed attack to lake down a power grid.

This attack, much like Stuxnet, was extremely sophisticated and precise, making authorities and researchers believe it was formulated by a well-funded, well-trained team. The hackers that performed this attacks were experienced strategists who carefully planned their assault over many months. They started by doing reconnaissance to study the networks and obtain operator credentials, then launched a synchronized assault [39].

The attack reportedly started off with a spear-phishing campaign that targeted IT staff and system administrators that worked for companies responsible for electricity distribution. The campaign sent an email to workers of the companies with a malicious Word document attached. When workers clicked on the attachment, a popup appeared asking them to enable macros for the document. If they complied, a malware called BlackEnergy3 infected their machines and opened a backdoor. This initial attack only allowed attackers access to the corporate network, which has no impact on the grid itself. To impact the grid, they would have to access the SCADA networks that controlled the grid. To do this, the attackers conducted reconnaissance on the network over many months to be able to map the networks and find vulnerable machines. They managed to find and access the Windows Domain Controllers, where they gathered worker credentials that were used to remotely log in to the SCADA network via VPN. Once on the SCADA network, they started

making preparations for their attack. The first thing they did, was to reconfigure the uninterruptible power supply (UPS), that was responsible for providing backup power to two of the control centers. During the reconnaissance phase, they found that each of the three companies were running different distribution management systems. They studied each of these systems and proceeded to write malicious firmware to replace the legitimate firmware on serial-to-Ethernet converters linked to these systems [39].

Once the attack was ready, the attackers launched the attack by accessing the hijacked VPNs and disabled the UPS systems they had already reconfigured. They accessed a worker's computer at the control station and proceeded to turn the substations offline, without the worker being able to stop their actions. They followed this by logging him out of the control panel and altering his password to avoid re-connection. Before disconnecting the substations, they launched a telephone denial-of-service (TDoS) attack, to prevent customers from calling in to report the outage. Since they had overwritten the firmware of the converters with malware, operators were prevented from sending remote commands. The malware they developed was called KillDisk. It erased all the data from the operator computers and deleted the master boot record, preventing the computers from rebooting [39].

2.4 Conclusions

Supervisory Control And Data Acquisition (SCADA) systems are essential to monitor and control modern industrial and critical infrastructures (CI). These systems have been accompanying the advancement of computer technology, adopting the use of internet-based networks to improve their scalability, reliability, usability, and integration. This adoption also exposed these systems to threats and vulnerabilities they have never been exposed to before, and to a much greater extent.

SCADA systems have been in use and are expected to remain operational for decades. When first developed, all of these systems used proprietary communication protocols that were serial-based in data transfer. Since then, some of these proprietary protocols have become standard communication protocols that are used by the majority of current SCADA systems. Since they were originally developed for serial-based communications, most of the standard protocols had to adapt to remain functional on the newer internet-bases networks. Currently, most of the standard communication protocols used in SCADA systems operate on TCP/IP. They managed to do this by essentially encapsulating the original serial-based frames with TCP headers. But this "quick fix" left these protocols vulnerable to a variety of attacks such as packet sniffing, spoofing and MITM due to them lacking key security features like confidentiality, integrity and authentication.

All of the reasons mentioned above made these systems desirable targets for cyber-attacks. They are essentially critical systems that are operating on an internet-based network in which little to no security features were adopted. Attacks on these systems are capable of threatening the system's operation, safety and stability, consequently causing large economic losses. In recent years, the number of malicious cyber-security incidents that specifically targeted SCADA systems has grown substantially. Two of the most notorious cyber-attacks on SCADA systems were the

Stuxnet worm and Ukrainian power grid hack due to the sophistication, complexity and multiple phases of the attacks.

Chapter 3

Related Work

This chapter discusses previous work that is related to ours. Namely cybersecurity standard analysis, and vulnerability assessment and penetration tests on SCADA systems. To the extent of our knowledge, there is no work related to client cybersecurity requisite analysis, as this is concept is quite new.

3.1 Survey and Analysis of Cybersecurity Standards

Sommestad et al. [15] have analyzed the focus in SCADA security by using a comprehensive search of a large number of standards produced by governmental agencies and standardization bodies. To determine if a standard would be included in their study, they applied the following criteria: The standard is available in English; The standard is published by a standardization body or governmental agency; The standard must focus on SCADA system security; The standard/guideline must focus on SCADA systems as a whole. This criteria narrowed their findings down to the following security standards:

- **Good Practice Guide, Process Control and SCADA Security** by Centre for the Protection of National Infrastructure (CPNI);
- **Cyber Security Procurement Language for Control Systems** by Department of Homeland Security (DHS);
- **21 steps to Improve Cyber Security of SCADA Networks** by U.S. Department of Energy (DOE);
- **CIP-002-1 - CIP-009-1** by North American Electric Reliability Corporation (NERC);
- **Guide to Industrial Control Systems (ICS) Security** by National Institute of Standards and Technology (NIST);
- **System Protection Profile - Industrial Control Systems** by NIST;
- **ANSI/ISA–99.00.01–2007 Part 1-3** by The International Society of Automation (ISA);
- **Cyber security for Critical Infrastructure Protection**.

Using these standards they proceed to quantify the focus of the the standards by creating 26 groups of security recommendations and 14 groups of threats and associating keywords and key

phrases to these groups. As an example, the Firewall group was associated with firewall, packet filtering, stateful inspection, application proxy, boundary protection. They then count the number of occurrences of each group and normalize the value for statistical purposes. They do the same procedure for the generally used IT standard ISO/IEC 17799 so they can compare the focus of SCADA standards relative to IT. The results they obtain after all the values were normalized is in Figure 3.1

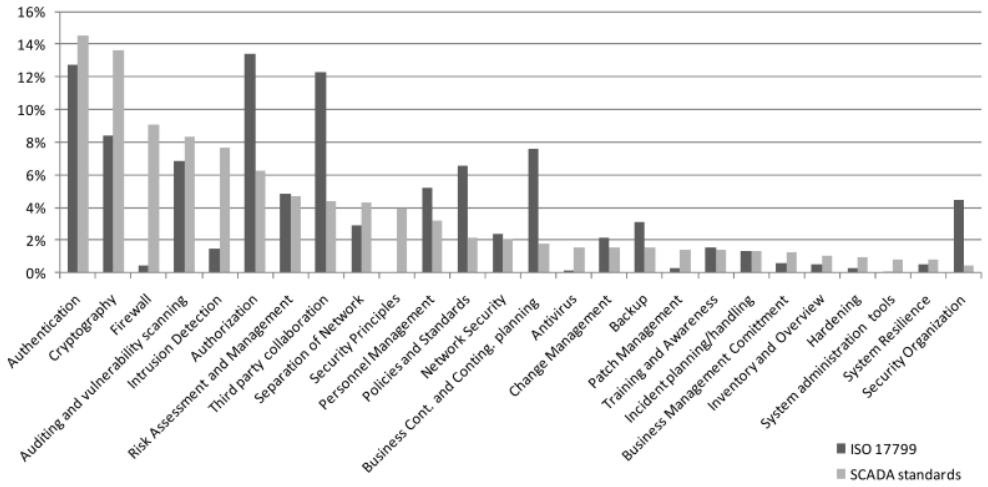


Figure 3.1: Focus on countermeasures for ISO/IEC 17799 and SCADA standard, normalized

As can be seen, the main focus in SCADA systems security in terms of recommendations is authentication followed by cryptography. IT systems on the other hand focus primarily on authorization. These is however common ground between the two in terms of authentication.

They also performed the same analysis for the 14 groups of threats, concluding that SCADA security standards focus mainly on malicious code, followed by Denial of Service attacks (DoS) and Distributed DoS (DDoS), and finally a combination of spoofing techniques like MITM, and lastly replay attacks.

Disterer [8] presents the ISO/IEC 27000, 27001 and 27002 standards, their development and actual dissemination, as well as the ISO 27k family of standards. He performs a general overview of each document and specifies the objective of every control the documents recommend. He also concludes that each of the documents can serve as a framework to design and operate an information security management system.

3.2 Vulnerability Assessment and Penetration Testing

Maynard et al. [2] developed a MITM attack for ettercap that is capable of altering packets of the SCADA-specific communication protocol IEC 60870-5-104 SCADA. They start by providing details of the protocol's packet payload structure. They then explain how their MITM attack follows the three stages of the anatomy of and attack: detection, that is the process of identifying

the target (specific IEC 104 packets ate chosen); capture, that involves collecting the packets data information; and nally attack, where the packet's data is actually altered. The experiments conducted by them cover both relay and MITM attacks. Lastly they explain how attackers with varying degrees of experience are able compromise a SCADA system by hiding fault conditions from a SCADA server.

Waagsnes et al. [40] present an Intrusion Detection System (IDS) test framework for SCADA networks in the electrical energy sector. Although IDS is out the scope of our work, they proceed to test their IDS by preforming a variety of attacks on a simulated SCADA network. They setup an attacker machine that is running Kali Linux and perform reconnaissance, man-in-the-middle (MITM) attack, denial of service (DoS) attacks, brute force attacks, vulnerability exploitation and other attacks on SCADA target. In their experiment, they attack two SCADA-specific communication protocols, namely IEC 104 and Modbus. IEC 104 packets are altered using an ettercap plugin developed by Peter Maynard and use metasploit to send unauthorized read and write requests to a PLC that communicates using Modbus.

Nazir et al. [14] servery a number of tools and techniques to uncover SCADA system vulnerabilities. They mention known vulnerabilities in SCADA systems such as: SCADA systems having Generic OS; Legacy systems with long operational life (old technology); Multiple points of entry and failure (geographically spread); Communications protocols; Real-time and complex interactions; Conicting priorities; Social engineering and insider attacks (malicious employees); and Backdoors. They also present a vast survey of various attacks that can be simulated such as: Malware attacks; Network attacks; Communication protocol attacks; Denial-of-service/MITM; False data injection; False sequential logic attacks; Integrity attacks; and Real-time and simulation monitor. They also recommend a large variety of tools and techniques that perform different tasks, being NMAP (scanning tool), metasploit (penetration testing) the ones relevant for this project.

Javate et al [18] created a penetration testing operating system called Moxi Linux. It is an ICS-centric version of the famous Kali Linux tailored with defensive and adversarial tools for security researchers in the ICS domain. The primary tools the systems contains that are ICS specific are: Quickdraw SCADA Snort Rules from Digital Bond; CoDeSys exploit from Digital Bond; PLC Scan from Dmitry Efanov; Modscan from Mark Bristow; Siemens S7 metasploit modules from Dillon Beresford; Siemens S7 wireshark dissector from Thomas Wiens. These tools are PLC penetration testing tools. They developed this OS by performing a survey of publicly available defensive and adversarial ICS related tools. Of all the penetration testing tools they surveyed they found only two that were specifically designed for ICS. These were INL's Kali Linux and SamuraiSTFU.

Chapter 4

Theoretical Approach to Cybersecurity Analysis

In this chapter we propose a theoretical approach to evaluate and improve the security of SCADA systems. To achieve this, we start by undertaking a survey of the most common security standards that are used and referenced in ICS. We then provide a general overview of each standard, and highlight and explain the chapters and/or requirements of each standard that are relevant to SCADA systems. After we suggest an approach on how to perform an analysis of a generic client's cybersecurity requisites, by firstly dividing the requisites into theme groups to allow faster analysis, and associating each requisite with standard recommendations and/or requirements. In this chapter we also present solutions to assure each requisite's compliance.

Additionally, we tested this approach on the ScateX# prototype and confronted each requisite with the prototype's specifications. We then performed a study of the system's architecture to identify vulnerabilities and non-compliances with both the client's requisites and the standards. We were able to identify some non-compliances, and presented the solutions we had recommended in the approach to attempt to assure requisite compliance. We will not describe or display this process in this paper as it would contain confidential information.

4.1 Cybersecurity Standards

Both the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) define a standard as "a document, established by consensus and approved by a recognized body, that provides, for common and repeated use, rules, guidelines, or characteristics for activities or their results, aimed at the achievement of the optimum degree of order in a given context" [41]. Standards emerge through the development of detailed descriptions of particular characteristics of a product or service by experts from companies and scientific institutions. These characteristics include quality, security and reliability. Once consensus is established that these characteristics can remain applicable for an extended period of time, they are

documented and published in the form of a standard. Standards are developed to provide support to both companies and individuals when acquiring/providing products and services [8].

That said, cybersecurity standards are standards developed and approved by recognized standard-setting bodies such as ISA, ISO, IEC, NIST, NERC, IEEE, and others, that aim to improve the security of information technology (IT) systems, networks, and critical infrastructures. They attempt to do this by providing various recommendations and/or requirements that may be enforced through penalties and fines, depending on the industry of the standard.

The goal of this section is to help readers, SCADA developers and SCADA customers better understand the most common security standards that are used and referenced in ICS, as well as what recommendations and/or requirements they mention. To achieve this, we not only provide a general overview of each standard, but also highlight and explain the chapters and/or requirements of each standard that are relevant to SCADA systems in particular. In this work we will not analyze recommendations and/or requirements for physical security, as this cannot be assured by SCADA developers.

4.1.1 Global Security Standards

Global security standards can also be referred to as general security standards. These standards were developed with one main focus, information security. They are standards that are deliberately broad in scope, covering more than just privacy, confidentiality and IT security issues. Since most systems nowadays operate on networks, information security is always an issue, therefore these standards are applicable to a vast amount of organizations, regardless of their type or size. This includes ICS and SCADA systems. There are quite a few standards that can be globally used, such as the ISO/IEC 27000 series, the NIST Cybersecurity Framework, and others. These two standards in particular, although not specific to ICS, are often referenced in conjunction with ICS cybersecurity standards.

For the sake of this thesis, we will only be dissecting and analyzing the ISO/IEC 27000 series, ISO/IEC 27002 in particular that is titled Code of practice for information security management.

4.1.1.1 ISO/IEC 27002

The ISO/IEC 27000-series (or ISO27k for short) are a series of standards published by ISO, IEC and the American National Standards Institute (ANSI). It provides good practice recommendations on information security management, risks and controls within the context of an overall Information Security Management System (ISMS). Figure 4.1 shows how the ISO 27000 series is organized, as well as their interrelations.

The first part of the series, ISO 27001, is the baseline for the whole of the series. It defines the requirements needed to create an ISMS, based on widely accepted risk management techniques. The second part of the series, ISO 27002 , is the part of the series that is typically referenced in ICS cybersecurity policies and standards. It outlines numerous control objectives that, in effect, address risks that are identified when using the techniques mentioned in ISO 27001.

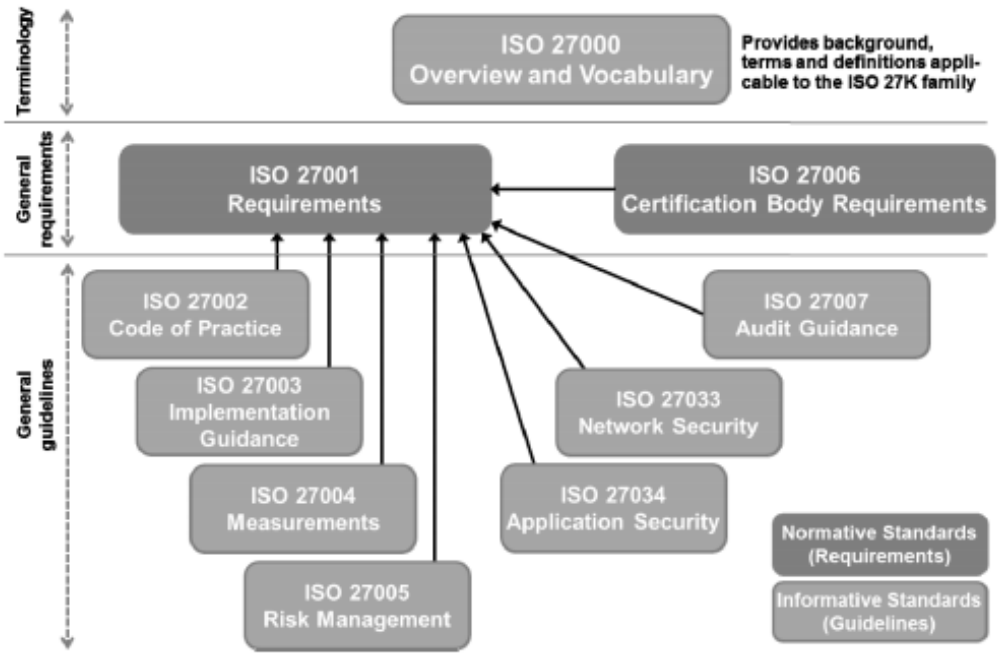


Figure 4.1: Origanization and interrelations within the ISO 27K series [8]

Chapters 0-4 of ISO 27002 [42] aren't relevant to this study as they are composed of the introduction, scope, normative references, terms and definitions and structure. The chapters on this document that are relevant are the following:

5. **Security Policy Management**

   5.1. Provide management direction and support
   Management should define a set of policies to clarify their direction of, and support for, information security.

6. **Corporate Security Management**

   6.1. Establish an internal information security organization
   The organization should lay out the roles and responsibilities for information security, and allocate them to individuals. Information security should be an integral part of the management of all types of project.
   6.2. Protect your organization's mobile devices and telework
   There should be security policies and controls for mobile devices and teleworking.

7. **Personnel Security Management**

   7.1. Emphasize security prior to employment
   7.2. Emphasize security during employment
   7.3. Emphasize security at termination of employment

8. **Organizational Asset Management**

   8.1. Establish responsibility for corporate assets
   All information assets should be inventoried, and owners should be identified to be held accountable for their security. 'Acceptable use' policies should be defined, and assets should be returned when people leave the organization.

   8.2. Develop an information classification scheme
   Information should be classified and labelled by its owners according to the security protection needed, and handled appropriately.

   8.3. Control how physical media are handled
   Information storage media should be managed, controlled, moved and disposed of in such a way that the information content is not compromised.

9. **Information Access Management**

   9.1. Respect business requirements

      9.1.1. Access control policy
      The organization's requirements to control access to information assets should be clearly documented in an access control policy and procedures.

      9.1.2. Access to networks and network services
      Network access and connections should be restricted.

   9.2. Manage all user access rights
   The allocation of access rights to users should be controlled from initial user registration through to removal of access rights when no longer required, including special restrictions for privileged access rights and the management of passwords plus regular reviews and updates of access rights.

      9.2.1. User registration and de-registration

      9.2.2. User access provisioning

      9.2.3. Management to privileged access rights

      9.2.4. Management of secret authentication information of users

      9.2.5. Review of user access rights

   9.3. Protect user authentication
   Users should be made aware of their responsibilities towards maintaining effective access controls e.g. choosing strong passwords and keeping them confidential.

   9.4. Control access to systems
   Information access should be restricted in accordance with the access control policy e.g. through secure log-on, password management, control over privileged utilities and restricted access to program source code.

      9.4.1. Information access restriction

      9.4.2. Secure log-on procedures

      9.4.3. Password management system

9.4.4. Use of privileged utility programs
9.4.5. Access control to program source code

10. **Cryptography Policy Management**

There should be a policy on the use of encryption, plus cryptographic authentication and integrity controls such as digital signatures and message authentication codes, and cryptographic key management.

10.1. Control the use of cryptographic controls and keys
10.1.1. Policy on the use of cryptographic controls
10.1.2. Key management

12. **Operational Security Management**

12.1. Establish procedures and responsibilities
There should be a policy on the use of encryption, plus cryptographic authentication and integrity controls such as digital signatures and message authentication codes, and cryptographic key management.
12.2. Protect your organization from malware
Malware controls are required, including user awareness.
12.3. Make backup copies on a regular basis
Appropriate backups should be taken and retained in accordance with a backup policy.
12.4. Use logs to record security events
System user and administrator/operator activities, exceptions, faults and information security events should be logged and protected. Clocks should be synchronized.
12.5. Control your operational software
Software installation on operational systems should be controlled.
12.6. Address your technical vulnerabilities
Technical vulnerabilities should be patched, and there should be rules in place governing software installation by users.
12.7. Minimize the impact of audit activities
IT audits should be planned and controlled to minimize adverse effects on production systems, or inappropriate data access.

13. **Network Security Management**

13.1. Protect networks and facilities
Networks and network services should be secured, for example by segregation.
13.2. Protect information transfers
There should be policies, procedures and agreements (e.g. non-disclosure agreements) concerning information transfer to/from third parties, including electronic messaging.

14. **System Security Management**

14.1. Make security an inherent part of information systems
Security control requirements should be analyzed and specified, including web applications and transactions.
14.2. Protect and control system development activities
Rules governing secure software/systems development should be defined as policy. Changes to systems (both applications and operating systems) should be controlled. Software packages should ideally not be modified, and secure system engineering principles should be followed. The development environment should be secured, and outsourced development should be controlled. System security should be tested, and acceptance criteria defined to include security aspects.
14.3. Safeguard data used for system testing purposes
Test data should be carefully selected/generated and controlled.

15. **Supplier Relationship Management**

15.1. Establish security agreements with suppliers
15.2. Manage supplier security and service delivery

16. **Security Incident Management** (risk assessment)

16.1. Identify and respond to information security incidents
There should be responsibilities and procedures to manage (report, assess, respond to and learn from) information security events, incidents and weaknesses consistently and effectively, and to collect forensic evidence.

17. **Security Continuity Management**

17.1. Establish information security continuity controls
The continuity of information security should be planned, implemented and reviewed as an integral part of the organization's business continuity management systems.
17.2. Build redundancies into information processing facilities
IT facilities should have sufficient redundancy to satisfy availability requirements.

18. **Security Compliance Management**

18.1. Comply with legal security requirements
18.2. Carry out security compliance reviews

4.1.2 ICS-related Security Standards

Until a few years ago, there were hardly any ICS specific security standards. This is due these systems being first developed before the prevalent use of Internet. The small amount of standards that existed focused primarily on providing guidance on physical issues, such as protecting the physical state of the computers on the system and the physical access to them. Since then, SCADA systems have adopted internet-based techniques, exposing them to new threats and vulnerabilities.

Unlike traditional administrative IT systems, ICS systems manage and control critical infrastructures, making them require much higher levels of performance and availability. For this reason, conventional security solutions are not always applicable to ICS systems. This lead to standard-setting bodies developing security standards specifically designed for ICS. There are now a vast amount ICS-related security standards that have been developed by different bodies. Each standard will defer from one another depending on the scope of the standard (some are for specific ICS sectors), the country/zone they were designed for, the county/zone they're enforced on, and other factors.

In this work, we focus on the three most widely recognized and referenced ICS security standards, NIST SP 800-82, ISA/IEC 62443, and NERC CIP. Both NIST SP 800-82 and ISA/IEC 62443 are broader in scope, being developed to be applied to any organization or industry that uses ICS. NERC CIP on the other hand, specifically focuses on the electric utility industry, but is often referenced in the oil & gas refinery industry. Additionally, we reference the Good Practice Guide - Process Control and SCADA Security that is a set of guidelines developed by CPNI that focuses specifically on SCADA industries.

4.1.2.1 ISA/IEC 62443

Formerly known as ISA-99, ISA/IEC 62443 is a series of standards that is organized into four groups (shown in Figure 4.2) that attempt to implement cybersecurity robustness and resilience into industrial automation control systems (IACS).

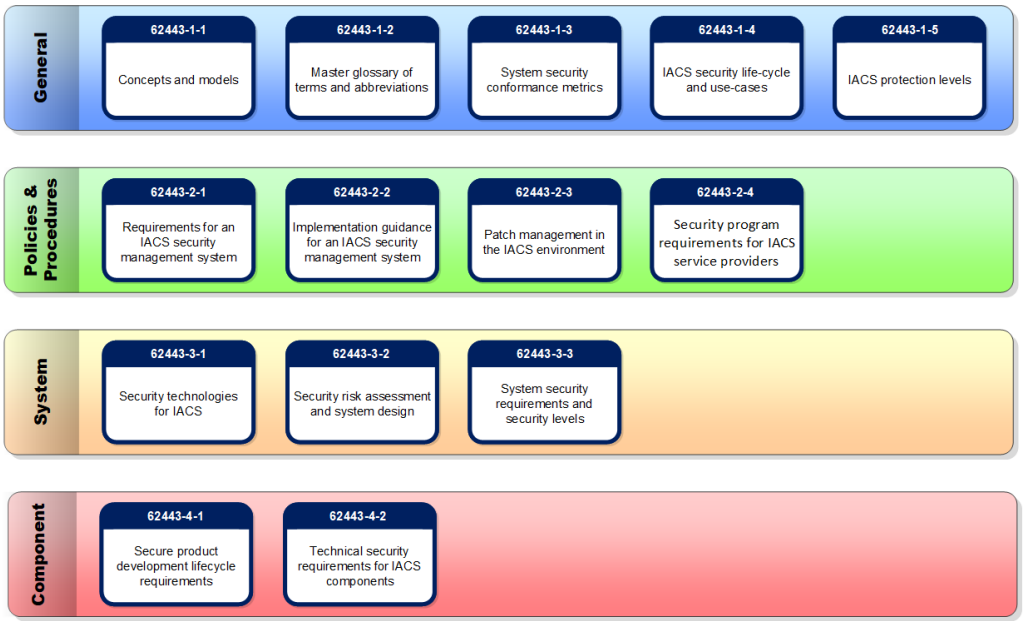


Figure 4.2: 62443 Elements [9]

The series goal is to improve the safety, availability, integrity and confidentiality of components or systems used for industrial automation control, and to provide guidelines for acquiring

and implementing secure IACS. Requirements are also addressed to improve electronic security and help identify and address vulnerabilities. The series is built on already established standards that are used to provide security in general IT systems (ISO 27002), identifying and addressing the important differences present in IACS (relative to IT) [9].

The relevant documents and chapters in this standard [26] are the following:

62443-1-1

As shown in Figure 4.2, this document is part of the general group, that includes documents that address topics that are common to the entire series.

The standard explains the series in general, why it's used, security elements, and introduces definitions, the concepts and models used throughout the series.

62443-2-1

Part of the Policies and Procedures group as seen on 4.2, that focuses on explaining the policies and procedures associated with IACS security.

The standard describes what is required to define and implement an effective IACS cybersecurity management system. Relevant topics in the standard are:

- Description and requirements for an IACS security management system;
- Information security policies;
- Organization of information security;
- Human resource security;
- Asset management;
- Access control;
- Cryptography;
- Equipment;
- Operations security (Protection from malware, Backup, Logging, etc.);
- Communication Security;
- System acquisition, development and maintenance;
- Supplier relationships;
- Information security incident management;
- Information security aspects of business continuity management (continuity);
- Compliance.

62443-2-2

Guidance on what is required to operate an effective IACS cybersecurity management system.

62443-2-3

Guidance on the subject of patch management.

62443-3-1

System requirements group addresses requirements at the system level.

This document is a technical report the describes the application of various security technologies to an IACS environment. It describes technologies such as:

- Authentication and authorization technologies;

- Network protection technologies;
- Encryption technologies and data validation;
- Management, audit, measurement, monitoring, and detection tools;
- Remote access technologies;
- Cybersecurity program.

62443-3-2

Addresses security risk assessment and system design.

The standard has two big topics:

- Establishing zone and conduits;
- Perform a detailed cybersecurity risk assessment on each zone and conduit.

62443-3-3

Describes the foundation system requirements and security assurance levels.

62443-4-1

Component requirements, provides information about the more specific and detailed requirements associated with the development of IACS products.

The standard describes the derived requirements that are applicable to the development of products.

It contains a set of practices for development:

Practice 1. Security management (SM);
Practice 2. Specification of security requirements (SR);
Practice 3. Secure by design (SD);
Practice 4. Secure implementation (SI),
Practice 5. Security verification and validation testing (SV;
Practice 6. Security defect management (DM);
Practice 7. Security update management (PM);
Practice 8. Security guidelines (SG).

62443-4-2

Contains set of derived requirements that provide a detailed mapping of the system requirements to subsystems and components of the system under consideration.

It contains common control system security constraints, the following 7 foundation requirements (FRs) (described in 62433-1-1):

FR 1. Identification and authentication control (CR);
FR 2. Use control (CR);
FR 3. System integrity (CR);
FR 4. Data confidentiality (CR);
FR 5. Restricted data flow (CR);
FR 6. Timely response to events (CR);

FR 7. Resource availability (CR).

It also contains:

- Application requirements (ACR);
- Embedded device requirements (ECR);
- Host device requirements (HCR);
- Network device requirements (NCR).

4.1.2.2 NIST SP 800-82 Rev. 2

The NIST 800-82 special publication, or Guide to Industrial Control Systems (ICS) Security, is probably the most important standard studied in this work. It is the baseline, if not the *de facto*, security standard to be followed by ICS. Many other standards reference this publication and are created with it as a basis [11]. The standard's objective is to provide guidance for securing various ICSs, which include SCADA systems [25]. This is done by providing recommendations instead of hard regulations subject to compliance and enforcement, unlike other standards [11].

The document provides an overview of industrial control systems, reviews typical system architectures and topologies, identifies known threats and vulnerabilities of these systems, and recommends security countermeasures to mitigate the associated risks. It also presents an ICS-tailored security control overlay that is based on NIST SP 800-53 Rev. 4 (Recommended Security Controls for Federal Information Systems and Organizations), to provide a customization of controls that apply to the ICS domain.

As mentioned before, this publication is referenced by many other standards. This is due to the fact that it itself references other ICS standards of great importance, such as ISA/IEC 62443 and CPNI's Good Practice Guide on Firewall Deployment for SCADA.

The second revision of this publication improves the way in which the NIST SP 800-53 is mapped, providing the following enhancements:

- Updates to ICS threats and vulnerabilities;
- Updates to ICS risk management, recommended practices, and architectures;
- Updates to current activities in ICS security;
- Updates to security capabilities and tools for ICS;
- Additional alignment with other ICS security standards and guidelines.

The sections on this document [25] that are relevant are the following:

2. Provides an overview of ICS (including SCADA systems). A comparison between ICS and IT systems is also included in this section.

3. Provides a discussion of ICS risk management and assessment.

4. Provides an overview of the development and deployment of and ICS security program to mitigate the risk of vulnerabilities, in other words, it explains how to implement an ICS Security Risk Management Framework.

5. Provides recommendations for integrating security into network architectures typically found in ICS. This is done by highlighting network segregation practices, recommending architecture defenses, recommending firewall rules for specific services (DNS, HTTP, FTP, telnet, DHCP, SSH, SOAP, SMTP, SNMP, DCOM, SCADA and industrial protocols), and mentioning the importance of:

    - Incorporating NAT;
    - Unidirectional gateways;
    - Preventing MITM attacks;
    - Authentication and authorization;
    - Monitoring, logging and auditing;
    - Incident detection, response, and system recovery.

6. Provides a summary of the management, operational, and technical controls identified in NIST 800-53, presenting recommendations and guidance on how these security controls apply specifically to ICS. It also explains how to execute the risk management framework tasks for ICS, providing guidance on how to apply the following security controls to ICS:

    - Access control (RBAC, web servers, VLAN, dial-up modems and wireless);
    - Awareness and training;
    - Audit and accountability;
    - Security assessment and authorization;
    - Configuration management;
    - Contingency planning (Business continuity planning, disaster recovery planning);
    - Identification and authentication (Password, challenge/response, physical token, smart card, bio-metric);
    - Incident response;
    - Maintenance;
    - Media protection;
    - Planning;
    - Personnel security;
    - Risk assessment;
    - System and services acquisition;
    - System and communications protection (Encryption, VPN (IPsec, SSL, SSH));
    - System and Information Integrity (Virus and malicious code detection, intrusion detection and prevention, patch management);
    - Program management;
    - Privacy controls (8 privacy control families: Authority and purpose; Accountability, audit, and risk management; Data quality and integrity; Data minimization and retention; Individual participation and redress; Security; Transparency; Use limitation)

The document also contains interesting appendixes, that are also relevant to the work, such as:

C. Provides a list of ICS threats, vulnerabilities and incidents;

D. Provides a list of current activities in ICS security;

E. Provides a list of ICS security capabilities and tools;

G. Provides and ICS overlay, listing security controls, enhancements, and supplemental guidance that apply specifically to ICS.

Table G-1 - Lists all security control baselines.

4.1.2.3 NERC CIP

The North American Electric Reliability Corporation (NERC) is an international self-regulatory authority that relies on the diverse and collective expertise of industry participants. Their primary mission is to improve the reliability and security of the bulk power system in North America, by assuring the effective and efficient reduction of risks to them. To reduce these risks, NERC develops and enforces reliability standards; assesses seasonal and long-term reliability; monitors the bulk power system through system awareness; and educates, trains, and certifies industry personnel. As the Electric Reliability Organization (ERO), NERC is subject to oversight by the U.S. Federal Energy Regulatory Commission (FERC) and governmental authorities in Canada [43].

NERC has issued a set of cybersecurity standards, known as the Critical Infrastructure Protection (CIP) standards, to reduce the risk of Bulk Electric Systems (BES) being compromised. BES are electrical generation resources and high-voltage transmission systems above 100 kV. These standards include audit measures and levels of noncompliance that can be tied to penalties [11].

Although the NERC CIP security standards are only enforceable within North American BES, the requirements that they request are technically sound and in alignment with other ICS security standards. The critical infrastructures targeted by these standards utilize common ICS assets and protocols, making the standards relevant to a wider base of ICS operators [11].

The important requirements recommended/enforced by this set of documents [44] are:

CIP-002-5.1a - BES Cyber System Categorization

Requires the identification and documentation of the BES Cyber Assets associated with the Critical Assets that support the reliable operation of the BES.

CIP-003-6 - Security Management Controls

Requires that Responsible Entities have minimum security management controls in place to protect BES.

CIP-004-6 - Personnel & Training

Requires that personnel having authorized Assets or authorized unescorted physical access to BES, including contractors and service vendors, have an appropriate level of personnel risk assessment, training, and security awareness.

1. Security Awareness Program
2. Cybersecurity Training Program

3. Personnel Risk Assessment Program
4. Access Management Program
5. Access Revocation

CIP-005-5 - Electronic Security Perimeter(s)

Requires the identification and protection of the Electronic Perimeter(s) inside which all Critical Cyber Assets reside, as well as all access points on the perimeter.

1. Electronic Security Perimeter

    1.1. All applicable Cyber Assets connected to a network via a routable protocol shall reside within a defined ESP.
    1.2. All External Routable Connectivity must be through an identified Electronic Access Point (EAP).
    1.3. Require inbound and outbound access permissions, including the reason for granting access, and deny all other access by default.
    1.4. Where technically feasible, perform authentication when establishing Dial-up Connectivity with applicable Cyber Assets.
    1.5. Have one or more methods for detecting known or suspected malicious communications for both inbound and outbound communications.

2. Interactive Remote Access Management

    2.1. Utilize an Intermediate System such that the Cyber Asset initiating Interactive Remote Access does not directly access an applicable Cyber Asset.
    2.2. For all Interactive Remote Access sessions, utilize encryption that terminates at an Intermediate System.
    2.3. Require multi-factor authentication for all Interactive Remote Access sessions.

CIP-007-6 - System Security Management

Requires Responsible Entities to define methods, processes, and procedures for securing those systems determined to be Critical Cyber Assets, as well as the other (non-critical) Cyber Assets within the Electronic Security Perimeter(s).

1. Ports and Services (Port management)

    1.1. Where technically feasible, enable only logical network accessible ports that have been determined to be needed by the Responsible Entity, including port ranges or services where needed to handle dynamic ports. If a device has no provision for disabling or restricting logical ports on the device then those ports that are open are deemed needed.
    1.2. Protect against the use of unnecessary physical input/output ports used for network connectivity, console commands, or Removable Media.

2. Security Patch Management (Patching security issues)

2.1. A patch management process for tracking, evaluating, and installing cybersecurity patches for applicable Cyber Assets. The tracking portion shall include the identification of a source or sources that the Responsible Entity tracks for the release of cybersecurity patches for applicable Cyber Assets that are updateable and for which a patching source exists.

2.2. At least once every 35 calendar days, evaluate security patches for applicability that have been released since the last evaluation from the source or sources identified in Part 2.1.

2.3. For applicable patches identified in Part 2.2, within 35 calendar days of the evaluation completion, take one of the following actions:

- Apply the applicable patches; or
- Create a dated mitigation plan; or
- Revise an existing mitigation plan.

Mitigation plans shall include the Responsible Entity's planned actions to mitigate the vulnerabilities addressed by each security patch and a time-frame to complete these mitigations.

2.4. For each mitigation plan created or revised in Part 2.3, implement the plan within the time-frame specified in the plan, unless a revision to the plan or an extension to the time-frame specified in Part 2.3 is approved by the CIP Senior Manager or delegate.

3. Malicious Code Prevention

3.1. Deploy method(s) to deter, detect, or prevent malicious code.

3.2. Mitigate the threat of detected malicious code.

3.3. For those methods identified in Part 3.1 that use signatures or patterns, have a process for the update of the signatures or patterns. The process must address testing and installing the signatures or patterns.

4. Security Event Monitoring (Logging)

4.1. Log events at the BES Cyber System level (per BES Cyber System capability) or at the Cyber Asset level (per Cyber Asset capability) for identification of, and after-the-fact investigations of, Cybersecurity Incidents that includes, as a minimum, each of the following types of events:

4.1.1. Detected successful login attempts;

4.1.2. Detected failed access attempts and failed login attempts;

4.1.3. Detected malicious code.

4.2. Generate alerts for security events that the Responsible Entity determines necessitates an alert, that includes, as a minimum, each of the following types of events (per Cyber Asset or BES Cyber System capability):

4.2.1. Detected malicious code from Part 4.1; and

4.2.2. Detected failure of Part 4.1 event logging

4.3. Where technically feasible, retain applicable event logs identified in Part 4.1 for at least the last 90 consecutive calendar days except under CIP Exceptional Circumstances.
4.4. Review a summarization or sampling of logged events as determined by the Responsible Entity at intervals no greater than 15 calendar days to identify undetected Cybersecurity Incidents.

5. System Access Control (non-physical access control)

5.1. Have a method(s) to enforce authentication of interactive user access, where technically feasible. (User identification)
5.2. Identify and inventory all known enabled default or other generic account types, either by system, by groups of systems, by location, or by system type(s).
5.3. Identify individuals who have authorized access to shared accounts. (Password control)
5.4. Change known default passwords, per Cyber Asset capability
5.5. For password-only authentication for interactive user access, either technically or procedurally enforce the following password parameters:
5.5.1. Password length that is, at least, the lesser of eight characters or the maximum length supported by the Cyber Asset; and
5.5.2. Minimum password complexity that is the lesser of three or more different types of characters (e.g., uppercase alphabetic, lowercase alphabetic, numeric, non-alphanumeric) or the maximum complexity supported by the Cyber Asset.
5.6. Where technically feasible, for password-only authentication for interactive user access, either technically or procedurally enforce password changes or an obligation to change the password at least once every 15 calendar months. (Authentication attempt control)
5.7. Where technically feasible, either:
- Limit the number of unsuccessful authentication attempts; or
- Generate alerts after a threshold of unsuccessful authentication attempts.

CIP-008-5 - Incident Reporting and Response Planning

Ensures the identification, classification, response, and reporting of Cybersecurity Incidents related to Critical Cyber Assets.

1. Cybersecurity Incident Response Plan Specifications (Plan on how to respond to attack)

1.1. One or more processes to identify, classify, and respond to Cybersecurity Incidents. (Report incident)
1.2. One or more processes to determine if an identified Cybersecurity Incident is a Reportable Cybersecurity Incident and notify the Electricity Sector Information Sharing and Analysis Center (ES-ISAC), unless prohibited by law. Initial notification to the ES-ISAC, which may be only a preliminary notice, shall not exceed one hour from the determination of a Reportable Cybersecurity Incident.

1.3. The roles and responsibilities of Cybersecurity Incident response groups or individuals.
1.4. Incident handling procedures for Cybersecurity Incidents.

2. Cybersecurity Incident Response Plan Implementations and Testing

2.1. Test each Cybersecurity Incident response plan(s) at least once every 15 calendar months:
- By responding to an actual Reportable Cybersecurity Incident;
- With a paper drill or tabletop exercise of a Reportable Cybersecurity Incident; or
- With an operational exercise of a Reportable Cybersecurity Incident.

2.2. Use the Cybersecurity Incident response plan(s) under Requirement R1 when responding to a Reportable Cybersecurity Incident or performing an exercise of a Reportable Cybersecurity Incident. Document deviations from the plan(s) taken during the response to the incident or exercise.
2.3. Retain records related to Reportable Cybersecurity Incidents.

3. Cybersecurity Incident Response Plan Review, Update, and Communication (Review, update, and communicate the plan)

3.1. No later than 90 calendar days after completion of a Cybersecurity Incident response plan(s) test or actual Reportable Cybersecurity Incident response:
3.1.1. Document any lessons learned or document the absence of any lessons learned;
3.1.2. Update the Cybersecurity Incident response plan based on any documented lessons learned associated with the plan; and
3.1.3. Notify each person or group with a defined role in the Cybersecurity Incident response plan of the updates to the Cybersecurity Incident response plan based on any documented lessons learned.
3.2. No later than 60 calendar days after a change to the roles or responsibilities, Cybersecurity Incident response groups or individuals, or technology that the Responsible Entity determines would impact the ability to execute the plan:
3.2.1. Update the Cybersecurity Incident response plan(s); and
3.2.2. Notify each person or group with a defined role in the Cybersecurity Incident response plan of the updates.

CIP-009-6 - Recovery Plans for BES Cyber Systems

Ensures that recovery plan(s) are put in place for Critical Cyber Assets and that these plans follow established business continuity and disaster recovery techniques and practices.

1. Recovery Plan Specifications

1.1. Conditions for activation of the recovery plan(s)
1.2. Roles and responsibilities of responders.

1.3. One or more processes for the backup and storage of information required to recover BES Cyber System functionality.
1.4. One or more processes to verify the successful completion of the backup processes in Part 1.3 and to address any backup failures.
1.5. One or more processes to preserve data, per Cyber Asset capability, for determining the cause of a Cybersecurity Incident that triggers activation of the recovery plan(s). Data preservation should not impede or restrict recovery.

2. Recovery Plan Implementation and Testing

3. Recovery Plan Review, Update and Communication

CIP-010-2 - Configuration Change Management and Vulnerability

Prevent and detect unauthorized changes to BES Cyber Systems by specifying configuration change management and vulnerability assessment.

1. Configuration Change Management (Manage System Configuration)

1.1. Develop a baseline configuration, individually or by group, which shall include the following items:
1.1.1. Operating system(s) (including version) or firmware where no independent operating system exists;
1.1.2. Any commercially available or open-source application software (including version) intentionally installed;
1.1.3. Any custom software installed;
1.1.4. Any logical network accessible ports; and
1.1.5. Any security patches applied.
1.2. Authorize and document changes that deviate from the existing baseline configuration
1.3. For a change that deviates from the existing baseline configuration, update the baseline configuration as necessary within 30 calendar days of completing the change.
1.4. For a change that deviates from the existing baseline configuration:
1.4.1. Prior to the change, determine required cybersecurity controls in CIP-005 and CIP-007 that could be impacted by the change;
1.4.2. Following the change, verify that required cybersecurity controls determined in 1.4.1 are not adversely affected; and
1.4.3. Document the results of the verification.
1.5. Where technically feasible, for each change that deviates from the existing baseline configuration:
(Test on simulation before producing)
1.5.1. Prior to implementing any change in the production environment, test the changes in a test environment or test the changes in a production environment where the test is performed in a manner that minimizes adverse effects, that models the

baseline configuration to ensure that required cybersecurity controls in CIP-005 and CIP-007 are not adversely affected; and

1.5.2. Document the results of the testing and, if a test environment was used, the differences between the test environment and the production environment, including a description of the measures used to account for any differences in operation between the test and production environments.

2. Configuration Monitoring

2.1. Monitor at least once every 35 calendar days for changes to the baseline configuration (as described in Requirement R1, Part 1.1). Document and investigate detected unauthorized changes.

3. Vulnerability Assessments

3.1. At least once every 15 calendar months, conduct a paper or active vulnerability assessment.

3.2. Where technically feasible, at least once every 36 calendar months:

3.2.1. Perform an active vulnerability assessment in a test environment, or perform an active vulnerability assessment in a production environment where the test is performed in a manner that minimizes adverse effects, that models the baseline configuration of the BES Cyber System in a production environment; and

3.2.2. Document the results of the testing and, if a test environment was used, the differences between the test environment and the production environment, including a description of the measures used to account for any differences in operation between the test and production environments

3.3. Prior to adding a new applicable Cyber Asset to a production environment, perform an active vulnerability assessment of the new Cyber Asset, except for CIP Exceptional Circumstances and like replacements of the same type of Cyber Asset with a baseline configuration that models an existing baseline configuration of the previous or other existing Cyber Asset.

3.4. Document the results of the assessments conducted according to Parts 3.1, 3.2, and 3.3 and the action plan to remediate or mitigate vulnerabilities identified in the assessments including the planned date of completing the action plan and the execution status of any remediation or mitigation action items.

CIP-011-2 - Cyber Security

Requires information protection to prevent unauthorized access to BES Cyber System Information.

1. Information Protection

1.1. Method(s) to identify information that meets the definition of BES Cyber System Information.

1.2. Procedure(s) for protecting and securely handling BES Cyber System Information, including storage, transit, and use.

2. BES Cyber Asset Reuse and Disposal (Proper reuse and disposal of information)

2.1. Prior to the release for reuse of applicable Cyber Assets that contain BES Cyber System Information (except for reuse within other systems identified in the "Applicable Systems" column), the Responsible Entity shall take action to prevent the unauthorized retrieval of BES Cyber System Information from the Cyber Asset data storage media.

2.2. Prior to the disposal of applicable Cyber Assets that contain BES Cyber System Information, the Responsible Entity shall take action to prevent the unauthorized retrieval of BES Cyber System Information from the Cyber Asset or destroy the data storage media.

4.1.2.4 CPNI – Good Practice Guide - Process Control and SCADA Security

The Centre for the Protection of National Infrastructure (CPNI) is the government authority for protective security advice to the United Kingdom (UK) national infrastructure and are part of the Security Service (MI5). Their goal is to protect national security by helping to reduce the vulnerability of the national infrastructure [45]. They achieve this by bringing together security managers in particular sectors to share experiences and findings, and by developing good practice guides based on these findings that provide good practices that should be followed to prevent threats [46].

Although CPNI develops good practice guides specifically for UK's national infrastructures, these infrastructures generally have common assets, technology and protocols as the infrastructures of other counties. This is also true in the case of ICS. Some of their good practice guides are referenced in other ICS security standards, such as NIST SP 800-82 Rev. 2.

The Good Practice Guide - Process Control and SCADA Security, is a set of guidelines that was designed to prove good practices for securing ICS. It proposes a framework consisting of seven elements for addressing process control security. It's important to note that these guides only provide guidance, and consequently do not provide detailed techniques, solutions, procedures or information on how to implement the recommendations they provide.

This set of guides [47] provides the following recommendations:

Guide 1 - Understand the Business Risk

Provides guidance on assessing the business risk and ongoing assessment of this risk. It shows good practices on how to apply a risk assessment framework, to evaluate risk.

Guide 2 - Implement Secure Architecture

Provides good practice guidance on deciding on appropriate security architecture for process control systems (PCS).

Guide 3 - Establish Response Capabilities

Provides guidance on establishing response capabilities relating to digital security threats in PC and SCADA systems. It provides good practices to respond to failures, such as continuity plans, incident response, warning systems, processes and procedures, and incident reporting.

Guide 4 - Improve Awareness and Skills

Develops the element by looking in detail at each of the key areas and provides generic guidance on improving PC security skills within organizations.

Guide 5 - Manage Third Party Risk

Provides good practice guidance managing third party risks to process control system security, such as how to manage risk from vendors, risk from support organizations, and risk in the supply chain.

Guide 6 - Engage Projects

Provides good practice guidance on building security considerations into process control security projects.

Guide 7 - Establish Ongoing Governance

Provides good practice guidance for defining and implementing appropriate governance frameworks for process control systems security.

4.2 Analysis of a Client's Requisites

Ever since SCADA systems adopted internet-based techniques, they have been exposed to a larger number of cyber-threats. As time goes by, the number of vulnerabilities discovered in these systems keeps increasing. Due to these issues, cybersecurity has become an important topic for both SCADA developers and SCADA clients.

When a client wants to purchase a SCADA system, they will generally provide a set of specifications, or requisites, to the developers indicating what features they desire on their SCADA system. Due to the cybersecurity issues mentioned before, clients have started to include cybersecurity requisites in their sets of specifications. Since cybersecurity standards are one of the most trustworthy documents regarding this topic, clients will formulate their security requisites based on these standards.

In this section, we suggest an approach on how to perform an analysis of a generic client's cybersecurity requisites, and present solutions to assure their compliance. In this analysis, we will assume that the developer of the SCADA system is capable of providing the necessary segregation and firewall protection during the configuration of the SCADA system. That said, we start our analysis by dividing all the requisites into ten main groups that represent the following:

- **Communications:** Network segregation, firewall protection, communication channel encryption.
- **Access Control:** Access control and authentication.
- **Data Protection:** Private and personal data protection.
- **Database Security:** Database good practices.

- **Patch Management:** Software updates and tests.
- **Monitoring and Logging:** Monitoring system and security events, and log generation of these events.
- **Backups:** Backup mechanisms and disaster recovery.
- **Compliance:** Product documentation and vulnerability tests.
- **General:** General good practices.
- **Development:** Coding good practices.

This division has two purposes. The first is to allow SCADA developers to focus on one area at a time when verifying/validating the requisites compliance. The second purpose is for our second step in the cybersecurity requisite analysis. By using the groups we divided and the cybersecurity standard overview that we performed in Section 4.1, we are able to quickly associate each requisite with a respective standard recommendation, control and/or requirement, allowing developers to verify/validate not only the requisite's compliance, but also compliance with the associated standards.

4.2.1 Communications

No.	Requisite	Related Standards
1.1	The Energy Management System (EMS) should support users remote access through Client A's local net or DMZ	- CIP-005-5: R2 - NIST 800-82 r2: 5.1/ 5.2/ 5.4/ 5.5 - ISO 27002:2013: 13.1.3 - ISA-62443-2-1: 13.1.3/ C.4.3.6.5.3

This procedure is generally done when configuring the SCADA system, during the development of the client's SCADA system.

1.2	Communications between EMS and user interfaces should be established throughout secured connections (standard protocols and algorithms), guaranteeing confidentiality of communications and data integrity (namely SSH/SSL3, HTTPS)	- NIST 800-82: 5.8.6/ 6.2.16.2 - ISA-62443-2-1: C.4.4.4.3.2 - ISA-62443-3-1: 7.2 - OWASP Secure Coding Practices: Session Management

Proposed solution:

When selecting cryptographic protocols to implement secure communication channels, we strongly recommend avoiding all versions of SSL, and TLS versions below 1.2, due to the lack of security found in these protocols. All versions of SSL were prohibited and/or deprecated by the Internet Engineering Task Force (IETF) due to serious security flaws in the protocols, as shown on Request for Comments (RFC) 6176 [48] for SSL2 and RFC 7568 [49] for SSL3. Being the

successor of SSL, TLS is considered to be much more secure than its predecessor. Although none of TLS's versions are currently deprecated, advances are being made by the IETF to deprecate versions 1.0 and 1.1 of the protocol by 2020 due to security flaws found in them, as shown on [50]. In addition to this, NIST also recommends using TLS 1.3 or, at a minimum, 1.2 in NIST SP 800-52 r2 [51]. Despite SSL3 being mentioned in this requisite, we will assume that it was an error committed by the client as on 1.17 TLS 1.2 is requested.

That said, this issue can be resolved by implementing TLS on any proprietary or standard protocols, using OpenSSL[1] for example, which supports both versions 1.2 and 1.3. In order to support secure TLS, HTTPS, and SSH, the system should assure Forward Secrecy (FS), and use multiple and diverse private keys that are generated for each EMS.

The NIST SP 800-113 [52] provides recommendations and explains how to establish secure communications following the TCP/IP model:

- Application layer: This layer should be protected using HTTPS and SSH. For the specific case of the ICCP that uses TASE.2, the security guides provided by the Electric Power Research Institute (EPRI) should be followed [53].
- Transport layer: Both TCP, and UDP communications can also be secured using TLS.
- Network/Internet layer: This layer should be secured using IPsec and SSL tunnel VPNs. If the devices in question are legacy systems, a Bump-in-the-Wire (BITW) VPN can used. A BITW VPN is a device or security appliance that can secure a VPN communication.

When files are transferred, secure FTP (SFTP/SSH File Transfer) is also required.

Additional resources:

Guidance on the security recommendations proposed can be found on the IEC 62351 standard series [54]. The standard specifies the importance of these security measures and how they are implemented.

Guidance on selection, configuration, and use of TLS can be found on the NIST SP 800-52 r2 [51] (includes TLS 1.2 and 1.3).

Additional guidance on how to implement TLS 1.3 using OpenSSL can be found on their official Wiki page on [55].

No.	Requisite	Related Standards
1.3	DMZ/FTP/WEB Servers: Network where FTP/WEB Servers should be located. Secured Internet access should be available (minimum Https + Authentication)	- NIST 800-82 r2: 5.8.6/ 6.2.16.2 - ISA-62443-2-1: C.4.4.4.3.2 - OWASP Secure Coding Practices: Transport Layer Protection

Proposed solution:

[1] OpenSSL, Cryptography and SSL/TLS Toolkit. https://www.openssl.org/

By following the authentication guidance on requisite 2.7 and implementing HTTPS as described on 1.2, this should be achieved.

1.4	SCADA: Network specific for SCADA Servers that communicate with RTUs at Substations. This network should be accessed by OSs and External Gateways	- CIP-005-5: R1 - ISO 27002:2013: 13.1.3 - ISA-62443-2-1: 13.1.3
1.5	Intranet Client A: Client A's Global Intranet Network. Client A's private network. It consist of many interlinked local area networks with a wide area network	- CIP-005-5: R1 - NIST 800-82 r2: 5.1/ 5.2/ 5.5 - ISO 27002:2013: 13.1.3 - ISA-62443-2-1: 13.1.3
1.6	Operating Stations (OS): Network where Operators' terminals should be located. These terminals should mostly communicate with SCADA Servers and eventually with Substations' RTUs	- CIP-005-5: R1 - NIST 800-82 r2: 5.1/ 5.2/ 5.5 - ISO 27002:2013: 13.1.3 - ISA-62443-2-1: 13.1.3
1.7	External Gateways: Network where external Gateways should be located. These servers should provide information to other entities and should communicate with data centers	- CIP-005-5: R1 - ISO 27002:2013: 13.1.3 - ISA-62443-2-1: 13.1.3

Generally, all networks that are requested by the client are configured accordingly either during pre-production or during the installation of the system, depending on whether the network has to be created from scratch or is provided by the client.

Additionally we recommend using data diodes to filter the data being transmitted between these networks. Only allowing the transfer of data that is essential for the systems function reduces the surface on which an attacker can operate, therefore increasing the security of the system.

1.8	All these network zones should be isolated with a firewall and should get restricted access only for the what should be strictly needed for operating	- CIP-005-5: R1: 1.1 - NIST 800-82 r2: 5.1/ 5.2/ 5.5/ 6.2.1.3 - ISO 27002:2013: 13.1.3 - ISA-62443-2-1: 13.1.3 - ISA-62443-3-1: 6.5
1.9	Communication between the Dispatch Center with remote units, external entities, Internet and others, should be protected with firewalls acting as security perimeter	- CIP-005-5: R1: 1.2 - NIST 800-82 r2: 5.3/ 5.5/ 5.7 - ISO 27002:2013: 13.1.2/ 13.1.3 - ISA-62443-2-1: 13.1.2/ 13.1.3 - ISA-62443-3-1: 6.1/ 6.2/ 6.4 - ISA-62443-4-1: Practice 8
1.10	Firewalls should be combined with different VLAN's for traffic split in order to provide different levels of security for each network zone	- CIP-005-5: R1: 1.1 - NIST 800-82 r2: 5.1/ 5.2/ 5.5/ 6.2.1.3 - ISO 27002:2013: 13.1.3 - ISA-62443-2-1: 13.1.3 - ISA-62443-3-1: 6.5

In most cases, firewalls are configured during the installation of the SCADA system, as the firewalls are provided by the client and only require restriction rules for the ports used. But, if needed, firewalls can also be provided to the client, being fully configured during pre-production.

VLANs can also be configured in the same way, according to the firewall being used.

In terms of proprietary firewall solutions, we strongly recommend using Waterfall's[2] Industrial Applications & Protocols solution for EMS.

1.11	All links to / from Internet should be made through VPN Tunnels and IPSec with strong encryption	- NIST 800-82: 6.2.16.2 - ISO 27002:2013: 9.1.2 - ISA-62443-2-1: 9.1.2 - ISA-62443-3-1: 6.6 - ISA-62443-4-2: A.1.3.2 - NIST 800-77

Proposed solution:

SCADA systems normally only allow internet access to servers/machines that require remote access. For these specific servers, VPN tunnels and IPsec can be implemented following the NIST SP 800-113 recommendations provided on 1.2. Once again, we recommend assuring that the system provides Forward Secrecy (FS) in order to implement secure IPSec. Although not

[2]Waterfall, Stronger then Firewalls. https://waterfall-security.com/solutions

recommended, if other types of servers require internet access, this feature can be adapted to those servers.

Additional resources:

Guidance on how to implement a secure IPSec solution using Probabilistic Signature Scheme (PSS) is shown on [56].

1.12	Different security zones are drawn in Figure A, with the main equipment connected to each network represented	- CIP-005-5: R1 - NIST 800-82 r2: 5.1/ 5.2/ 5.5 - ISO 27002:2013: 13.1.3 - ISA-62443-2-1: 13.1.3

All types of network segregation are done during the configuration of the system, either during pre-production or installation.

Note: For confidential reasons, Figure A is not displayed in this document. It simply consists of a network drawn by the client with various zones and equipment contained in each zone.

No.	Requisite	Related Standards
1.13	RTU: Network for RTUs. It should be the only one communicating with dispatch centers	- CIP-005-5: R1 - NIST 800-82 r2: 5.1/ 5.2/ 5.5 - ISO 27002:2013: 13.1.3 - ISA-62443-2-1: 13.1.3
1.14	Other Safe systems: Network where other safe systems are located. Safe systems should communicate with RTUs for information sending	- CIP-005-5: R1 - NIST 800-82 r2: 5.1/ 5.2/ 5.5 - ISO 27002:2013: 13.1.3 - ISA-62443-2-1: 13.1.3
1.15	Corporate Network: (Client A Intranet) it should be present at Substations and should be isolated from Substations' Network and RTU Zone	- CIP-005-5: R1 - NIST 800-82 r2: 5.1/ 5.2/ 5.5 - ISO 27002:2013: 13.1.3 - ISA-62443-2-1: 13.1.3
1.16	Other Systems: Network for other systems and third parties. It should be isolated or access should be strictly controlled. The access control should be costumer responsibility	- CIP-005-5: R1 - NIST 800-82 r2: 5.1/ 5.2/ 5.5 - ISO 27002:2013: 13.1.3 - ISA-62443-2-1: 13.1.3

Although these networks are part of a different zone, they are configured in the same manner as 1.4-1.7.

No.	Requisite	Related Standards
1.17	RTU: Communications and Cryptography: Every communication with the server must be encrypted regardless of the utilized channel (web, mobile or other). When there are no constraints involved, the newest versions of the protocols must be used (e.g. HTTPS, TLS1.2).	- CIP-005-5: R2 - NIST 800-82 r2: 5.8/ 6.2.16/ SC-8/ SC-13 AC-17(2)/ AC-18(1)/ AC-19(5) - ISO 27002:2013: 10.1 - ISA-62443-2-1: 10.1 - ISA-62443-3-1: 7 - IEC-62351-13:2016: 6 - OWASP Secure Coding Practices: Transport Layer Protection

This requisite should be assured by following the recommendations proposed on 1.2, as measures to assure cryptographic communications are also proposed there.

1.18	OS Firewall: It must be guaranteed that the OS firewall is active and configured to allow only necessary communications.	- CIP-005-5: R1: 1.3/ 1.5 - NIST 800-82 r2: 5.3/ 5.7 - ISO 27002:2013: 13.1.2/ 13.1.3 - ISA-62443-2-1: 13.1.2/ 13.1.3 - ISA-62443-3-1: 6.1/ 6.2/ 6.4 - ISA-62443-4-1: Practice 8

This is a standard procedure when developing all SCADA systems.

No.	Requisite	Related Standards
1.19	Backdoor Prevention: Information must be provided about all connections and ports available on the communications network from the application to ensure that there are no backdoors.	- NIST 800-82 r2: CA-7/ SI-4 - ISO 27002: 2013: 9.1.2/ 13.1.1 - ISA-62443-2-1: 9.1.2/ 13.1.1 - ISA-62443-3-3: SR 6.2 - ISA-62443-4-2: CR 6.2

SCADA developers always provide information to their clients on all ports, communications and protocols that are vital for the functionality of the system. If unnecessary, they are disabled.

1.20	Network Access Control: It must be ensured that the system operates in a network which is adequately segregated, where it is possible to identify all terminals and block access from unauthorized terminals.	- CIP-007-6: R1 - NIST 800-82 r2: 5.1/ 5.5/ SC-7 - ISO 27002: 2013: 9.1.2/ 13.1/ 13.2.1/ 14.1.2/ 14.1.3 - ISA-62443-2-1: 9.1.2/ 13.1/ 13.2.1/ 14.1.2/ 14.1.3

Segregation of the network will be done according to the requests provided on 1.3 and 1.13-1.16. Once again, this segregation can be reinforced with data diodes to only allow unidirectional access from trusted terminals.

4.2.2 Access Control

2.1	EMS passwords and log-in should be managed with secure criteria and reinforced with others additional mechanisms (eg: captcha, sms, etc.)	- CIP-005-5: R2: 2.3 - NIST 800-82 r2: 6.2.7/ 6.2.7.2/ IA-8 - ISA-62443-3-1: 5.3.4

Proposed solution:

As recommended by NIST in SP 800-82 r2 [25], all types of authentication used to gain access to the ICS network should be secure. This security can be provided using a combination of complex passwords and multi-factor authentication methods. Multi-factor authentication (MFA) is a method of authentication which involves a user successfully satisfying two or more factors in an authentication mechanism. These factors are knowledge (Ex: passwords, secret questions), possession (Ex: security tokens: OTP, physical, contactless; SMS), and inherence (Ex: biometric). The most common subset of MFA is Two-factor authentication (2FA) which requires two of the factors mentioned to be satisfied. These factors are generally knowledge (password) and possession.

Although SMS is one of the most used and requested MFA possession factors, we recommend avoiding it, as it is one of the unsafest methods available [57]. NIST has even stopped recommending this authentication method in their SP 800-63B [58]. For a greater reinforcement, and if possible, non-SMS forms of two-factor authentication, such as code generators on devices, are recommended, as they provide stronger security [57].

If SMS is the only MFA factor option, other than knowledge (password), available, we recommend using it as it still provides more security than just a password requirement. That said, reinforcing the log-in using a SMS-based (one-time password) two-factor authentication is not only more secure than a standalone password or captcha solution, but it also improves user experience. SMS PINs are also completely customizable. Although Captcha[3] isn't part of any MFA

[3]BotDetect CAPTCHA Generator for multiple services https://captcha.com/

factor, it can provide protection to authentication mechanisms by prevent abuse via automated scripts, spam and bots. It has the advantage of being easy to implement and should only be used if MFA isn't an option or as an extra layer of protection in conjunction with MFA.

Additional resources:

Owasp provides tests made to advanced OTP (one-time password) multi factor authentication methods, such as "User ID, password, and Disposal password", "One Time Password Tokens", "Crypto Devices with certificates (Token USB, Smart Cards)", among others. The article evaluates the strength of each method and provides guidance on their implementation [59].

No.	Requisite	Related Standards
2.2	EMS should be able to manage accesses and permissions regarding each user profile	- NIST 800-82 r2: 6.2.1.1 - ISO 27002:2013: 9.1/ 9.2.3/ 9.2.4 - ISA-62443-2-1: 9.1/ 9.2.3/ 9.2.4 - ISA-62443-3-1: 5.1 - ISA-62443-4-1: SD-6/ Practice 8 - ISA-62443-4-2: 3.4/ CR 1.1/ CR 2.1

Proposed solution:

Guidance on user permissions is described in more detail on 2.11.

2.3	EMS should be integrated with Client A's Active Directory	- ISA-62443-2-1: C.4.3.6.5.2

Proposed solution:

This is done during the configuration phase. Additional information is required from the client to determine if the Active Directory should only apply to graphical sessions or to all sessions.

When using Active Directory, there is a feature known as Group Policies that can control the working environment of user accounts and computer accounts, essentially implementing Role-based Access Control (RBAC) to user accounts. If an AD is used, the requirement on 2.12 should be satisfied using this feature.

Additional resources:

Guidance on how to implement RBAC using Group policies can be found on Microsoft's official doc website on [60].

2.4	Users should be requested with username and password for EMS access, and reinforced with others additional mechanisms (eg: captcha, sms, etc.)	- CIP-005-5: R2: 2.3 - NIST 800-82 r2: 6.2.7/ 6.2.7.2/ IA-8 - ISA-62443-3-1: 5.3.4

Proposed solution:

This requisite can be satisfied by following the recommendations mentioned on 2.1.

2.5	Every user should be assigned to a profile/role with customized permissions according to it (allowing the access just to the specific profile enabled functionalities)	- CIP-004-6:R4: 4.1 - NIST 800-82 r2: AC-6 (6.2.1.1) - ISO 27002:2013: 9.1/ 9.2.3/ 9.2.4 - ISA-62443-2-1: 9.1/ 9.2.3/ 9.2.4 - ISA-62443-3-1: 5.1 - ISA-62443-4-1: SD-6/ Practice 8 - ISA-62443-4-2: 3.4/ CR 1.1/ CR 2.1

Proposed solution:

Can be achieved following the proposed solution on 2.11. The access control proposed implements RBAC in a way that makes it customizable.

2.6	The log-in session should expire due to timeout time (predefined with no user activity)	- NIST 800-82 r2: AC-2 (5) - ISO 27002: 2013: 9.4.2 - ISA-62443-2-1: 9.4.2 - OWASP Secure Coding Practices: Session Management

Proposed solution:

This option is usually is set by default on non-graphical sessions, such as SSH and TLS connections. For graphical sessions such as HMIs and Engineering workstations, should be implemented during development.

NIST's SP 800-82 r2 recommends employing non-automated mechanisms (including login) or procedures so inactivity may be detected [25]. Auto-login mechanisms are considered unsafe since various operators share the same HMI/workstation. If an operator has malicious intents, he can auto-login on an administrator session that wasn't terminated or use a fellow operators account to conceal his identity.

2.7	Authentication: All available services must request authentication (Oauth2/HMAC).	- CIP-007-6 Table R5: 5.1 - NIST 800-82 r2: 5.15/ 6.2.7/ IA-2 - IA-8 ******

Proposed solution:

Services should require not only the authentication of users, but also require authorization to access such services.

OAuth2 [4] is an authorization framework/open standard that enables applications to obtain limited access to user accounts on an HTTP service. If Oauth is to be used, the client must specify if they will provide an Oauth server, if a 3rd party will provide it or if it is to be developed by the SCADA developer.

HMAC (hash-based message authentication code) is a mechanism for message authentication using cryptographic hash functions. It can be used with a cryptographic hash functions such as MD5, SHA-1 or SHA-256 in combination with a secret shared key. The strength of HMAC's cryptography depends on the properties of the hash function [61]. Although quite antique, this mechanism is one of the main ways to provide message authentication, as we can use ever more advanced cryptographic hash functions as long as they are iterative.

Additional resources:

If needed, OAuth's official web-page offers guidance on how this framework is implemented, including the latest OAuth2.

Both how the definition of HMAC was deduced and some implementations notes are provided on [61], but this document only provides guidance on implementing HMAC with the MD5 and SHA-1 hash functions. For more advanced techniques, the IEC-62351-5 standard provides guidance on implementing a challenge-response authentication mechanism using HMAC, with pre-shared secret keys for integrity protection of data [54].

2.8	Fail-safe System: The system must be fail-safe. If a fault occurs, it must fail in a secure manner, keeping all security controls unaltered. (e.g. there is no possibility of bypassing authentication during failure, or leaving uncompleted tasks such as writing or querying a database)	- NIST 800-82 r2: SI-17 (5.13)

According to the NIST standard associated to the requirement, if a component fails, it should fail in a manner that does not generate unnecessary traffic on the ICS, or does not cause another problem elsewhere, such as a cascading event. The meaning of "loss of communications" should be defined. The appropriate fail-safe process for the industry should be defined. Fail-safe procedures should be defined and may vary among baselines. The same failure event may trigger different response depending on the impact level. Backups should be performed using the "backup-in-depth" approach, with layers of backups [25].

Whenever the server fails, security controls either return a message of success if they were able to be implemented before the server failed, or return a message that the operation was aborted, guaranteeing that the systems fails in a safe way. In addition, the system can also include a redundant server (for most relevant servers). The server can be kept up to date with the main server with either a cold or hot standby state, meaning that when it commutes, it's state is basically identical to the server that failed (hot standby is more accurate then cold standby), maintaining functionality

[4] OAuth, Open Authentication https://oauth.net/

and allowing aborted operations to be resent. After discussing the previous definitions with the client, this can be assured.

2.9	Unauthorized Software: The system must control the utilization of unauthorized software, as well as fetching files from external sources	- NIST 800-82: CM-7/ CM-10/ CM-11 - ISO 27002:2013: 12.2/ 12.5.1/ 12.6.2 - ISA-62443-2-1: 12.2/ 12.5.1/ 12.6.2

Proposed solution:

The system should apply a software usage restrictions, as well as restrictions to software installation. This can be done following a least functionality approach, when implementing the minimum privilege of 9.1 and RBAC on 2.12. In terms of fetching files from external sources, the system should only allow the transfer of files between authorized users using the same RBAC principle.

Additional resources:

The importance of this measure and to whom privileges should be granted are provided on the ISO [42] and ISA [26] standards attached. Controls for this measure are also provided by NIST [25].

2.10	Unauthorized Access: The system must deny unauthorized access from editors, compilers and other utility software coming from the production environment	- NIST 800-82: CM-7/ CM-10/ CM-11 - ISO 27002:2013: 12.2/ 12.5.1/ 12.6.2 - ISA-62443-2-1: 12.2/ 12.5.1/ 12.6.2

By applying restrictions to software usability the same way as in 2.9, this should be assured.

No.	Requisite	Related Standards
2.11	Access Control: Access control mechanisms must be applied to ensure that the system is protected from any unauthenticated or unauthorized access. Integration with a centralized user management system must be guaranteed (preferably AAA, or Active Directory), allowing privilege differentiation by user and user group	- CIP-004-6: R4/R5 - CIP-007-6: R5 - NIST 800-82 r2: 6.2.1/ AC-2/ AC-6/ AC-17/ AC-19/ IA-4/ IA-5 - ISO 27002: 2013: 9.1/ 9.2.2-9.2.6/ 9.3.1/ 9.4.1 - ISA-62443-2-1: 9.1/ 9.2.2-9.2.6/ 9.3.1/ 9.4.1 - ISA-62443-4-1: SD-1/ SG-6

The client should define what they would like to use: their AD, an AD developed by the SCADA developer or an external server that has an AD implemented. After being established, privilege differentiation can be implemented on the AD.

Since the AD can fail, the preferred fallback should be specified. This fallback should be a secure one, as using an unsafe fallback can allow backdoors or credential theft.

Additionally, if a protocol that implements AAA is opted, the client should specify their preferred protocol in order for it to be implemented.

Additional resources:

The NIST SP 800-120 provides guidance on how to implement the two most known AAA protocols: RADIUS and Diameter [62].

2.12	Role-based Access Control: Role-based access control (RBAC) must be implemented to ensure that it is not possible to dodge security controls (e.g. privilege escalation)	- CIP-004-6:R4 - NIST 800-82 r2: IA-2 (6.2.1.1) - ISA-62443-3-1: 5.1 - IEC-62351-8

Proposed solution:

RBAC is a security process that assigns specific rules or policies (privileges/access) to individual users or groups of users [63]. This security process is valuable, as it ensures users have limited access to services, as they are provided with only what they need, reducing the exposure of valuable services. There are four types of RBAC, flat, hierarchical, constrained and symmetric [64]. The client should provide information on which type of RBAC they prefer.

This security process should be implemented in conjunction with the AD/AAA from 2.11.

Additional resources:

Guidance on RBAC implementation can be found on NIST's publication that standardized the security process [64].

The IEC 62351-8 standard provides guidance on how to implement RBAC in power systems. It covers access control not only to users, but also automated agents [54].

2.13	Unauthorized Access Detection: The system must log unsuccessful authentication attempts, and an alarm must be issued when a specific number of attempts (configurable) is reached for a given time, blocking the respective account	- CIP-007-6:R4/ R5: 5.7 - NIST 800-82 r2: 5.16/ AC-7 - ISO 27002: 2013: 9.4.2/ 12.4.1 - ISA-62443-2-1: 9.4.2/ 12.4.1/ C.4.3.6.5.3 - ISA-62443-3-3: SR 1.11 - ISA-62443-4-2: CR 1.11 - OWASP Secure Coding Practices: Authentication

Unsuccessful authentication attempts must be logged on both security and system logs, as mentioned on 6.2.

There should be blocking mechanism in both graphical and non graphical sessions. Non graphical sessions can easily implement this procedure on the respective operating systems (OS). When implementing the amount of attempts to raise the alarm, it should be configurable, not static.

Proposed solution:

NIST standards recommend employing limited login attempt mechanisms to prevent unauthorized access. In order for this to be implemented, the source code of this procedure would have to be changed.

Additional resources:

This is a standard procedure required to prevent brute-force attacks, with the addition of it being logged.

Since this procedure can only be implemented on the source code of the SCADA systems (for graphical sessions), good coding practices are provided by OWASP on their authentication chapter to prevent brute force attacks [65].

2.14	Unauthorized Access Unblock: There must be the possibility of unblocking automatically an account that has been blocked after a certain time (configurable)	- ISO 27002: 2013: 9.2.2 - ISA-62443-2-1: 9.2.2 - OWASP Secure Coding Practices: Authentication

If there is a blocking mechanism, there has to be an unblocking mechanism. Usually this mechanism is time-based, blocking the user that is trying to be accessed for a certain time (timeout). When implementing the timeout limit, it should be configurable and not static.

This requisite has the same challenges as mentioned on 2.13.

Additional resources:

OWASP's authentication chapter also provides guidance and good practices when implementing this timeout mechanism [65].

2.15	Idle Time: The system must allow configuring an idle time to terminate user sessions.	- NIST 800-82 r2: AC-2 (5) - ISO 27002: 2013: 9.4.2 - ISA-62443-2-1: 9.4.2 - OWASP Secure Coding Practices: Session Management

The standards on this requisite recommend applying this feature to prevent inactive accounts from being accessed.

This feature should be configured for both graphical and non-graphical user sessions as mentioned on 2.6.

Additional resources:

OWASP's good practice guide also contains a chapter that provides session management guidance [65].

2.16	Password Strength: The application must enforce a password policy to local user accounts. This policy should support at least the criteria defined in Client-A-POL. It must be configurable only by accounts with administration privileges.	- CIP-007-6: R5: 5.5/ 5.6 - NIST 800-82 r2: 6.2.7.1 - ISO 27002: 2013: 9.4.3 - ISA-62443-2-1: 9.4.3 - ISA-62443-4-2: CR 1.7 - OWASP Secure Coding Practices: Implement Proper Password Strength Controls

Password strength policies are generally only configurable during the configuration of the system. And can't be changed by admins or any other user. This feature can only be implemented by altering the SCADA system's source code.

The configuration of this feature should require administration privileges, this can be configured when implementing the minimum privilege of 9.1 and RBAC on 2.12.

2.17	Changes of Password: It must be guaranteed that during the process of changing a password, the existing one must be entered prior to accepting the new password. When passwords are successfully changed, the system should forward a message to the email address of the owner of the user id, and the user should be forced to re-authenticate.	- ISO 27002: 2013: 9.4.3 - ISA-62443-2-1: 9.4.3

All these measures should be implemented to provide a secure password alteration. Once again, this can only be implemented by altering the SCADA system's source code.

Additional resources:

Additional information on password change policies can be found on OWASP's article [65]. How to test the strength of the password change or reset policy can also be found on [66].

2.18	Communication Sessions: The system must authenticate or allow configuring authentication for every communications session established with other devices or systems.	- NIST 800-82 r2: 5.3/ 5.8/ SC-23

Proposed solution:

This can be implemented by follow the guidance provided on 1.2 in conjunction with the firewall rules of 1.18.

2.19	Data Access: User access must be restricted to data and information which is really needed for performing his/her tasks.	- NIST 800-82 r2: AC-3 - ISO 27002: 2013: 9.4.1 - ISA-62443-2-1: 9.4.1

Proposed solution:

This can be fulfilled by adding user restrictions to data and information when implementing the minimum privilege principle on 9.1 and RBAC on 2.12.

4.2.3 Data Protection

3.1	Sensitive Data: Transmission of personal data [GDPR] and sensitive personal data [GDPR2] must be encrypted according to the recommendations in ENISA report . Internal data must be encrypted only when intended to be and/or used outside the Organization internal infrastructure	–

Proposed solution:

As from May 2018, this will be an obligation for all organizations operating within the European Union (EU) or organizations outside the EU which offer goods or services to customers or businesses in the EU. It will only apply to personal data that is to be handled by the company. According to Enisa, all of the companies that manage personal data are obligated to protect it from misuse and exploitation. This includes encrypting personal data used in transmission, storage, logging and circulation. If the encryption of this data, or logs containing this data is not possible, it should at least be digitally signed to provide the authenticity, integrity and non-repudiation of the data.

Additional resources:

ENISA provided guidelines on how to securely manage personal data [67].

3.2	Plain Text: Passwords or other confidential information must not be stored in plain text or any other unencrypted form.	- NIST 800-82 r2: 6.2.7.1 - ISO 27002: 2013: 9.4.2/ 9.4.3 - ISA-62443-2-1: 9.4.2/ 9.4.3

Although passwords and other confidential should always be encrypted when stored, the type of encryption chosen should be a secure one. NIST has even deprecated both MD5 and SHA-1, as collision attacks against these encryptions are quite affordable nowadays. They encourage the use of SHA-256 at a minimum [68].

No.	Requisite	Related Standards
3.3	Private Data: The system must store all critical and potentially private data in a secure manner, protecting them from leakage or unauthorized modification (database encryption).	- NIST 800-82 r2: 6.2.19 - ISO 27002: 2013: 18.1.4 - ISA-62443-2-1: 18.1.4

Proposed solution:

The database should encrypt private data to minimize the damage an attacker can cause when it's compromised. Most SCADA systems use databases that are Oracle or SQL based [69]. In exceptional cases, implementing this feature can be quite difficult, as database encryption is quite challenging on older versions of Oracle and SQL. For these cases, the database should be upgraded to the latest (or at minimum a newer) version before applying encryption.

Additional resources:

Oracle has a white paper that provides best practices for applying transparent data encryption on oracle databases [70].

3.4	Personal Data Integrity: The system must monitor personal data integrity, and give alarms in case of unwanted change or data vanishment.	- NIST 800-53 r4: DI-2 - NIST 800-82 r2: 6.2.17/ SI-1/ SI-7 - ISO 27002: 2013: 12.2.1/ 14.1.2/ 14.1.3 - ISA-62443-2-1: 12.2.1/ 14.1.2/ 14.1.3/ B.4/ B.17 - ISA-62443-3-3: SR 3.1/ SR 3.4 - ISA-62443-4-2: CR 3.1/ CR 3.4

Proposed solution:

This can be achieved by following the guidance provided on 3.3 and by implementing a data integrity policy and verification mechanisms.

3.5	Personal Data Partition: Personal data must be partitioned ("data subject" information and identification must be stored in different tables), in order to reduce the possibility of creating correlations and prevent data leakage	- NIST 800-53 r4: SC-32

Additional information is needed from to the client, as this feature is quite challenging to implement. It requires comparing two different tables for the same user (personal data can't have a correlation on the table that associates it with the user).

4.2.4 Database Security

No.	Requisite	Related Standards
4.1	Parameterized Queries: Parameterized queries must be used to prevent unwanted changes in requests. It can be thus assured that parameter values are combined with the compiled instruction, not with the SQL string.	- NIST 800-82 r2: SI-10 - OWASP Secure Coding Practices: DB security

Proposed solution:

OWASP recommends the use of parameterized queries to prevent queries from being altered. Queries used in this manner are recommended because the parameter values (date) are combined with the compiled statement, not a SQL string [71].

Additional resources:

Guidance on how to prepare these queries are provided in OWASP's article [71].

4.2	Database Access: The application must use an account with the least possible privilege to access the database.	- NIST 800-82 r2: AC-6 - ISO 27002:2013: 9.1.2 - ISA-62443-2-1: 9.1.2 - ISA-62443-4-1: SD-6 - ISA-62443-4-2: 3.4 - OWASP Secure Coding Practices: DB security

This can be done when configuring the SCADA system, by adding an account with the least possible privilege, following the recommendations on minimum privilege principle on 9.1.

4.3	Database Connections: All database connections must be terminated as quickly as possible.	- OWASP Secure Coding Practices: DB security

This requirement can only be implemented by altering the source code of the SCADA systems.

Additional resources:

The article by OWASP explains recommends this action to prevent sniffing [71].

4.4	Default Accounts: All database default accounts must be removed or have their passwords changed.	- OWASP Secure Coding Practices: DB security

During the configuration of the client's system, all default database accounts should be disabled. If the accounts are necessary, their password should be changed to a secure one.

4.5	Minimization of Controls: All unnecessary database functionalities must be disabled.	- OWASP Secure Coding Practices: DB security

This is a standard procedure that that is implemented during the configuration of the client's SCADA system.

Additional resources:

According to OWASP, all unnecessary database functionality should be turned off (e.g., unnecessary stored procedures or services, utility packages, install only the minimum set of features and options required) [71].

4.2.5 Patch Management

5.1	All EMS software (base software and firmware) should be continuously updated to the latest versions in terms of security, with security patches if needed	- CIP-007-6: R2 - NIST 800-82 r2: 6.2.17/ SI-2/ SI-3 - ISO 27002: 2013: 12.6.2/ 14.2.4 - ISA-62443-2-1: 12.6.2/ 14.2.4 - ISA-62443-4-1: Practice 7

Proposed solution:

Before updating any of the software, a change management policy should be defined as described on the NERC-CIP-010-2 standard [44], table R1. More specifically, this should include a security update management policy as described on ISA-62443-4-1: Practice 7 [26]. After having a defined policy, software updates and security updates should be tested, as required on 5.7 and 5.8, delivered in a timely manner (depending on the urge), and properly registered and documented, as required on 5.6.

Additional resources:

Additional guidelines (apart from the ones already referenced) on how to implement this strategy are described on the standards associated.

When implementing a change management policy, we recommend following the Information Technology Infrastructure Library (ITIL) framework's change management process. According to [72], ITIL's change management process consists of the following steps:

- **Request for change**;
- **Accept and Classify Change Requests**;
- **Assess Impact of Changes**;
- **Aprove and Schedule Changes**;
- **Coordinate Change Implementations** or **Distribute and Install Changes**;
- **Review and Close Change Requests**;
- **Monitor and Report Change Management**.

In this work, a best practice approach for automating IT management processes is also presented.

5.2	The vendor / integrator should provide a proactive and predictive technical support in security add-ons and updates	–

Proposed solution:

By following the procedure described on 5.1, this should be accomplished.

5.3	Operating System: It must be guaranteed that the OS remains updated.	- ISO 27002:2013: 12.5 - ISA-62443-2-1: 12.5

Proposed solution:

It is important for the OS to be up to date, due to security issues. A large number of OS updates are security patches (vulnerabilities that were unearthed/exploited), if these are not implemented as soon as possible, the system may be exploited. The OS should be update the same way other software are updated, following the procedures described on 5.1. When OS updates are applied to the client's "real" system, the redundancy feature should be used, when available, to assure the system's availability.

For devices that are running Microsoft Windows, this issue can be resolved using the Windows Server Update Services (WSUS) tool. WSUS allows IT administrators to deploy the latest updates of Microsoft products on the network. It can be used to fully manage the distribution of updates released through Microsoft Update.

Additional resources:

Guidance on how to implement WSUS can be found on Microsoft's official doc page [73].

5.4	Antimalware Updates: It must be guaranteed prior to deployment that malware software updates or patches do not contain incompatibilities with the application	- CIP-007-6: R2 - NIST 800-82 r2: 6.2.17/ SI-3 - ISO 27002:2013: 12.2 - ISA-62443-2-1: 12.2

Proposed solution:

Antimalware updates should be treated as the other software updates that are described on 5.1.

5.5	Pre-production Environment: A pre-production environment must be made available for testing and validating new versions. This system must remain identical to the production one	–

Proposed solution:

Although this isn't a specific standard requirement, this is can easily be implemented by SCADA developers. It can be achieved by copying the client's SCADA system (before alterations) to a test environment using virtual machines.

The only issue when maintaining a pre-production environment that is identical to the production one is time synchronization. To resolve this issue we recommend using a time sync tool such as NetTime[5].

[5]NetTime, Network Time Synchronization Tool. http://www.timesynctool.com/

No.	Requisite	Related Standards
5.6	Change Records: It must be guaranteed that every change made on infrastructures, OS or software is properly registered and documented.	- CIP-010-2: R1 - NIST 800-82 r2: 6.2.5 - ISO 27002:2013: 12.1.2/ 14.2.2 - ISA-62443-2-1: 12.1.2/ 14.2.2

Proposed solution:

It is considered a standard procedure to provide documented information about infrastructure and system changes (change tests, impacts of updates/patches, outdated software, fallback configuration). In other words it's the documented version of the requirements that this section contains.

Additional resources:

Detailed instructions of what should be documented and how changes should be managed is provided on the ISA 12443-2-1 standard [26], as well as the first requirement (R1) on the NERC-CIP-010-2 [44].

5.7	Change Testing: It must be guaranteed that every change made on OS or software is properly planned and tested	- CIP-010-2: R1 - NIST 800-82 r2: 6.2.5 - ISO 27002:2013: 12.1.2 - ISA-62443-2-1: 12.1.2

Changes are always tested by SCADA developers on a test environment (like on 5.5) before they are implemented on the real system. In case of emergency they are immediately implemented following the test.

5.8	Change Impact: It must be guaranteed that an analysis is undertaken to assess impact on security for every system change.	- CIP-010-2: R1 - NIST 800-82 r2: 6.2.5 - ISO 27002:2013: 12.1.2 - ISA-62443-2-1: 12.1.2

When testing changes, their impact is also tested.

5.9	Change Management: It must be guaranteed that the change management system individually identifies all software and hardware that requires alteration.	- ISO 27002:2013: 14.2.2 - ISA-62443-2-1: 12.1.2/ 14.2.2

Proposed solution:

In order to implement this requirement, the same procedure should be followed as described on 5.1.

5.10	Incorrect Changes: A formal procedure must be established to interrupt ongoing incorrect changes and, if necessary, recover a previous state (fallback)	- ISO 27002:2013: 12.1.2 - ISA-62443-2-1: 12.1.2

Generally, there are only fallback mechanisms in the system for changes made to the configuration or software installation.

Additional information is required from the client, as there is a confliction of requirements, as there are both a request to only have defined procedures on 9.2, and a request to fallback from procedures.

4.2.6 Monitoring and Logging

No.	Requisite	Related Standards
6.1	Security Log: The system must have a data record of all security events (LOG_02) for auditing purposes (security log)	- CIP-007-6: R4 - NIST 800-82 r2: 5.16 - ISO 27002: 2013: 12.4.1/ 12.4.3 - ISA-62443-2-1: 12.4.1/ 12.4.3 - NIST 800-92 - OWASP Secure Coding Practices: Logging

Proposed solution:

By using a Security Information and Event Management (SIEM) software, all security events that are generated in the system can be centralized into one general security log. Allowing audition of the whole system in one centralized place. The SIEM software should gather all the events that were recorded by the devices on 6.2.

For proprietary SIEM solutions, we propose the use of SIEMs such as ArcSight[6] or Splunk[7], as they are leaders in commercial SIEM solutions.

In terms of open source SIEM solutions, there are currently no open source platforms that include all of the core SIEM capabilities in their raw form [74]. Existing solutions generally require combining add-ons or other tools to achieve all of the core capabilities. According to [75] and [76], the core capabilities a SIEM should have are:

- **Log management and analysis** - log collection, log processing, storage/retention and display (dashboards);
- **Event correlation** - Look for common attributes (correlation rules) in events to link them together into meaningful bundles that could be indicative of a breach in security;

[6] ArcSight, Enterprise Security Manager (ESM). https://www.microfocus.com/en-us/products/siem-security-information-event-management/overview

[7] Splunk, Enterprise Security https://www.splunk.com/

- **Alerts** - Alerts when a correlation rule is triggered;
- **Incident management** - Response to incidents identified and alerted.

Although not a complete SIEM solution, we propose using the Elasticsearch, Logstash, and Kibana (ELK) stack[8], now known as the Elastic Stack with the addition of Beats, as it is a very complete and powerful tool for log management and analysis. ELK stack contains various modules that are organized in levels. Each of these models have a specific function. Models are chosen to be added to the stack, depending on what functions are desired when managing and analyzing logs. This tool however is unable to perform the other functions required in a SIEM solution such as event correlation, alerts and incident management in its original state [76]. To partially solve this issue, Elastic Stack can be combined with other open source SIEM solutions such as: Open Source Security Information and Event Management (OSSIM[9]), Open Source HIDS Security (OSSEC[10]), SIEMonster[11] (already uses the Elastic Stack in its solution), or Prelude[12]. These tools underperform in terms of log management, but are excellent solutions for the capabilities that the Elastic Stack lacks, making them versatile and powerful SIEM solutions when used in conjunction. Most of these tools have been tested and perform well on smaller deployments, but have been seen to have performance issues when used on a larger scale [74]. This is a serious issue for SCADA SIEM solutions, as most SCADA systems have large networks.

It all comes down to what the client wants and what the developers are capable on implementing. The open source SIEM solutions mentioned above are versatile and powerful tools, but they require large engineering feats by the company, requiring a great deal of expertise in the area, time to assure proper deployment and financial costs to be implemented. In some cases, it might be cheaper and easier to implement a commercial solution [74].

Additional resources:

Guidance on logging both security and system events are provided on the standards attached to this requirement. Owasp in particular provides best practices for logging in general [65].

An example of ELK stack used in SCADA systems is demonstrated in [40].

[8] Elk stack, The Elastic Stack. https://www.elastic.co/elk-stack

[9] OSSIM, AlienVault OSSIM: The World's Most Widely Used Open Source SIEM. https://www.alienvault.com/products/ossim

[10] OSSEC, Host-based intrusion detection security. https://www.ossec.net/

[11] SIEMonster, Affordable Security Monitoring Software Solution. https://siemonster.com/

[12] Prelude, a Universal "Security Information & Event Management" (SIEM) system. https://www.prelude-siem.org/

6.2	Security Events: The system must record SUCCESSFUL and UNSUCCESSFUL attempts for the following security events: 1. Failed login attempts for user and system accounts 2. Stop, pause or restart of applications and critical processes/services 3. Creation of new accounts 4. Deletion of accounts 5. Changes to assigned roles and privileges 6. Escalation of host privileges 7. Input validation failures 8. Attempts to connect with invalid or expired session tokens 9. System exceptions 10. Administrative functions, including changes to the security configuration settings 11. Backend TLS connection failures 12. Attempts to change a password Security Log Data should include the following: 1. Time stamp from a trusted system component 2. Severity rating for each event 3. Tagging of security relevant events, if they are mixed with other log entries 4. Identity of the account/user that caused the event 5. Source IP address associated with the request 6. Event outcome (success or failure) 7. Description of the event	- CIP-007-6: R4 - NIST 800-82 r2: 5.16 - ISO 27002: 2013: 12.4.1/ 12.4.3 - ISA-62443-2-1: 12.4.1/ 12.4.3 - NIST 800-92 - OWASP Secure Coding Practices: Logging

Proposed solution:

All of the firewalls, IDSs, IPSs, antiviruses, access points, and AD servers on the SCADA system should be configured to generate such events. Additionally, the SCADA systems software should be developed to also generate such events.

All event logs should then be centralized using the SIEM solution presented on 6.1.

6.3	Security Log Access: Access control configurations must block any attempt of modifying the security log	- NIST 800-82 r2: AU-9 - ISO 27002: 2013: 12.4.2 - ISA-62443-2-1: 12.4.2

Proposed solution:

When limiting permissions of each user with RBAC on 2.12, the option to alter security logs should be denied to every user, only granting this privilege to the applications generating the logs. In other words, the log file shouldn't have writing privileges for users.

6.4	Log Differentiation: There must be two separate logs, one for the system and another one for security, with different access privileges	- OWASP Secure Coding Practices: Logging

Proposed solution:

This is also a best practice recommended by OWASP, that recommends separation of logs, since the types of events and details collected will tend to be different [65].

The SIEM solution proposed on 8.1 is capable of dividing the various types of events gathered into specific groups of logs. It only requires configuration to do so.

Access privileges can be set when defining RBAC on 2.12.

6.5	User Actions: Every user activity must be recorded in logs.	- CIP-007-6:R4 - NIST 800-82 r2: 5.16 - ISO 27002: 2013: 12.4.1/ 12.4.3 - ISA-62443-2-1: 12.4.1/ 12.4.3 - OWASP Secure Coding Practices: Logging

Proposed solution:

This log record is very difficult to implement, as the log would be too extensive. But user activities must be included on logs according to the standards attached. For user actions to be logged, the SCADA systems software should be developed to do so. Additional information should be provided by the client for log recording, namely:

- Duration of the logs: how long should logs be kept before being removed (storage space issue);

- What information they would like the log to contain.

Additional resources:

According to NERC-CIP [44], the activities recommended to be logged include:

- successful and unsuccessful authentication;
- account management;
- object access;
- processes started and stopped.

Both ISO and ISA also specify what information should be included on the logs regarding user activities [42] [26].

6.6	Auditing and Logging Role: A dedicated role for auditing and logging must be created	- ISO 27002: 2013: 6.1.1 - ISA-62443-2-1: 6.1.1

Proposed solution:

Both these roles can easily be implemented when creating user/application roles on 2.12. As mentioned on 6.3, only the application(s) that is logging should be able to write on the logs, or in this case, have the logging role. The auditing role should be granted exclusively to users who are permitted to operate the SIEM software.

6.7	Transactions: Every transaction performed on the system on the user interface must be recorded for auditing purposes, with the respective motive for the realization of that specific action	- ISO 27002: 2013: 12.4.1 - ISA-62443-2-1: 12.4.1 - ISA-62443-4-2: CR 6.2 - OWASP Secure Coding Practices: Logging

For the same reasons mentioned on 6.5, this is a challenging task. Additional information should be provided from client to determine if these transactions are only from external sources. 6.5 also mentions how this measure can be implemented.

Additional resources:

OWASP Secure Coding Practices provides some guidance on this subject in particular [65].

6.8	Sensitive Information in Logs: The system must not store sensitive information on its logs, including unnecessary system details, session identifiers or credentials/passwords.	- OWASP Secure Coding Practices: Logging

According to OWASP, intercepting some communications, monitoring employees, and collecting some data without consent may all be illegal. Thus such information should be excluded from logs [65].

Additionally, if consent is given, the logs should be protected.

No.	Requisite	Related Standards
6.9	Monitoring: There must be the possibility of monitoring the total amount of running services	- NIST 800-82 r2: CA-7 - ISO 27002:2013: 12.4 - ISA-62443-2-1: 12.4

Proposed solution:

Nagios[13] can be configured to perform this task. The software can monitor a variety of information, such as services running on each device of the network, active users, requests from external sources, and more. It does however require a custom check plugin to be developed to do so.

Since SCADA networks are quite large and dispersed, making a single device monitor the whole network with Nagios can potentially overload the device. For this reason, we recommend having multiple devices with Nagios spread throughout the network that periodically send the information they gathered to the device containing the main Nagios servers, allowing the load to be distributed, reducing the chance of overload. These smaller Nagios devices are known as Nagios proxy.

6.10	Requests from External Applications: There must be the possibility of monitoring and tracing (with timestamp for every action) all requests made to the external interface coming from external applications	- NIST 800-82 r2: AU-8/ CA-7 - ISO 27002:2013: 12.4 - ISA-62443-2-1: 12.4

This requisite depends on what the client considers to be external applications.
The solution proposed on 6.9 is capable of satisfying this requirement.

6.11	Active Users: There must be the possibility of monitoring the number of users with an active session	–

There is no standard that requires current users to be monitored. However, this it is possible to achieve this using the program mentioned on 6.9.

4.2.7 Backups

[13]Nagios, The Industry Standard In IT Infrastructure Monitoring. https://www.nagios.org/

No.	Requisite	Related Standards
7.1	Disaster Recovery: A disaster recovery plan must be established and properly documented.	- CIP-009-6: R1/R2/R3 - NIST 800-82 r2: 6.2.6.2 (6.2.6) - ISO 27002:2013: 17.1.1 (17.1) - ISA-62443-2-1: 17.1.1(17.1) (C.4.4.5)

According to the standards attached to this requisite, a recovery plan should be projected and documented.

7.2	Backups: Periodical backups must be configured both for system and databases according to the criteria defined by Client A	- CIP-009-6: R1: 1.3/1.4 - NIST 800-82 r2: CP-9 - ISO 27002:2013: 12.3 - ISA-62443-2-1: 12.3 - ISA-62443-4-2: CR 7.3

Most SCADA developers have already implemented redundancy features on their systems. This feature is normally implemented to assure availability, but, since these servers have option to configured on hot standby, they can be used as backups for both system and databases.

4.2.8 Compliance

8.1	Penetration Tests: The system must undergo penetration tests performed by the vendor and/or Client A, allowing detection and correction of potential vulnerabilities and backdoors. The vendor must provide documentation as proof of testing.	- CIP-010-2: R3 - NIST 800-82 r2: CA-2/ CA-8/ RA-5/ SI-6/ PM-6/ PM-14 - ISO 27002:2013: 14.2.8/ 18.2 - ISA-62443-2-1: 14.2.8/ 18.2 (C.4.4.3.5.5) - ISA-62443-4-1: Practice 5

Recently became a standard procedure for SCADA systems due its increased importance.

Additional resources:

Information on the importance and how this procedure should be documented is provided on the standards specified.

No.	Requisite	Related Standards
8.2	Documentation: The vendor must provide documented evidence that all the requirements from this manual have been contemplated and implemented	- ISO 27002:2013: 14.2.7 - ISA-62443-2-1: 14.2.7

Documentation regarding cybersecurity requests compliance should be provided as is done with other system requirements.

4.2.9 General

9.1	Minimum Privilege: The system must implement the minimum privilege principle, restricting access to certain functionalities, data and information to users with adequate permissions.	- CIP-004-6:R4 - NIST 800-82 r2: AC-6 (6.2.1.1) - ISO 27002:2013: 9.1/ 9.2.3/12.6.2 - ISA-62443-2-1: 9.1/ 9.2.3/12.6.2 - ISA-62443-3-1: 5.1 - ISA-62443-4-1: SD-6/ SI-1 - ISA-62443-4-2: 3.4/ CR 1.1/ CR 2.1

Proposed solution:

In the least privilege principal, every program and every user of the system should operate using the least set of privileges necessary to complete the job, reducing the number of potential interactions among privileged programs to the minimum for correct operation, so that unintentional, unwanted, or improper uses of privilege are less likely to occur [77].

To implement the least privilege principal in this system, it would have to be enforced in Role-Based Access Control (RBAC). The principal has to be implemented when configuring RBAC on 2.12.

Additional resources:

Guidance on how to specify and enforce the principal in RBAC is provided in [78]

9.2	Minimization of Controls: All unnecessary services, protocols and applications must be disabled or removed	- ISA-62443-4-2: CR 7.7

During the SCADA's configuration, all unnecessary services and protocols are generally disabled. The only applications that remain installed are applications that are necessary for the systems functionality.

4.2.10 Development

10.1	Input Validation: Input validation must be implemented, as well as output encoding.	- NIST 800-82 r2: SI-10 - ISA-62443-3-3: SR 3.5 - ISA-62443-4-2: CR 3.5 - OWASP Secure Coding Practices: Input Validation/ Output Encoding

Proposed solution:

All input should be validated in order to ensure only properly formed data is entering the workflow in an information system. It helps (doesn't ensure) prevent XXS, SQL injection and other attacks [65].

Output should also be encoded for the same reason.

Additional resources:

OWASP's article has a full section dedicated to this measure, explaining why and how it should be implemented and [65].

No.	Requisite	Related Standards
10.2	Coding Best Practices: It must be guaranteed that software is developed according to the "OWASP Secure Coding Practices" standard to protect it from SQL Injection, Buffer Overflow, XSS, etc.	–

Every developer, not just SCADA, should follow these practices to avoid having vulnerabilities on their software.

Additional resources:

Guidance on the secure coding practices requested can be found on OWASP's official website on both [71] and [65].

No.	Requisite	Related Standards
10.3	Notifications and Alerts: The system must allow configuring alerts from a certain set of rules, sending messages (if considered relevant) to specific users associated with those alarms via e-mail or SMS	- NIST 800-82 r2: 6.2.17/ SI-5 - ISA-62443-2-1: B.11/ C.4.4.5.1

Generally implemented by developers when developing the client's SCADA system.

All notifications and alerts should be fully customized to trigger according to the client's requests. These notifications and alerts should also have additional options to send SMS' and/or e-mails.

4.3 Conclusions

As mentioned before, the evolution of SCADA systems exposed these systems to threats and vulnerabilities they have never been exposed to before. Since these systems are unique, traditional IT solutions can't always provide the adequate protection. Due to this, cybersecurity in SCADA systems has become an important topic not only for standard-setting organizations, but also SCADA developers and SCADA clients. The approach we propose on this chapter is mainly destined for SCADA developers.

That said, in order to perform a complete theoretical cybersecurity analysis of a SCADA system, one should consider both client cybersecurity requisites and standards. Clients will formulate their security requisites based on their knowledge of a standard recommendations and/or requirements. When joining these two topics a SCADA developer can assure the compliance of both the requisite and the standards related to this requisite simultaneously.

Additionally, the solutions we propose on section 4.2 were also obtained considering both client requisites and standards. If the client specified a certain feature or technology that was desired on the respective requisite, we would verify if they were recommended by any of the standards, and would search for resources on how they could be implemented. If there were no specifications, we would follow the recommendations of the standards and propose an adequate solution accompanied by resources.

Chapter 5

Practical Approach to Cybersecurity Analysis

When dealing with SCADA systems, it's important to note that these systems have networks that are typically very well isolated (via firewalls or physically) and segregated (shown on section 2.1), making it hard to breach the system's critical network. Due to this, many SCADA developers only focus on preventing external attacks that would allow an attacker to gain access to one of the system's networks, such as phishing campaigns, physical access (USB sticks), social engineering, and remote attacks on both vulnerable access points (AP) and machines with unprotected internet access that can easily be discovered using tools such as Shodan as shown by [79][80]. By focusing solely on this problem, SCADA developers typically forget or don't have enough time to protect the network internally when developing their SCADA systems, neglecting software security updates, passing unencrypted data between machines, avoiding Intrusion Detecting Systems (IDSs), and other bad practices. There are also cases of SCADA users that have systems that were developed decades ago, before cybersecurity was even an issue. Most of these users simply request their SCADA developers to patch the system's networks with additional segregation and firewalls as a quick solution.

In this chapter we propose a practical approach to perform a cybersecurity analysis of a SCADA system's internal network. We start by presenting a methodology to establish a threat model to help readers and SCADA developers identify common entry points, desirable assets and possible attack vectors that could allow access to such assets within their SCADA system's critical network. We then present a penetration testing methodology that will help validate the attack vectors of the threat model. When using this penetration testing methodology on the ScateX# prototype, we were able to not only validate the attack vectors, but also evaluate some standard and client requisite compliances and provide evidence of other noncompliances. For confidential reasons, we will not be describing or displaying the latter results on this document.

5.1 Threat Model

The goal of this threat modelling architecture is to identify possible attack vectors that can cause problems to the system internally, or, in other words, what can an attacker do after breaching the SCADA network or a machine of on this network.

In this methodology, we will be assume that the attacker has already compromised the SCADA network or a device on the network using the methods that where described above. The most common entry points for an attacker to access the SCADA network would be:

- **HMIs/Workstations** - Devices with the most human interaction;
- **APs** - If not configured correctly, they can have default usernames and passwords, allowing internal access to attackers;
- **Devices with VPN access** - Credentials of system operators or maintenance personnel can be stolen;
- **RTUs/PLCs** - If there are any on the SCADA network, they can be breached. If misconfigured, they could have the default manufacturers credentials and open ports that are externally accessible via internet.

Other entry points could be physical accesses to any of the other servers, but these are generally well guarded and not likely to be breached.

In the case of a network breach via remote access point, an attacker will most likely:

1. Perform scans on the network - Map the network, with the objective of identifying critical devices that contain the most information or that would cause the most damage when attacked, such as HMI's, SCADA's, Engineering Workstations, DMS's and Historians (FEP's, RTU's and PLC's can also be present on this network, but it's unlikely) ;
2. Perform port scans on said devices - To discover information about the devices' OS, what services are running on open ports, and the version of these services;
3. Attack the critical devices to:
   (a) Gain access to the device via:
      i. Exploitation of vulnerable services discovered during port scan using attack libraries (CVE lists, exploits databases, etc.);
      ii. Brute force of services that require credentials;
   (b) Sniff packet information (Passive MITM);
   (c) Alter packet information (Active MITM);
   (d) Deny communication between the servers (DoS).
   (e) Reduce system performance, by flooding or crashing a machine using:
      i. Exploitation of vulnerable services discovered during port scan using attack libraries (DoS);
      ii. ICMP vulnerability (Ping of Death);

In the case of a network breach via one of the other mention devices, an attacker attempt to pivot their connection and proceed to follow the same steps as above. Pivoting is a term that refers to using a compromised system as a "relay" for attacking further into the network [11] .

5.2 Penetration Testing Methodology

Before discussing this topic, it is important to highlight that penetration testing and vulnerability assessment aren't the same thing. These two tasks are commonly seen as the same procedure by the standards referenced in this document, but they generally have different goals.

Unlike the vulnerability assessment preformed in the theoretical approach on chapter 4, where we analyzed the security posture of the SCADA system using common ICS security standards and client requisites, the goal of penetration testing isn't to discover security vulnerabilities (in the majority of cases). Penetration tests are instead authorized simulated attacks, whose purpose is to exploit a system using an attackers point of view, mimicking his strategies, methods and techniques in order to:

- Validate vulnerabilities found during vulnerability assessment;
- Verify successful/unsuccessful implementation of security controls;
- As part of a risk assessment program, in which probabilistic data should be provided on whether a vulnerability is valid, if it can be exploited, and to what degree of difficulty.

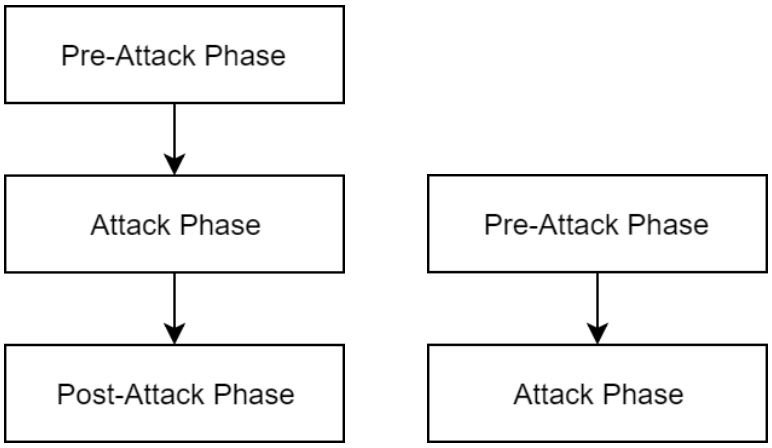
Penetration testing will generally consist of three phases as shown on Figure 5.1a. These phases mimic the phases that an actual attacker would use when conducting a real attack. The phases are [10]:

- Pre-attack phase: Reconnaissance, or preparation of an attack;
- Attack phase: Execution of the actual attack;
- Post-attack phase: Returning target to original state.

In the case of an actual attacker, there is no Post-attack phase. With the exception of covering their tracks, attackers don't bother returning the target to its original state. For and attacker, the longer the target is compromised, the better, as it's an indication of a successful attack. In this work we won't place much emphasis on the Post-attack Phase, as in this phase we simply returned the targets to their original state, before the attack were executed.

When performing penetration tests on a system, the targets to be tested can be one of the following types:

- Black Box: tester has minimal to no prior knowledge of the target's devices, applications, systems or company;
- White Box: tester has full knowledge of the the target's environment and systems, including:
  - company information;
  - IP addresses;



(a) Penetration tester (b) Attacker

Figure 5.1: Cyber attack phases [10]

  - devices;
  - applications;
  - source code.
- Grey Box: middle-ground between black box and white box, tester has some knowledge of the target.

Before testing a system, it's important to know that there are essentially two types of testing methods:

- External testing: Refers to performing tests on Internet-facing environments/networks from the Internet itself. In other words these tests are performed to find ways to enter the ICS network;
- Internal testing: These tests are performed considering external attacks where successful, and the attacker has access to the ICS network. The objective of these tests is to find ways to take control of targets, sniff information, alter information, deny services, or even shutdown a target.

When performing penetration tests on active ICS systems, there are multiple issues to be considered. One cannot approach ICS penetration testing in the same manner as IT systems. As mentioned before, ICS systems such as SCADA monitor and control critical infrastructures. Due to this, these systems are required to be high-availability systems that maintain an uptime of 99.999%. This requirement limits penetration testing in a substantial way when testing active ICS systems. Since many active systems where developed decades ago, there are various limitations in terms of the technology used on these systems, as most of it is outdated. A simple active scan on a server of these systems is able to crash the server. Due to this, we recommend performing penetration tests on ICS systems that are setup in a lab environment. In this work, the issues above weren't taken in to account, as the environment that was provided to perform the tests was a development lab, with servers configured on virtual machines.

In this chapter we will be performing penetration tests on grey boxes, as information about the SCADA system was either previously provided by the company, discovered during the vulnerability assessment or reveled during the penetration tests. We will assume the scenario that an attacker has already found a way to gain access to the SCADA network, and therefore will not perform external tests on the system, only internal tests.

The methodology we followed and propose when performing the penetration tests is the following:

- Reconnaissance: host discovery, port scanning, service enumeration;
- Vulnerability mapping: matching discovered services and versions with known vulnerabilities (that are exploitable);
- Exploit known vulnerabilities.

To perform these tasks, we used a virtual machine that that was running the Kali linux operating system [81].

5.3 Reconnaissance

Reconnaissance, or footprinting, is the act of gathering useful information about a targeted system or network and the devices that this network contains [82] [83]. This process is part of the pre-attack phase shown on Figure 5.1 and can be sub-divided into two types:

- Passive reconnaissance - Information about the targeted system is gathered without "touching" the network in order to avoid detection. This type of reconnaissance involves finding the maximum amount of useful information about the targeted system/company on the Internet, by researching company information and discovering IP address blocks and URLs associated with the target, using methods/tools such as [11]:
  - Discover scripts (passive scripts);
  - Google hacking databases;
  - WHOIS queries;
  - Maltego;
  - Shodan.
- Active reconnaissance - Information is gathered by actually doing something on the targeted network, increasing the chance of detection, but obtaining the most valuable information. This type of reconnaissance is usually done once all available sources for passive reconnaissance have been exhausted. As mentioned on Chapter 5.1, this task can be performed on both types of network compromises, and is generally done using the following methods:
  1. Passive scanning, or packet sniffing(only on internal testing):
     - Interface traffic (if on compromised machine);
     - Broadcast packets (if on compromised network).

2. Active scanning:
   - Network mapping and host discovery;
   - Port scanning, service enumeration, application discovery, and OS discovery.

In summary, reconnaissance is the gathering of information in all of its many forms. Attackers, and consequently penetration testers, spend more time on reconnaissance then on actual attacks. For an attacker, the more information is gathered about the compromised network, the better, as they can precisely identify their target, perform precise vulnerability maps and, consequently, craft attacks that are more sophisticated and complex as mentioned in Section 2.3.2. In the penetration tester's point of view this phase is important because it simulates both how the attacker sees the network and what information can be extracted from each device on the network. A penetration tester will be interested in finding and patching as many vulnerabilities as possible due to not knowing which vulnerability the attacker will exploit.

Both types on reconnaissance can be used in both external and internal testing. External testers tend to favor passive reconnaissance as this is how an attacker will discover network entry points. Internal testers on the other hand prefer active reconnaissance due to them assuming the attacker already has access to the network and will attempt to sniff traffic for information or scan for vulnerabilities.

In this work, the tools that we used to perform the internal reconnaissance were NMAP (Active Scan) and Ettercap (Passive Scan). Additionally, vulnerability scanners such as OpenVAS and Nessus can be used to perform active scans on devices. They are capable of performing tasks such as network port scanning, service fingerprinting, vulnerability probing, authenticated scanning and custom audit checks. The advantage of these tools over powerful port scanners such as NMAP is the authenticated scanning feature, that is capable of detecting vulnerable applications running on the device that don't require ports, such as compilers.

5.3.1 Host Discovery

Although NMAP can determine what hosts are available on the network, it was not used in this work to do so. The reason being is that the hosts relevant to the penetration tests were located on a development lab environment. Using NMAP to scan the network could reduce the performance or even crash hosts that were being worked on. The hosts to be tested (shown on Table 5.1) on the network were therefore provided by the company. Once the important hosts are known and mapped, we can proceed to the second phase of reconnaissance, port scanning.

5.3.2 Port Scan

As mentioned above, port scanning is a form of active scanning. In an attacker's point of view, this means that the risk of being detected by an eventual IDS is high. The goal of a port scan is to identify which ports are open on the targeted device, to enumerate the services that are running on each of these ports, to discover the applications providing the service, the application's version

Table 5.1: Important hosts on the network

Server	IP Adress
Front-End	172.18.200.120
RTU	172.18.200.77
DMS	172.18.214.6
SCADA	172.18.214.7
Historian	172.18.214.9
Front-End	172.18.214.14
HMI	172.18.214.15
RTU	172.18.214.28
SCADA	172.18.214.47

and what operating system (OS) the device is running. It basically displays all the information an attacker desires.

Like we mentioned before, there are special considerations to be taken into account when performing port scans on ICS devices that are "real" physical devices. Some ICS security specialists, such as Justin Searle who is the Director of ICS Security at InGuardians[1] and a senior instructor for the SANS Institute, recommend not performing port scans at all on legacy embedded systems, as they can crash or decrease the performance of the system, reducing the system's availability and reliability. If port scans are absolutely necessary to evaluate the cybersecurity posture of a legacy system, Searle [84] recommends performing low risk port scans that avoid the causes that are most likely to crash these systems. According to Searle, the most likely causes are:

- **OS fingerprinting** - Penetration testers should avoid using NMAP's -O and/or -A flags when scanning embedded systems, as it is by far the most likely cause of crashes in these systems.
- **Scanning with SYN scans** - NMAP uses this scan by default, as it is the most used and popular scan on the tool due to it being quicker than the other scan types and less likely to be blocked by firewalls. This scan type however isn't a proper Request for Comments (RFC) behavior, allowing only mature TCP/IP stacks to handle this scan properly. To avoid this issue, ICS penetration testers are recommended to always specify TCP scanning (-sT) in their scans.
- **Scanning too fast** - NMAP's default scan speed is too fast for legacy systems. For this reason, it is recommended to slow down the scan to use less bandwidth and less target machine resources by setting the interval to 0.4 seconds (-T2), setting a delay between probes (–scan-delay 0.1), or setting NMAP to scan one port at a time per host (–max-parallelism 1).
- **Scanning UDP ports with null payloads** - Scanning UDP ports with null payloads can interrupt these communications due to their nature. This issue can also affect ICS software

[1] InGuardians, an independent information security consulting company that specializes in RedTeam penetration testing, hardware & application security assessments, threat hunting and incident response. https://www.inguardians.com/

on both Windows and Linux. For this reason it is generally recommended not to perform UDP scans (-sU) on legacy systems.

- **Service fingerprinting** - This scan is usually safe to perform, but it can occasionally cause problems. Penetration testers are thereby recommended to use NMAP's fingerprinting service (-sV) selectively on new or unknown subnets. Additionally, NMAP's script the scans service banners can be used for this purpose (–script=banner).

Although these port scan recommendations are primarily directed to penetration testers that are testing "real" legacy embedded systems that are used in operational ICSs, they also apply to tests performed on virtual machines of ICS devices that are running on older hardware and tests performed on practically all devices (even modern ones) being used on ICSs that are operational. In other words, these recommendations should be followed when testing any device on an operational ICS, as any crash or reduction in performance of these devices can affect the system's availability.

Since we are on a virtual environment used for testing, we only partially considered these recommendations, as any problem caused by our scans could easily be fixed with reboots or reinstallations of the affected devices. For this reason, our main goal in these scans is to see how much information each device displays about its ports, operating system, services and versions of these services. In order to do this, we included UDP port scans and operating system fingerprinting. When scanning, we need to specify what we want the scanner to discover. For our work, we are interested in discovering open TCP (-sT) or UDP (-sU) ports, the version of the services running on the port (-sV), and information about the OS running on the device (-O). For additional information on each device, we included the -A flag in our scans which enables OS detection, version detection, script scanning, and traceroute. In order to maintain the performance of the devices (since development tests were also being performed on them), we followed some of the safety precautions we mentioned above when scanning the machines, namely avoiding SYN scans, selective service fingerprinting and increasing the interval between requests that the scanner has. The result of each scan is similar to the scan we present bellow. We will be omitting specific services and versions that are confidential. After obtaining the scanning information we desired, we can proceed to the next part of our methodology: Vulnerability mapping.

```
msf > nmap -sT -sV -A -T2 172.18.214.7

PORT STATE SERVICE VERSION
21/tcp open ftp -------
22/tcp open ssh -------

123/udp open ntp -------

OS details: --------------

TRACEROUTE
```

```
HOP RTT ADDRESS
1 2.46 ms ------- (172.18.214.7)
```

5.4 Vulnerability mapping

After acquiring the information that was provided by the port scans, we recommend performing a vulnerability map of the information. Vulnerability mapping is simply the process of manually matching the service, operating system, and service versions with databases containing known vulnerabilities such as:

- National Vulnerability Database
- Common Vulnerabilities and Exposures Database
- ICS-CERT Advisories
- Security Focus
- Exploit Database

Vulnerability mapping is also able to help penetration testers perform a vulnerability assessment of a targeted device, by identifying vulnerable applications the device is running. It also assists in providing corrective measures. The databases mentioned above not only archive known vulnerabilities, but also the solution to patch the vulnerability.

For the sake of this work we will only be interested in vulnerabilities that have already know exploits (specifically metasploit modules) associated to them. We will not be displaying the results of our searches in this work for confidential reasons.

5.5 Attacks

This section contains the attacks we were successfully able to execute.

5.5.1 MITM

Man-in-the-middle (MITM) is a type of attack where a malicious user secretly takes control of the communication channel between two or more hosts. Once the communication channel is controlled, the MITM attacker can intercept, modify, change, or replace the targeted victim's traffic while it's still in transit without being detected by the victims [85].

The attack not only targets the actual data that flows between the victims, but is also able to compromise:

- Confidentiality, by eavesdropping on the communication;
- Integrity, by modifying messages;
- Availability, by destroying messages or causing the communication to end.

There are various types of MITM attacks. Conti et al. [85] categorize them based on their impersonation techniques in order to fully focus on the attack itself, obtaining the following division:

- **Spoofing-based MITM** - As the name indicates, in this attack, the attacker intercepts a legitimate communication channel between two hosts via spoofing attack, controlling the transferred data without the knowledge of the hosts. The types of spoof attacks included in this technique are ARP, DHCP, DNS and IP.
- **SSL/TLS MITM** - Also known as SSL stripping, this attack is a form of active network interception, where the attacker inserts itself into the communication channel between two victims (usually a browser and a web server). The attacker then establishes two separate SSL/TLS connections with each victim.
- **BGP MITM** - This attack is based on IP hijacking, but in this case, the attacker delivers the stolen traffic to its original destination. The traffic is make to go through the attacker's Autonomous Station (AS), where it can be manipulated.
- **False Base Station bases (FBS-based) MITM** - In this attack, a third party forces the victim to create a connection with a fake Base Transceiver Station (BTS), using it to manipulate the victim's traffic.

SCADA systems are extremely susceptible to this kind of attack. The cause of this are the security issues that are present in SCADA-specific protocols. The IEC 104, DNP3 and MODBUS protocols are known for their lack of authentication, confidentiality, and integrity verification, making it easy to manipulate their packets [40] [2].

Since we're using a lab environment that was designed for production tests, both the hosts that communicated with each other using SCADA-specific communication protocols where in the same LAN, making ARP spoofing the obvious choice for the MITM attack [85]. The Address Resolution Protocol (ARP) is primarily used in LAN networks for the resolution of IP addresses (network layer addresses) into Medium Access Control (MAC) addresses (data link layer addresses). In order to obtain the MAC of a destination host, a source host will broadcast an ARP request to all the hosts in the LAN requesting the MAC address of the destination host that has a certain IP address. The destination host will then respond to this request with the given IP and MAC using an ARP reply. Once the source host receives this reply it will cache the IP and MAC pairing in its local ARP cache table for future communication, avoiding the need to broadcast the same request in the near future. The ARP spoofing attack, or ARP poisoning, consists of modifying the IP and MAC pairing in the ARP cache table or, in other words, poisoning the cache table. An attacker will use this technique to associate a malicious host's MAC with the IP of the destination host, allowing the attacker to eavesdrop, perform MITM attacks or launch DoS attacks [19]. The attacker will attempt this attack in both directions of a communication channel in order to completely manipulate the communication. Figure 5.2 shows what a successful ARP spoof looks like on master/slave communications. Once two targets have been ARP spoofed, the attacker can visualize all of the packets being transferred by and to them.

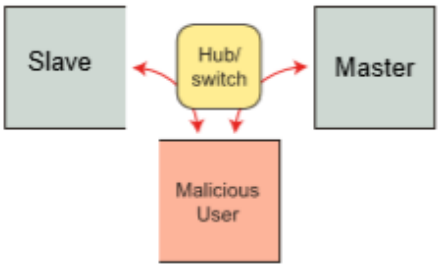


Figure 5.2: Successful ARP Spoofing [2]

In this work, ettercap is the tool used to perform MITM attacks. The tool was chosen due to it having an ARP spoof module and its capacity to dissect any protocol with the adequate plugin.

In order to dissect and perform MITM attacks on SCADA-specific protocols, plugins for ettercap had to be developed or obtained externally, since it didn't contain any plugin for this use in its original toolbox. After extensive research, the only ettercap plugin that was found that dissects a SCADA-specific protocol and/or performs a MITM attack on the protocol's communication channel, was the custom plugin developed by PMaynard that alters IEC 60870-5-104 TCP/IP packets on Github [86]. The plugin however, was only developed to work on the IEC 104 protocol over TCP/IP. To the extent of our knowledge, there were no ettercap plugins that could dissect and perform MITM attacks on the DNP3 protocol (TCP and UDP).

Due to this, we developed two ettercap plugins in C that are capable of conducting MITM attacks on IEC 104 and DNP3 communication channels over TCP, (or UDP for DNP3) by altering their packets in transit. The plugins are available for use on github [87]. These plugins were based on the custom plugin developed by PMaynard mentioned above. Both the 104 and the DNP3 plugins do the same thing in their respective protocol. They were developed to continuously monitor a communication channel for specific 104 or DNP3 packets. Once a packet that contains the plugin's specifications is detected, it is dropped, cloned and altered to what the attacker desires. The only part of the packet that is modified is the data of the SCADA-specific protocol, maintaining the TCP/UDP information of the original packet to try to avoid detection.

The following subsections provide insight on the procedures followed to create the plugins and show the results obtained from performing the MITM attacks on their respective SCADA-specific protocol. By following the procedures that we show, one can alter the plugins we provide to work on any type of DNP3 or IEC 104 packet.

5.5.1.1 Passive MITM

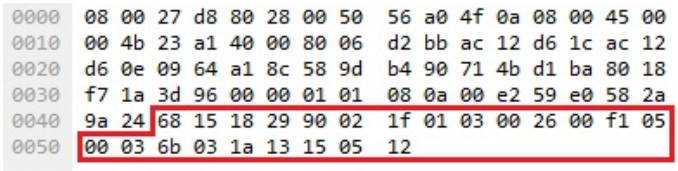
In order to execute the MITM attack on the DNP3 or IEC 104 protocols, we must first identify which devices are communicating with each other using these protocols, and which of these protocols is being used. As mentioned on Section 2.2, these SCADA-specific protocols are essentially used in RTU communication channels. This includes communications between RTUs and SCADA servers, and/or RTUs and FEPs. Although the reconnaissance we performed before

allowed us to identify these devices, we still need to identify which devices are communicating with each other using one of these protocols. To do this, we use ettercap to ARP spoof two of the devices, allowing us to eavesdrop (passive MITM) on all of their communications. During this eavesdrop, we use wireshark to dissect the packets being transferred between the two victims, and where able to identify:

- The devices that were communicating using DNP3 or IEC 104 (RTU-FEP)
- The request made by each packet
- The data being transmitted on each packet

Once we are able to identify this information, we can also use wireshark to view the exact data the protocol is sending as can be seen on Figure 5.3 and 5.4. By using this information provided about the data, we can proceed to the actual MITM attack.

(a) Bit representation



(b) Byte representation

Figure 5.3: IEC 60870-5-104 packet data

5.5.1.2 Active MITM

The ettercap plugins that we developed use the data information extracted from wireshark as a basis. Once activated, they proceed to clone the original packet and break it down into the various layers of the internet protocol suit. Using the information of wireshark, we can make the plugins break down the application layer that contains the DNP3 or IEC 104 protocol data into maneuverable bits and bytes that represent the data as it is shown on wireshark. Once this is done,

```
Application Layer: (FIR, FIN, Sequence 7, Response)
  Application Control: 0xc7, First, Final(FIR, FIN, Sequence 7)
  Function Code: Response (0x81)
  Internal Indications: 0x0000
  RESPONSE Data Objects
    Object(s): Binary Input With Status (Obj:01, Var:02) (0x0102), 1 point
    Object(s): 32-Bit Analog Input (Obj:30, Var:01) (0x1e01), 1 point
      Qualifier Field, Prefix: None, Range: 8-bit Start and Stop Indices
      [Number of Items: 1]
      Point Number 0 (Quality: Online), Value: 30
        [Point Index: 0]
        Quality: Online
        Value (32 bit): 30
```

(a) Bit representation

```
08 00 27 d8 80 28 00 15  f2 52 09 d6 08 00 45 00
00 57 d9 01 40 00 80 06  38 b4 ac 12 c8 4d ac 12
c8 78 4e 20 9d c0 51 1a  da f0 19 2d 85 2b 80 18
fc 69 41 37 00 00 01 01  08 0a 00 0a c6 6d 16 dc
4f f1 05 64 1a 44 01 00  e8 03 85 5f cf c7 81 00
00 01 02 00 00 00 01 1e  01 00 00 00 ed f7 01 1e
00 00 00 73 9a
```

(b) Byte representation

Figure 5.4: DNP3 packet data

we can target specific packets, and alter any data value by simple altering the bits associated to that value. The plugins then change this value in the cloned packet and send it to the original destination. This can be seen in Figure 5.5 and 5.6 for IEC 104. Figure 5.7 and 5.8 show the alteration of DNP3 packets over UDP, but wireshark isn't able to display detailed information in UDP as it does in TCP.

```
Frame 7627: 89 bytes on wire (712 bits), 89 bytes captured (712 bits) on interface 0
Ethernet II, Src: Vmware_a0:4f:0a (00:50:56:a0:4f:0a), Dst: PcsCompu_d8:80:28 (08:00:27:d8:80:28)
Internet Protocol Version 4, Src: 172.18.214.28, Dst: 172.18.214.14
Transmission Control Protocol, Src Port: 2404, Dst Port: 41356, Seq: 1, Ack: 1, Len: 23
IEC 60870-5-104-Apci: -> I (5260,328)
IEC 60870-5-104-Asdu: ASDU=38 M_DP_TB_1 Spont   IOA=1521 'double-point information with time tag CP56Time2a'
  TypeId: M_DP_TB_1 (31)
  0... .... = SQ: False
  .000 0001 = NumIx: 1
  ..00 0011 = CauseTx: Spont (3)
  .0.. .... = Negative: False
  0... .... = Test: False
  OA: 0
  Addr: 38
  IOA: 1521
    IOA: 1521
    DIQ: 0x03
      .... ..11 = DPI: Indeterminate (3)
      ...0 .... = BL: Not blocked
      ..0. .... = SB: Not Substituted
      .0.. .... = NT: Topical
      0... .... = IV: Valid
    CP56Time: May 21, 2018 19:26:00.875000000 Hora de Verão de GMT
```

Figure 5.5: Original packet's ASDU with DPI set to 'Indeterminate'

```
> Frame 6: 77 bytes on wire (616 bits), 77 bytes captured (616 bits)
> Ethernet II, Src: PcsCompu_d8:80:28 (08:00:27:d8:80:28), Dst: Vmware_a0:62:54 (00:50:56:a0:62:54)
> Internet Protocol Version 4, Src: 172.18.214.28, Dst: 172.18.214.14
> Transmission Control Protocol, Src Port: 2404, Dst Port: 41356, Seq: 24, Ack: 7, Len: 23
> IEC 60870-5-104-Apci: -> I (5260,328)
v IEC 60870-5-104-Asdu: ASDU=38 M_DP_TB_1 Spont   IOA=1521 'double-point information with time tag CP56Time2a'
    TypeId: M_DP_TB_1 (31)
    0... .... = SQ: False
    .000 0001 = NumIx: 1
    ..00 0011 = CauseTx: Spont (3)
    .0.. .... = Negative: False
    0... .... = Test: False
    OA: 0
    Addr: 38
  v IOA: 1521
      IOA: 1521
    v DIQ: 0x01
        .... ..01 = DPI: OFF (1)
        ...0 .... = BL: Not blocked
        ..0. .... = SB: Not Substituted
        .0.. .... = NT: Topical
        0... .... = IV: Valid
    > CP56Time: May 21, 2018 19:26:00.875000000 Hora de Verão de GMT
```

Figure 5.6: Altered packet's ASDI with DPI set to 'OFF'

```
c8 78 18 9e 17 72 00 2b  8e 78 05 64 1a 44 01 00
e8 03 85 5f c2 cc 81 00  00 01 02 00 00 00 01 1e
01 00 00 00 d9 22 01 14  00 00 00 f5 a7
```

Figure 5.7: Original DNP3 Response (0x81) packet with Value set to '20' (0x14)

```
c8 78 18 9e 17 72 00 2b  8e 64 05 64 1a 44 01 00
e8 03 85 5f c2 cc 81 00  00 01 02 00 00 00 01 1e
01 00 00 00 d9 22 01 28  00 00 00 f5 a7
```

Figure 5.8: Altered DNP3 Response (0x81) packet with Value set to '40' (0x28)

5.5.2 Device Access

Accessing a device remotely is one the main goals of an attacker when a SCADA system is breached. This is due to the importance that each device has in controlling the actual system. If an attacker is able to obtain remote access to a HMI, he can remotely control the system and cause large amounts of damage.

After vulnerability mapping all the services that were running on open ports, we filtered our findings to only include metasploit remote attack modules. These modules exploit failures in the vulnerable versions of the application that can allow an attacker to remotely execute arbitrary code, exploit buffer overflows, upload and execute arbitrary files, and others. Before executing one of these attacks, a payload is chosen. These payloads contain scripts that allow an attack to establish a connection to the targeted device, effectively granting the attacker remote access when executed.

Knowing this, we used the metaploit framework (msf) and proceed to execute these models found in out mapping. Most of the attacks failed to grant access to the target devices, as some devices were patched accordingly. We were however able to successfully access some of the devices remotely, allowing us to access the devices terminal.

Due to confidential reasons, we will not be discussing or displaying which modules were used, nor what servers were accessed in this paper. The payloads utilized in our remote access attacks were all meterpreter payloads. Meterpreter is a dynamically extensible payload that uses in-memory DLL injection stagers and is extended over the network at runtime. It communicates over the stager socket and provides a comprehensive client-side Ruby API [88].

5.6 Conclusions

The practical approach to cybersecurity analysis we propose in this chapter is able to educate SCADA developers and clients, and readers on what a potential attacker is able to do when he has obtained access to the SCADA network. It does this mainly through the threat model that we introduce. The model identifies the most common and probable entry points an attacker uses to access the SCADA network, targets an attacker will set (assets) and attacks an attacker will perform on said target.

The penetration testing methodology helps us test the network in terms of information gathering, vulnerability discovery, and exploitation of these vulnerabilities. Allowing us to see exactly what information the network can provide to an attacker, and what attacks the attacker is likely to execute. As we discuss on the penetration testing section, the most important phase of an attack-/penetration test is reconnaissance. All attacks derive from information that was gathered prior to the attack. Since external reconnaissance is out of the scope of this work, we will discuss internal reconnaissance solutions. It's impossible to protect a device and network from all types of scanning tools, lots of them use scanning methods that explore essential network protocols that cannot be disabled. That said, possible solutions we recommend to attempt to mitigate the chance

of attacks is to implement firewalls with strict rules to filter the ports that are open, and to disable banner information of applications that require open ports.

For attack prevention solutions, we recommend encrypting all communication channels that use SCADA-specific protocols to operate, this will ensure protection against all types of MITM attacks, and not just ARP spoofing. Although antique these protocols are essential for the SCADA systems operation. Since data encryption isn't always possible on these systems, we also recommend implementing other anti MITM measures such as installing software with anti-ARP spoofing solutions and/or setting up packet filtering to avoid certain ARP packets. If possible, adding static ARP entries into the cache is one of the best methods to prevent ARP spoofing, but it is a difficult solution to implement on larger systems (like SCADA), as each device in the network would need to be configured manually. Other MITM solutions include locking MAC addresses. Additionally, we can prevent a large number, if not all of attacks that exploit vulnerable applications running on open ports by upgrading the said applications. Although this is an issue in SCADA systems, as they need to maintain availability, we strongly recommended attempting software upgrades by exploring the redundancy feature.

Chapter 6

Conclusions and Future Work

Supervisory Control And Data Acquisition (SCADA) systems play a vital role in modern industrial and critical infrastructures (CI). Cyber-attacks on these systems can case large amounts of financial loss or even cause disasters. Due to the critical importance of these systems, this work's aim was to evaluate and improve the security of these systems by performing a cybersecurity analysis of the system, using current standards, client requisites and penetration testing as a basis.

During the work's duration, we developed two approaches to attack this problem, a theoretical approach and practical approach.

To apply the theoretical approach we developed, one must follow two major steps. The first step is to highlight the recommendations, requirements, controls, and best practices from the standards provided that are applicable to ones SCADA system. Following this we recommend one to analyze a client's set of specifications, and to associate each cybersecurity requisite with its respective recommendation, requirement, control and best practice. By doing this, when solutions are presented to assure requisite compliance, they will automatically assure standard compliance.

When applying the practical approach, we also recommend a two-step approach. Firstly we recommend the use of our threat model to help identify the most common and probable entry points to access the SCADA network, network assets, and attack vectors to obtain these assets. Once all these key elements are identified, we recommend using our penetration testing methodology to verify the existed of these attack vectors.

With the assistance of EFACEC, we tested these approaches, by applying them to a prototype system of their ScateX# system. The results we obtained showed that these approaches successfully identified the requisites that weren't complied. In the theoretical approach we analyzed the systems architecture to evaluate requisite compliance. With the corrective measures that were also provided on this approach, we were able to propose effective solutions to correct the non-compliances that were found. We also applied the practical approach on the system, and were able to demonstrate both attack vectors and requisite non-compliance. These results were not demonstrated in this work for confidential reasons.

In terms of future work, there is still work that can be done in each of the proposed approaches regarding the cybersecurity analysis of a SCADA system. For more advances future work we

could test the solutions that were proposed to assure compliance by implementing the solution and applying the penetration testing methodology once again.

For approach improvements we could:

- For the theoretical approach, additional work can be done on the solution provided to assure requisite compliance. Since each SCADA system is different depending on the client, we could opt to present various solutions to the problem, effectively guaranteeing a solution to solve the problem.
- For the practical approach, additional tests could be conducted in the penetration methodology. There are a large variety of attack types that weren't tested in this work, such as replay attacks, forged packets, sql injections, DDoS, and others.

Appendix A

DNP3 Function Codes

Table A.1: DNP3 Application Layer Function Codes (Complete List) [12]

Requests (Hex)	
0 Confirm	11 Start application
1 Read	12 Stop application
2 Write	13 Save configuration
3 Select	14 Enable unsolicited
4 Operate	15 Disable unsolicited
5 Dir operate	16 Assign class
6 Dir operate – No Ack	17 Delay measurement
7 Freeze	18 Record current time
8 Freeze – No Ack	19 Open file
9 Freeze clear	1A Close file
A Freeze clear – No Ack	1B Delete file
B Freeze at time	1C Get file information
C Freeze at time – No Ack	1D Authenticate file
D Cold restart	1E Abort file
E Warm restart	1F Activate config
F Initialize data	20 Authentication request
10 Initialize application	21 Authentication request –No acknowledgment
Responses (Hex)	
81 Response	83 Authentication response
82 Unsolicited response	

Appendix B

Operating System and Tools Used

B.1 Kali Linux

Kali Linux is an open source operating system that is maintained and funded by Offensive Security. The operating system has over 600 pre-installed tools used for perpetration testing, reverse engineering and forensics. Kali Linux was released in 2013, and is an extension of the previously famed penetration testing OS, BackTrack Linux [81].

B.2 Wireshark

Wireshark is the world's most widely used network protocol analyzer. It allows the capture and dissection of traffic running in the network. Wireshark is an open source tool that is capable of running on most computing platforms including Windows, Linux, and UNIX [89].

B.3 NMAP

NMAP, or Network Mapper, is a free and open source tool used for network discovery and security auditing. It uses raw IP packets to determine what hosts are available on the network, what services are being offered by the hosts, what operating system they are running, what services are being filtered by firewalls and other characteristics [90].

B.4 Ettercap

Ettercap is a comprehensive suite for MITM attacks. It provides sniffing and content filtering of live connections. The tool supports active and passive dissection of many protocols and includes many features for network and host analysis [91].

B.5 Metasploit

The Metasploit Project is a computer security project that provides information about security vulnerabilities and assists in penetration testing and IDS signature development. The Metasploit Framework (msf) is a Ruby-based open source sub-project. It's a collection of commonly used tools that provide a complete environment for penetration testing and exploit development [92].

References

[1] Petr Matoušek. Description and analysis of iec 104 protocol. Technical report, 2017. URL: http://www.fit.vutbr.cz/research/view_pub.php?id=11570.

[2] Peter Maynard, Kieran McLaughlin, and Berthold Haberler. Towards Understanding Man-in-the-middle Attacks on IEC 60870-5-104 SCADA Networks. In *ICS-CSR*, 2014.

[3] Guillermo A Francia III, Xavier P Francia, and Anthony M Pruitt. Towards an In-depth Understanding of Deep Packet Inspection Using a Suite of Industrial Control Systems Protocol Packets. *Journal of Cybersecurity Education, Research and Practice*, 2016(2):2, 2016.

[4] Jeyasingam Nivethan and Mauricio Papa. A Linux-based firewall for the DNP3 protocol. In *Technologies for Homeland Security (HST), 2016 IEEE Symposium on*, pages 1–5. IEEE, 2016.

[5] SCHNEIDER AUTOMATION. Modbus messaging on tcp/ip implementation guide v1. 0b. *MODBUS Organization*, 30, 2015.

[6] Acromag. Introduction to modbus tcp/ip. Technical report, ACROMAG INCORPORATED, January 2005.

[7] David Kushner. The real story of stuxnet, 2013. Available at https://spectrum.ieee.org/telecom/security/the-real-story-of-stuxnet/, accessed 06-January-2019.

[8] Georg Disterer. Iso/iec 27000, 27001 and 27002 for information security management. *Journal of Information Security*, 4(02):92, 2013.

[9] Isa99 Committee. The 62443 Series of Standards, February 2018. http://isa99.isa.org/Public/Information/The-62443-Series-Overview.pdf, accessed March 2018.

[10] John R Vacca. *Computer and information security handbook*. Newnes, 2012.

[11] Clint Bodungen, Bryan Singer, Aaron Shbeeb, Kyle Wilhoit, and Stephen Hilt. *Hacking Exposed Industrial Control Systems: ICS and SCADA Security Secrets & Solutions*. McGraw-Hill Education Group, 1st edition, 2016.

[12] DNP Technical Committee. DNP3 Application Note AN2013-004b Validation of Incoming DNP3 Data, 2014. Available at https://www.dnp.org/DNP3Downloads/DNP3%20AN2013-004b%20Validation%20of%20Incoming%20DNP3%20Data.pdf, accessed 15-December-2018.

[13] Kostas Mathioudakis, Nick Frangiadakis, Andreas Merentitis, and Vangelis Gazis. Towards generic scada simulators: A survey of existing multi-purpose co-simulation platforms, best practices and use-cases. *AGT Group (R&D)*, 2013.

[14] Sajid Nazir, Shushma Patel, and Dilip Patel. Assessing and augmenting scada cyber security: A survey of techniques. *Computers & Security*, 70:436–454, 2017.

[15] T. Sommestad, G. N. Ericsson, and J. Nordlander. Scada system cyber security - a comparison of standards. In *IEEE PES General Meeting*, pages 1–8, July 2010. doi:10.1109/PES.2010.5590215.

[16] Jan Vávra and Martin Hromada. An evaluation of cyber threats to industrial control systems. In *Military Technologies (ICMT), 2015 International Conference on*, pages 1–5. IEEE, 2015.

[17] Dale Barr and P Fonash. Supervisory control and data acquisition (scada) systems. *National Communications System (NCS), Technical Information Bulletin*, pages 04–1, 2004.

[18] Mark S Javate. Study of adversarial and defensive components in an experimental machinery control systems laboratory environment. Technical report, NAVAL POSTGRADUATE SCHOOL MONTEREY CA, 2014.

[19] Y. Yang, K. McLaughlin, T. Littler, S. Sezer, E. G. Im, Z. Q. Yao, B. Pranggono, and H. F. Wang. Man-in-the-middle attack test-bed investigating cyber-security vulnerabilities in smart grid scada systems. In *International Conference on Sustainable Power Generation and Supply (SUPERGEN 2012)*, pages 1–8, Sep. 2012. doi:10.1049/cp.2012.1831.

[20] Anam Sajid, Haider Abbas, and Kashif Saleem. Cloud-assisted iot-based scada systems security: A review of the state of the art and future challenges. *IEEE Access*, 4:1375–1384, 2016.

[21] Tim Kilpatrick, Jesus Gonzalez, Rodrigo Chandia, Mauricio Papa, and Sujeet Shenoi. An architecture for scada network forensics. In *Advances in digital forensics II*, pages 273–285. Springer, 2006.

[22] György Dán, Henrik Sandberg, Mathias Ekstedt, and Gunnar Björkman. Challenges in power system information security. *IEEE Security & Privacy*, 10(4):62–70, 2012.

[23] ICS-CERT NCCIC, DHS. Recommended practice: Improving industrial control system cybersecurity with defense-in-depth strategies, 2016. Available at https://ics-cert.us-cert.gov/sites/default/files/recommended_practices/NCCIC_ICS-CERT_Defense_in_Depth_2016_S508C.pdf, accessed 06-January-2019.

[24] Chee-Wooi Ten, Chen-Ching Liu, and Govindarasu Manimaran. Vulnerability assessment of cybersecurity for scada systems. *IEEE Transactions on Power Systems*, 23(4):1836–1846, 2008.

[25] K Stouffer, S Lightman, V Pillitteri, M Abrams, and A Hahn. NIST special publication 800-82, revision 2: Guide to industrial control systems (ICS) security. *National Institute of Standards and Technology*, 2014.

[26] ISA/IEC. 62443-1-1 - 62443-4-2, Multi. Multi. International Society of Automation (ISA), International Electrotechnical Commission (IEC).

[27] SANS. Industrial control systems (ICS) security resources, 2015. Available at https://www.sans.org/security-resources/posters/control-systems-target/120/download, accessed 07-February-2019.

[28] IndigoSCADA. Software architecture, 2014. Available at http://www.enscada.com/a7khg9/IndigoSCADA.html, accessed 02-January-2019.

[29] OSCADA. Architecture, 2018. Available at http://oscada.org/wiki/About, accessed 02-January-2019.

[30] Rapidscada. Software architecture, 2018. Available at http://doc.rapidscada.net/content/en/software-overview/software-architecture.html, accessed 02-January-2019.

[31] Siemens AG. Connecting to everything: Opc ua breaks communications barrier, 2016. Available at https://w3.siemens.com/mcms/human-machine-interface/en/visualization-software/simatic-wincc-open-architecture/news-overview/pages/wincc-oa-newsletter-2016-3-news03.aspx, accessed 06-January-2019.

[32] ABB. Abb ability™ system 800xa architecture, 2018. Available at https://new.abb.com/control-systems/system-800xa/800xa-dcs/system/architecture, accessed 06-January-2019.

[33] EFACEC. Scatex+ sistema para gestão de redes, 2016. Available at http://www.efacec.pt/wp-content/uploads/2016/10/CS45P1301B1_box_rounded.pdf, accessed 27-December-2018.

[34] James H. Graham. Security considerations in scada communication protocols. 2004.

[35] Gordon Clarke, Deon Reynders, and Edwin Wright. *Practical modern SCADA protocols: DNP3, 60870.5 and related systems*. Newnes, 2004.

[36] CISCO. What is the difference: Viruses, worms, trojans, and bots?, 2018. Available at https://www.cisco.com/c/en/us/about/security-center/virus-differences.html, accessed 04-January-2019.

[37] Stamatis Karnouskos. Stuxnet worm impact on industrial cyber-physical system security. In *IECON 2011-37th Annual Conference on IEEE Industrial Electronics Society*, pages 4490–4494. IEEE, 2011.

[38] Defense Use Case. Analysis of the cyber attack on the ukrainian power grid. *Electricity Information Sharing and Analysis Center (E-ISAC)*, 2016.

[39] Kim Zetter. Inside the cunning, unprecedented hack of ukraine's power grid, 2016. Available at https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/, accessed 04-January-2019.

[40] Henrik Waagsnes. Scada intrusion detection system test framework. Master's thesis, Universitetet i Agder; University of Agder, https://brage.bibsys.no/xmlui/handle/11250/2455016, 9 2017.

[41] ISO IEC. Rules for the structure and drafting of international standards, 2016. Available at https://boss.cen.eu/ref/ISO_IEC_Directives_Part2.pdf, accessed 28-December-2018.

[42] ISO. ISO/IEC 27002:2013, October 2013. International Organization for Standardization (ISO), International Electrotechnical Commission (IEC).

[43] NERC. About nerc, 2018. Available at https://www.nerc.com/AboutNERC/Pages/default.aspx, accessed 20-November-2018.

[44] NERC. CIP-002-5.1a - CIP-014-2, Multi. 2016-2018. North American Electric Reliability Corporation (NERC).

[45] CPNI. About cpni, 2018. Available at https://www.cpni.gov.uk/about-cpni, accessed 20-November-2018.

[46] Ross Anderson and Shailendra Fuloria. Security economics and critical national infrastructure. In *Economics of Information Security and Privacy*, pages 55–66. Springer, 2010.

[47] CPNI. Good Practice Guide, Process Control and SCADA Security. Guide 1-7, n/a 2008. Centre for the Protection of National Infrastructure (CPNI).

[48] Sean Turner and Tim Polk. Prohibiting secure sockets layer (SSL) version 2.0 (RFC 6176). Technical report, Internet Eng. Task Force IETF, March 2011. URL: https://www.rfc-editor.org/info/rfc6176, doi:10.17487/RFC6176.

[49] Richard Barnes, Martin Thomson, Alfredo Pironti, and Adam Langley. Deprecating secure sockets layer version 3.0 (RFC 7568). Technical report, Internet Eng. Task Force IETF, June 2015. URL: https://www.rfc-editor.org/info/rfc7568, doi:10.17487/RFC7568.

[50] K. Moriarty and S. Farrell. Deprecating TLSv1.0 and TLSv1.1 (DRAFT), 2018. Available at https://tools.ietf.org/html/draft-ietf-tls-oldversions-deprecate-00, accessed 10-February-2019.

[51] K McKay and D Cooper. NIST special publication 800-52, revision 2: Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations. *National Institute of Standards and Technology*, 2017.

[52] S Frankel, P Hoffman, A Orebaugh, and R Park. NIST Special Publication 800-113, Guide to SSL VPNs. *National Institute of Standards and Technology (NIST), US Department Commerce*, 2008.

[53] EPRI. Inter-Control Center Communications Protocol (ICCP, TASE.2): Threats to Data Security and Potential Solutions, October 2001. Electric Power Research Institute (EPRI), Palo Alto, CA. 1001977.

[54] IEC. IEC 62351-1 - 62351-13: Power systems management and associated information exchange - Data and communications security, Multi. Multi. International Electrotechnical Commission (IEC).

[55] OpenSSL. TLS1.3, 2018. Available at https://wiki.openssl.org/index.php/TLS1.3, accessed 11-February-2019.

[56] Tibor Jager, Saqib A Kakvi, and Alexander May. On the security of the pkcs# 1 v1. 5 signature scheme. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1195–1208. ACM, 2018.

[57] Chris Hoffman. Why You Shouldn't Use SMS for Two-Factor Authentication (and What to Use Instead), 2017. Available at https://www.howtogeek.com/310418/why-you-shouldnt-use-sms-for-two-factor-authentication/, accessed 10-April-2018.

[58] Paul A Grassi, JL Fenton, EM Newton, RA Perlner, AR Regenscheid, WE Burr, JP Richer, NB Lefkovitz, JM Danker, YY Choong, et al. NIST Special Publication 800-63B. Digital Identity Guidelines: Authentication and Lifecycle Management. *Bericht, NIST*, 2017.

[59] Owasp. Testing Multiple Factors Authentication (OWASP-AT-009), 2013. Available at https://www.owasp.org/index.php/Testing_Multiple_Factors_Authentication_(OWASP-AT-009), accessed 10-April-2018.

[60] H. Rowland, L. Poggemeyer, N. Washburn, J. Hall, and R. Puffer. Configure user access control and permissions, 2018. Available at https://docs.microsoft.com/en-us/windows-server/manage/windows-admin-center/configure/user-access-control, accessed 11-February-2019.

[61] Hugo Krawczyk, Ran Canetti, and Mihir Bellare. HMAC: Keyed-hashing for message authentication. 1997.

[62] Katrin Hoeper and Lily Chen. NIST special publication 800-120: Recommendation for EAP Methods Used in Wireless Network Access Authentication. *National Institute of Standards and Technology*, 2017.

[63] Ravi S Sandhu, Edward J Coyne, Hal L Feinstein, and Charles E Youman. Role-based access control models. *Computer*, 29(2):38–47, 1996.

[64] David F Ferraiolo, Ravi Sandhu, Serban Gavrila, D Richard Kuhn, and Ramaswamy Chandramouli. Proposed NIST standard for role-based access control. *ACM Transactions on Information and System Security (TISSEC)*, 4(3):224–274, 2001.

[65] Owasp. Secure Coding Cheat Sheet, 2018. Available at https://www.owasp.org/index.php/Secure_Coding_Cheat_Sheet, accessed 16-April-2018.

[66] Owasp. Testing for weak password change or reset functionalities (OTG-AUTHN-009), 2018. Available at https://www.owasp.org/index.php/Testing_for_weak_password_change_or_reset_functionalities_(OTG-AUTHN-009), accessed 16-April-2018.

[67] ENISA. Guidelines for SMEs on the security of personal data processing, January 2017. European Union Agency for Network and Information Security (ENISA).

[68] NIST. Hash Functions, 2018. Available at https://csrc.nist.gov/projects/hash-functions, accessed 18-April-2018.

[69] Valery Kamaev, Alexey Finogeev, Anton Finogeev, and Sergey Shevchenko. Knowledge discovery in the scada databases used for the municipal power supply system. In *Joint Conference on Knowledge-Based Software Engineering*, pages 1–14. Springer, 2014.

[70] Oracle Advanced Security Transparent Data Encryption Best Practices. White Paper, July 2012.

[71] Owasp. OWASP Secure Coding Practices - Quick Reference Guide, 2010. Available at https://www.owasp.org/index.php/OWASP_Secure_Coding_Practices_-_Quick_Reference_Guide, accessed 16-April-2018.

[72] Aaron B Brown and Alexander Keller. A best practice approach for automating it management processes. In *2006 IEEE/IFIP Network Operations and Management Symposium NOMS 2006*, pages 33–44. IEEE, 2006.

[73] C. Plett and L. Poggemeyer. Windows server update services (WSUS), 2017. Available at https://docs.microsoft.com/en-us/windows-server/administration/windows-server-update-services/get-started/windows-server-update-services-wsus, accessed 11-February-2019.

[74] Daniel Berman. 6 open source SIEM tools, 2018. Available at https://logz.io/blog/open-source-siem-tools/, accessed 13-February-2019.

[75] Moukafih Nabil, Sabir Soukainat, Abdelmajid Lakbabi, and Orhanou Ghizlane. Siem selection criteria for an efficient contextual security. In *2017 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–6. IEEE, 2017.

[76] Daniel Berman. Using the ELK stack for SIEM, 2018. Available at https://logz.io/blog/elk-siem/, accessed 13-February-2019.

[77] Jerome H Saltzer and Michael D Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 63(9):1278–1308, 1975.

[78] Xiaopu Ma, Ruixuan Li, Zhengding Lu, Jianfeng Lu, and Meng Dong. Specifying and enforcing the principle of least privilege in role-based access control. *Concurrency and Computation: Practice and Experience*, 23(12):1313–1331, 2011.

[79] Roland C Bodenheim. Impact of the shodan computer search engine on internet-facing industrial control system devices. Technical report, AIR FORCE INSTITUTE OF TECHNOLOGY WRIGHT-PATTERSON AFB OH GRADUATE SCHOOL OF ENGINEERING AND MANAGEMENT, 2014.

[80] Timo Kiravuo, Seppo Tiilikainen, Mikko Särelä, and Jukka Manner. Peeking under the skirts of a nation: finding ics vulnerabilities in the critical digital infrastructure. In *European Conference on Cyber Warfare and Security*, page 137. Academic Conferences International Limited, 2015.

[81] Kali. About kali linux, 2018. https://www.kali.org/about-us/, accessed 28-March-2018.

[82] Nicholas R Rodofile, Kenneth Radke, and Ernest Foo. Framework for SCADA cyber-attack dataset creation. In *Proceedings of the Australasian Computer Science Week Multiconference*, page 69. ACM, 2017.

[83] Ping Chen, Lieven Desmet, and Christophe Huygens. A study on advanced persistent threats. In *IFIP International Conference on Communications and Multimedia Security*, pages 63–72. Springer, 2014.

[84] Justin Searle. Dangers of port scanning, 2014. Available at https://drive.google.com/file/d/1lMaDVTNRXNr0yEfr2dW7HuZYohaEpHmL/view, accessed 06-February-2019.

[85] Mauro Conti, Nicola Dragoni, and Viktor Lesyk. A survey of man in the middle attacks. *IEEE Communications Surveys & Tutorials*, 18(3):2027–2051, 2016.

[86] Peter Maynard. ettercap-104-mitm, 2018. Available at https://github.com/PMaynard/ettercap-104-mitm, accessed 25-November-2018.

[87] Filipe Rocha. SCADA-ettercap-MITM, 2018. Available at https://github.com/filipepestana/SCADA-ettercap-MITM, accessed 20-December-2018.

[88] Offensive Security. About the metasploit meterpreter, 2018. Available at https://www.offensive-security.com/metasploit-unleashed/about-meterpreter/, accessed 28-December-2018.

[89] Wireshark. About wireshark, 2018. https://www.wireshark.org/about.html, accessed 28-March-2018.

[90] NMAP. Network Mapper Security Scanner, 2018. Available at https://nmap.org/, accessed 02-May-2018.

[91] Alberto Ornaghi and Marco Valleri. Ettercap, 2018. Available at https://www.ettercap-project.org/index.html, accessed 02-May-2018.

[92] Rapid7. Metasploit framework, 2018. https://metasploit.help.rapid7.com/docs/msf-overview, accessed 28-March-2018.