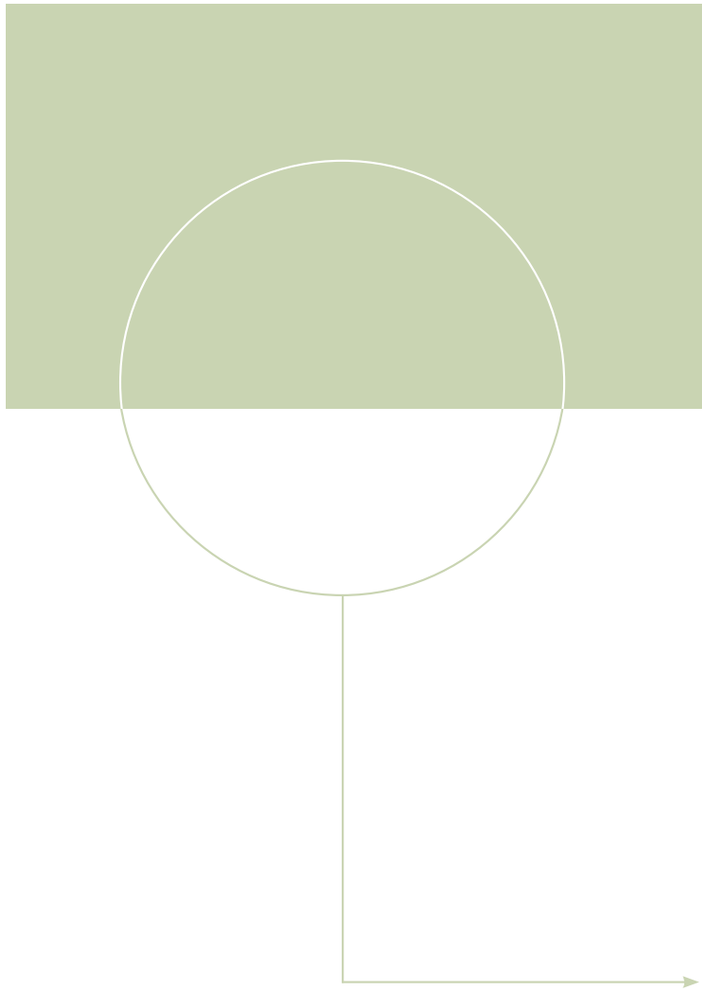

Master's thesis

2019

Kari Anette Sand

NTNU
Norwegian University of
Science and Technology
Faculty of Information Technology and Electrical
Engineering
Department of Information Security and Communication
Technology

Kari Anette Sand

Incident Handling, Forensics Sensors and Information Sources in Industrial Control Systems

June 2019

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

NTNU

Norwegian University of
Science and Technology

Incident Handling, Forensics Sensors and Information Sources in Industrial Control Systems

Information Security
Submission date: June 2019
Supervisor: Prof. Stephen D. Wolthusen
Co-supervisor: Dr. Vasileios Gkioulos

Norwegian University of Science and Technology
Department of Information Security and Communication Technology

Acronyms

AV Anti Virus.

CI Critical Infrastructure.

CIA Confidentiality, Integrity and Availability.

DNP3 Distributed Network Protocol.

ENISA European Union Agency for Network and Information Security.

EWS Engineering Workstation.

FF Foundation Fieldbus.

GOOSE Generic Object Oriented Substation Event.

HART Highway Addressable Remote Transducer.

HIDS Host-based Intrusion Detection System.

HMI Human Machine Interface.

ICS Industrial Control System.

IDS Intrusion Detection System.

IR Incident Response.

MMS Manufacturing Message Specification.

NIDS Network-based Intrusion Detection System.

NIST National Institute of Standards and Technology.

NSM Network Security Monitoring.

PLC Programmable Logic Controller.

RTU Remote Terminal Unit.

SCADA Supervisory Control And Data Acquisition.

SIEM Security Information and Event Management.

SMV Sampled Measured Values.

Preface

This Master's thesis is a product of a research study conducted at the Department of Information Security and Communication Technology at NTNU. It was carried out over the spring semester of 2019. The topic of this research study was provided in collaboration with KraftCERT and NTNU. The research topic is a part of a project that is concerned with the development of a cyber situational awareness architecture for industrial control systems, where this thesis focuses on monitoring sensors in the industrial control systems.

The content of this thesis is aimed at readers that have experience in the industrial controller systems field or the field of information security with interest for industrial controller systems.

01-06-2018
Kari Anette Sand

Acknowledgment

I want to thank my supervisor, Prof. Stephen D. Wolthusen, and co-supervisor, Dr. Vasileios Gkioulos, for continuous support and outstanding guidance throughout this project. They have demonstrated exceptional commitment to this project and given me the opportunity to learn and study the field of critical infrastructure.

Secondly, I would like to thank the employees at KraftCERT, and especially Lars Erik Smevold for excellent support and assistance in this project. KraftCERT provided the physical lab environment that was used in this project, introduced me to representatives from the power utility sector and was involved in numerous discussions around the topic of this thesis.

Last, I would like to thank my family for tremendous support and encouragement through the five years of my studies at Gjøvik. And lastly, my boyfriend and classmate Odin Jenseg, for incredible support and scientific discussions.

K.A.S.

Contents

List of Figures

List of Tables

Abstract

Industrial Control Systems are used for controlling physical processes. An example of a physical process is the distribution of power in the power industry. These processes are a part of nations critical infrastructure, and it's, therefore, essential to know how these systems operate in case of an event. Awareness of Industrial Control Systems is imperative to understand the state of the system. A way of obtaining the state of a system is to collect and correlate information from each part of the system. To be able to achieve this monitoring sensor can be used in the system to collect information to be analyzed. Industrial Control Systems' main priority is, however, physical safety if something happens. To be able to understand what happened when an incident occurred it is essential to have a deeper understanding of how an Industrial Control System operate and behave under normal operation. Therefore, it is imperative to know how the information flows in the system, which components that can retain a state and where to place sensors in the system to be able to capture data that can be used for correlating events and give a state of the system even after an event has occurred.

This research study proposes an architecture for monitoring of Industrial Control Systems, the location of the monitoring sensors and which sensor to use is presented. Further, a hardware architecture and a discussion of memory of the four components; Thermal sensor, Remote Terminal Unit, Human Machine Interface, and Supervisory Control And Data Acquisition server is presented. In terms of identifying relevant components that can retain a relevant state after an event has occurred. Lastly, the study proposes were to aggregate the collected network traffic to be able to identify augmenting and corroborating information. Next, to the theoretical results, a lab experiment was conducted in a lab environment to analyze real-time network traffic when a threat actor creates a disturbance in the system.

1 Introduction

This chapter introduces the topic of the thesis, together with the keywords, problem description, the justification, motivation, and benefits of the project. Further the research questions which the research study proposes an answer to are defined and the contribution of the thesis.

1.1 Topic

An Industrial Control System (ICS) is used to control physical processes, e.g., to control the process of distribution of power. An ICS contain controller components which communicate over a closed or distributed network for controlling such processes. Because these systems are controlling physical processes, the consequence of an event or incident can lead to physical damage or worst case, loss of life. An incident is defined as an event that poses a risk to the Confidentiality, Integrity and Availability (CIA) to the organization's operation [14]. Therefore, it is prominent to know how normal information flows in an ICS such that abnormal information flows can be detected and analyzed.

1.2 Keywords

Industrial Control System, Critical Infrastructure, Supervisory Control And Data Acquisition, forensics, protocols, analysis, information flow, power, energy, 61850, 60870, 9506 and 62351.

1.3 Cyber Situational Security Awareness Architecture for Industrial Control Systems

This thesis is a part of a project at NTNU, where the primary objective is to create a Cyber Situational Security Awareness Architecture (CSSA) for the area of ICS. An attack scenario and a ICS reference architecture are created in collaboration with Niclas Hellesen. The reference architecture is defined in section 4.1 and the attack scenario is defined in section 4.6.

1.4 Problem description

Unexpected downtime is an issue in ICSs because of the effect it can have on the physical process it is controlling. e.g., if downtime occurs in a power plant, the production of power can stop, and the main priority will then be to get the power plant back up again. Because of the need for uptime, it is difficult if an attack occurs because the main priority will be to get the system back up again and not analyze what happened and who attacked the system. There will, therefore, be a loss of information that can be used to track the attacker and to understand what happened.

The loss of information makes it harder for a forensics analyst or an Incident Response (IR) team to be able to understand what happened after an incident took place. The importance of having

enough knowledge about the system and where to collect the needed information get important at this stage.

Because of this problem, it is essential to be prepared before an incident occur. There is a need for guidelines on what type of monitoring sensors that should be used and where they should be placed, what type of data that can be collected from the components, and where aggregation of collected data can be analyzed for identifying augmenting and corroborating information.

1.5 Justification motivation and benefits

Reports show that the number of attacks on Critical Infrastructure (CI) is growing [15] [16]. Symantec indicates the likelihood of more attacks on CI in 2018 in their *Internet Security Threat Report*. Furthermore, Symantec specifically highlights the group Dragonfly that has targeted the European energy sector [16]. In Kaspersky's report *Threat Landscape for Industrial Automation Systems* it is reported that forty percent of ICS computers in their protection were attacked [15]. These numbers were from the first half of 2018.

Next to the growing number of attacks on ICS different research papers have called attention to the challenge of forensics on ICS components and systems [17] [18] [19]. The increasing attack rate next to the concerns of ICS forensics present the justification and motivation of this research paper. The product of this research will hopefully benefit the challenge of today's methods for forensics on CI.

1.6 Research questions

This thesis proposes an answer to the following research questions:

1. How can high-level semantics of control and information flow in a control system be collected/captured, and what is a typical architecture for capturing this traffic?
2. Which entities in a Industrial Control System can retain relevant state for Incident Response and analysis?
3. Where can augmenting or corroborating information and control flows be found in the Industrial Control System architecture?

1.7 Contribution

This research study focused on the study of identifying and collecting the information and control flow in ICSs. Multiple research papers have highlighted the lack of methods for conducting a forensic analysis on an ICS [11] [17] [18] [19].

The focus of the project has been on what type of monitoring sensors to use and where to place the sensors in the ICS. Four components have been discussed in more technical details to consider if the components are keeping a relevant state after an incident has happened. Lastly, a discussion is placed on where the collected network traffic should be aggregated.

1.8 Thesis outline

The thesis consists of eight chapters, as follows.

Chapter 1 is the introduction to this thesis, which includes the description of the research study.
Chapter 2 contain the related work which is analyzed and used in this master thesis.
Chapter 3 contain the choice of methods which followed in the making of this master thesis.
Chapter 4 is the overview chapter where relevant elements of Industrial Control Systems are given a small description.
Chapter 5 contain the contribution from this project, where each of the sections contains the answer to the research questions.
Chapter 6 contain the physical testing part of this study. Description and results of the testing phase are described.
Chapter 7 contain a discussion for this research study.
Chapter 8 contain the conclusion and a proposal for future work.

2 Related work

This chapter contains the related work to the research questions presented in the introduction chapter. This chapter the standards, guides and protocols are presented. Then follows network and physical data acquisition, forensics in other fields, intrusion detection systems and network security monitoring.

2.1 Standards, guides, and protocols

Standards exist for the different areas in critical infrastructure. The IEC 62443 series [20] focuses on the security of ICS. The ISO/IEC 27019:2017 [21] which is a part of the ISO/IEC 27000 standard series on information security include the control process from production until the last stage, distribution in the power sector. The ISO/IEC 27019 standard uses National Institute of Standards and Technology (NIST) guide as a reference for security in ICS [22, 21]. For this research, the IEC 61850 and IEC 60870 series are of most interest and are included in the overview chapter [3, 7]. The content of these standards focuses on the communication and information objects of the ICS.

The NIST SP 800-82 [22] is a security guide for ICSs. The guide highlights that monitoring, logging, and auditing are needed to be able to carry out a forensic analysis of an ICS. The guide defines the consequence of not collecting enough data as a vulnerability. The given reason for this is that without collecting enough data related to events, it can be unachievable to evaluate and identify the root cause of an event, and even detect an event or incident.

ICSs contains different types of components from the supervisory level down to the controller and physical production layer. These components communicate through the standards presented earlier. However, they also need set communication of protocols to be able to communicate [1]. It exists different proprietary and non-proprietary protocols, and some of these are the Modbus, DNP3, Profibus protocol, IEC 60870-5-101/104 and IEC 61850 series [23]. These protocols are communication protocols which are used between the components from the controller level up to the supervisory level of the ICS.

2.1.1 ENISA

In [1], the European Union Agency for Network and Information Security (ENISA), presents the dependencies of the communication network of an ICS. The report includes an example of an ICS/Supervisory Control And Data Acquisition (SCADA) architecture which will be used as the base for the reference architecture in this research. The architecture are included in figure 1 below. The architecture is an overview of an ICS/SCADA system from the upper level of the enterprise segment of the network down to the production level of the network.

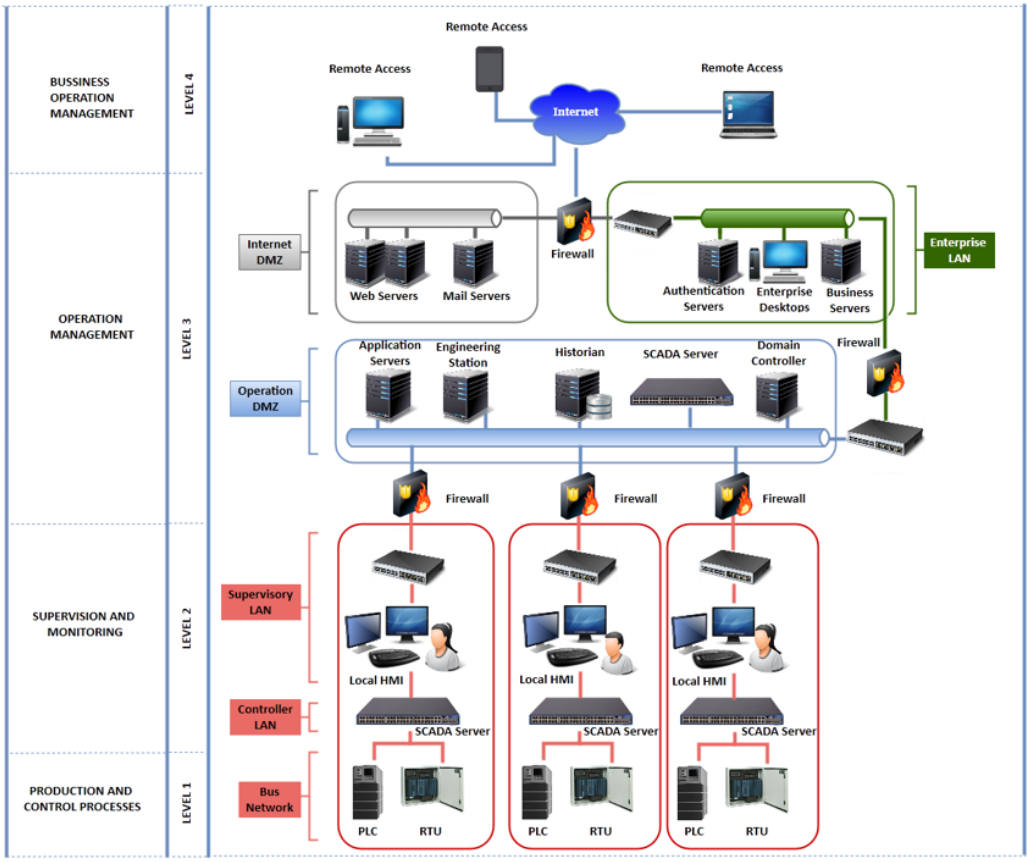

Figure 1: ENISA's ICS/SCADA architecture [1]

ENISA has presented the assets, threats, vulnerabilities and attack scenarios in ICSs. Further, ENISA has collaborated with experts and stakeholders from the field to present three different attack scenarios. One of these three attacks is the compromise of a SCADA system were the objective is the manipulation of data assets or crash system assets. The attack starts with performing reconnaissance such that the target can be identified. When the target is decided a social engineering attack will follow, where an employee of the organization will be deceived to give away login credentials for the enterprise network. The threat actor will then be able to login to the enterprise network and will follow up with exploiting vulnerabilities in components of the network where the goal is to acquire credentials to the SCADA system. When the threat actor has acquired the login credentials, he/she can install a firmware update to the system which includes malicious code such that the threat actor can enter the SCADA system. When the threat actor can log into the SCADA system, the system is compromised. The hacker can now launch an attack on the system which can cause the system to go into a failed state. Furthermore, the hacker can make it difficult for the operators to identify the incident by compromising the backup system and power supply. ENISA has identified

the affected assets in this scenario which is the full SCADA system including the Human Machine Interface (HMI) and the control system which is centralized.

The attacks are of interest to the proposed research because a specific attack scenario are used in the second part. In this part the theoretical assumptions made in part one is tested in an ICS lab environment.

2.2 Network and physical data acquisition

Network and physical data acquisition is a method which is used by multiple researchers [11] [24]. In [11], R.M. van der Knijff has focused on the use of collection through physical and network data acquisition. Furthermore, the researcher has focused on the information flow, to and from a Programmable Logic Controller (PLC) in a SCADA network. The functionalities of a PLC is introduced including an example of the language *Instruction list* in a switching function is presented. A strategy for which information sources that should be prioritized are presented were the *HMI, application server, engineering workstations, databases, historian and firewall logs* have been prioritized the highest. The strategies are followed up by considerations for the preservation of evidence. Here an important note is that when a PLC is turned off on there will be a loss of valuable data and the PLC will end up in an error state. This statement has also been presented in [13]. Network data acquisition is done through intercepting raw data or intercepting flow records in the network. Physical data acquisition is proposed with using the uploading function to the PLC if this is not possible the tool JTAG is suggested. For analyzing the collected data, the Security Onion OS is recommended as a tool for identifying events from the collected data. The paper closes with the concern that event logging and aggregation of logs must be focused on.

In [25], Wakchaure, Sarwade, and Siddavatam are focusing on the reconnaissance of ICSs by analyzing collected raw network data with Deep Packet Inspection (DPI). The collected data which explicitly uses the Modbus/TCP protocol has been selected, and from these, the group has been able to map the full ICS network topology. This is done by collecting raw network data with Wireshark and then aggregating the data to an offsite location. Then outside the network, an analysis of the Ethernet frames, port scanning and identification of payloads are done.

Ahmed, Obermeier, Sudhakaran, and Roussev have focused their research on PLC forensics [13]. In this research, an architectural overview of a PLC is presented. A tool for analyzing network traffic logs of the Programmable Controller Communication Commands (PCCC) protocol has been developed by the same research group [24]. The developed tool can be used to identify unknown file types by analyzing the content of the files. The group tested this, and the researchers were able to classify files with unknown filetypes.

A part of this thesis proposes where the different sensors in a network must be placed and where the information should be aggregated. The research papers above all retrieve communication data between a PLC, and a SCADA server, or retrieve information through physical data acquisition. These methods and their results are, therefore, useful in both the theoretical and practical part of the research topic.

2.3 Forensics in other fields

It is done minimal research on the forensics field in ICSs. Multiple researchers [18] [17] have pointed out that ICS forensics is a challenge.

Forensics methodologies from other fields such as mobile and internet are present [26] and raises the question if these methods can be applied to ICS. Salater [27] indicate that Industrial Automation and Control Systems focuses more on the physical processes while information technology (IT) are focused on information. There is a distinction between the two fields, however, from the papers reviewed earlier in this section it is clear that tools like Wireshark can be used for collecting network traffic between the components. In [28], Sanders has given a full introduction on how to analyze network traffic using Wireshark.

2.4 Intrusion Detection System

Multiple research projects regarding Intrusion Detection System (IDS) in ICS have been done [2], [29], [30]. In [2] Waagsnes has implemented a test framework for IDS in SCADA system. The communication protocols which are compliant and tested by the IDS is the *IEC 60870-5-104, DNP3 & Modbus*. An overview of the data frame of each protocol is presented and briefly explained. Both the general and TCP/IP Modbus frame is included. The data link frame of the DNP3 communication protocol is included below. (The figure is originally from a research paper of Francia III, Francia, and Pruit, as cited in [2]).

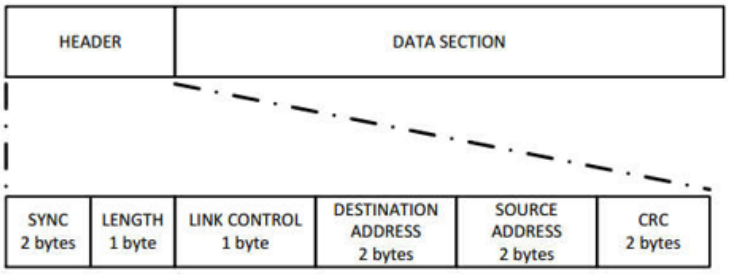

Figure 2: The Data Link frame of DNP3 [2]

The research papers about IDS is interesting because they have analyzed the network flow in the ICS system to be able to generate alarms. It is the research of the architectural level and of normal traffic behavior which is interesting for the proposed project. Figure 2 above is an example of this. The first research question is concerning the communication and information flow in an ICS system. Therefore research papers which have analyzed the architecture or normal behavior of such a system are valuable.

2.4.1 Behavior based anomaly detection

In [31] and [32], anomaly detection IDS is used to analyze collected network traffic. Both of the research studies have studies in normal periodic behavior of the ICS to be able to identify unnormal

behavior in the collected network traffic. In [31], the frequency, and size of packets are used in the analysis of the behavior. In [33], the researchers have made a practical integration for detecting unnormal behavior. The study show a practical take on anomaly detection in ICS.

2.5 Network Security Monitoring

Network Security Monitoring (NSM) consist of the process of detecting and responding to intrusions. The processes consist of collecting, analyzing and escalating systems events [34]. In [34] defines seven different types of data forms that can be used in NSM; *Full content data, extracted content data, session data, transaction data, statistical data, metadata, and alert data*. Full content data, statistical data, metadata, and alert data are described in more detail below.

2.5.1 Full content data

In this data form, the entire data packets transmitted on the wire is considered, which also include the header of each packet [34]. Which means that the collection of full content data can be considered as a full packet capture which can be carried out with tools such as *Wireshark* or *tshark*. Wireshark can also be used to analyze packet headers and complete content of packets.

2.5.2 Statistical data

In this data form, statistical data of the network flow is considered. This can, for example, be a percentage distribution of used protocol types in a set of captured packets or statistical data such as average data length of a set of captured packets [34]. The tool Wireshark have this capability. The command line tool *Capinfos* are also used to obtain statistical data such as the number of packets and the average packet size.

2.5.3 Metadata

In this data form, the metadata is considered, by this it's meant data that can say something more about data, *data about data*. In [34], Bejtlich gives examples of obtaining metadata for an IP address through services such as *whois* and the tool *Robtex*. Where *whois* is used to find information about who the IP can belong to and *Robtex* is used to analyze the routing data of the IP address.

2.5.4 Alert data

In this data form, the alert data is considered, by this it's meant alerts that are generated by tools such as an IDS or Anti Virus (AV). IDS tools such as *Zeek* (earlier Bro), *SNORT* or *Suricata* can be used to generate alerts. These types of tools analyzes the network traffic with a set of rules to decide if an alert should be generated or not. In [34], the tools *Sguil* and *Snorby* is suggested as tools to review the generated alerts.

In section 5.1.4, NIDS, HIDS and AV are discussed further relating to use in the ICSs environment.

3 Choice of Method

This chapter contains a description of the methods that have been used in this project. It begins with a description of the data collections techniques, then the research approach is presented, and followed up with requirements engineering, secure systems engineering and laboratory use.

3.1 Data Collection

In this research, a literature review and unstructured interviews were conducted to collect relevant data. These are described in more detail below in each of its subsections.

3.1.1 Literature review

A literature review comprises the process of retrieving and processing information from earlier research within a specific chosen topic. In [35], Leedy, and Ormrod specify that related literature can be used to formulate a research problem. Along with putting the research problem at hand into a bigger picture of earlier research within the same topic. Furthermore, the researcher's highlight that relevant literature can present a broader theoretical insight into the topic.

In this study, a literature review was conducted to retrieve information about the area of the research questions defined in the introduction. By doing this, the results of the study fit into the bigger picture of this area of research, and where it was possible relevant literature was used as a reference for supporting literature.

3.1.2 Unstructured interview

In the study of this research topic, unstructured interviews with representatives from the energy sector were conducted. The goal of these discussions was to compare the outcome from the early literature review with experts in the field. Such that the results of the theoretical part of this thesis reflected the reality of the field next to relevant scientific research.

The unstructured interviews were held at the early stages of the project, such that the retained knowledge was integrated into the result of the research study. The topics for the discussion were made in advance such that the objective of the meeting was clear.

Summary of meeting with a utility company

An unstructured interview with a utility company was conducted early in the semester, and the topics of the meeting are described in the list below.

- The reference architecture presented in section 4 was discussed and compared to their architecture.
- In this thesis, four different components have been described in more technical depth. In the meeting, a list of prioritized components was made after a discussion.

- In several research papers, it was stated that PLCs stored everything in RAM except for the firmware, and this, therefore, created a problem of conducting a forensic analysis. The presence of this problem in today's PLCs.
- Routines regarding the analysis after an event has happened, and especially the priority of actions.
- The aggregation of logs, network monitoring tool, and storage of network traffic logs.
- Discussion of augmenting and corroborating information and comparison of historical data.

Summary of meetings with kraftCERT

It has been ongoing communication with kraftCERT through this project. Different topics have been discussed, and some of these are listed below.

- General information about the power industry and technology.
- Reference architecture of ICS.
- Physical lab environments, including configuration and lab setup.
- Attack scenario used for the physical testing of this thesis.

3.2 Research Approach

3.2.1 Experimental development/design

Experimental design is a type of quantitative research, where the objective is to analyze a cause-and-effect relationship between a dependent and independent variable [35].

To answer the research questions in section 1.6, the study was divided into two parts. The first part included a literature review where theories were made regarding the research questions concerning the information flow and how the information was retrieved from a ICS. In part two, these theories were tested in a lab environment to verify the feasibility of the proposed methods given in part one.

3.3 Requirements engineering

Requirements engineering is the process of establishing and maintaining a set of requirements that a product needs to satisfy. The product needs to meet the requirements in each phase of the lifecycle. These requirements are jointly formulated by the product owner and the product developer [36].

In section 1.7 the contribution of this project was defined, and in section 1.6, a set of research questions was determined to ensure the objective of the contribution. The research questions served as the requirements for the project, and therefore, the objectives of each part of the project provide an answer to these questions.

3.4 Secure systems engineering

Secure systems engineering are concerned with the insertion of security measures into each phase of the life cycle of a product, such that the assets are protected against vulnerabilities [37].

To be able to insert security measures in a product, it is essential to have an understanding of what the product does and how it communicates. This research focused on the information flow

in an ICS, but also how each component operates and communicate with other components of the network. Therefore, the research study contributes to a more significant understanding of how the product interact. With this, it's simpler to identify where the security measures should be placed and which to use. More in-depth knowledge about the information flow in an ICS can contribute to detect when an incident is occurring, because the information flow will behave abnormally. Furthermore, if an incident occurs, it can be easier to perform a forensic analysis when you know how the normal flow of the system is.

3.5 Laboratory use

In research, a laboratory study can be conducted to assure internal validity [35, p.104]. By performing a laboratory study results from a theoretical research study can be confirmed or dismissed. In a laboratory study, the lab environment is regulated after desired conditions. Usually, an environment close to real-life is wanted.

In the introduction chapter of this thesis, three research questions were defined. A research study was conducted to answer these questions. However, the results of the questions did not always have supported scientific research. Therefore, a laboratory study has been conducted, where a close to a real-life environment of an ICS was used to run an attack scenario to test out the results from the earlier research study that was done.

4 Industrial Control System Overview

4.1 Reference Architecture

The reference architecture used in this thesis is based on ENISA's SCADA architecture in [1]. The reference architecture are included in figure 3 below.

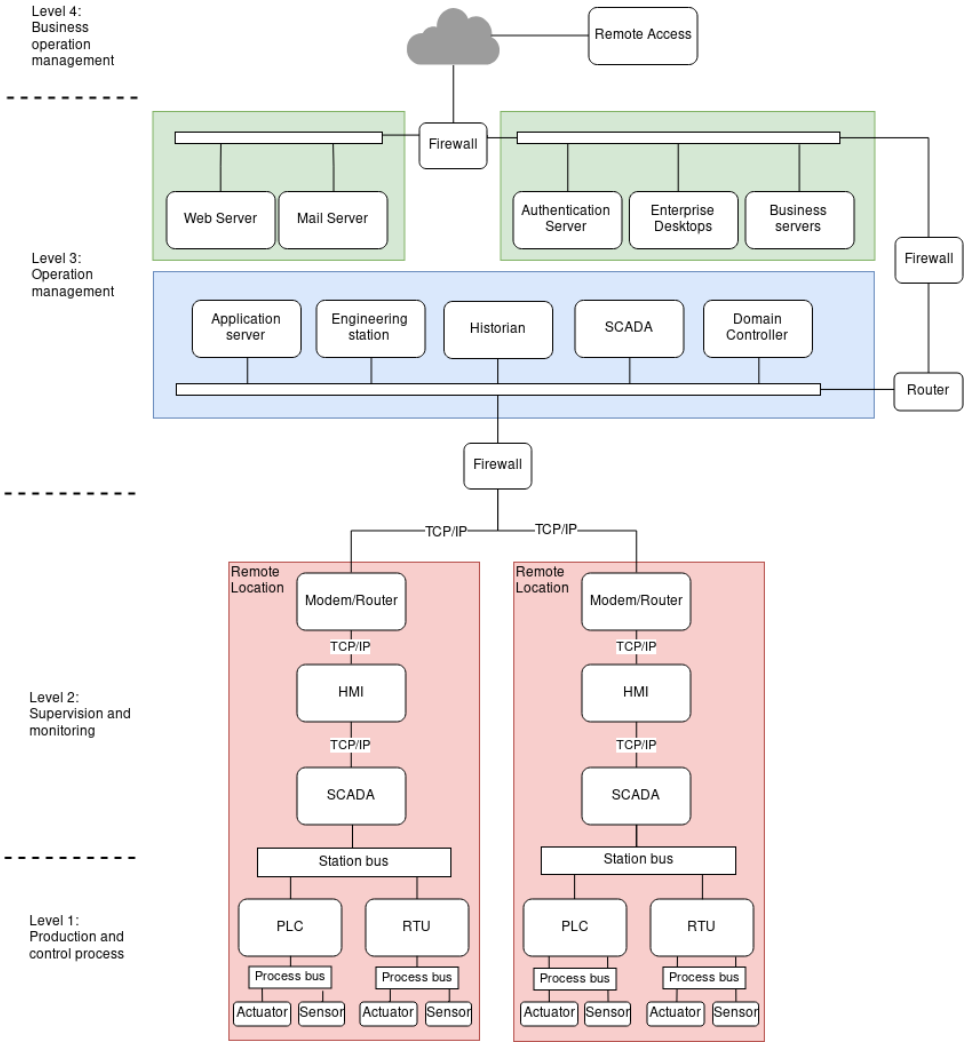

Figure 3: The reference architetcure, based on ENISA's SCADA architecture [1]

4.1.1 Description of architecture

The sections below include a description of components and levels located in an ICS. Afterward, the relevant standards and protocols are described. The sections are considered as a description of the reference architecture presented in section 4.1.

4.2 Levels

The levels of the ICS architecture presented in section 4.1, is using the defined ICS levels from the IEC 62264-1 standard [38]. The standard consist of five levels, from level zero to level four. Level zero is the physical process(es) in the plant, while level one consist of the controller and supervisory layer which is focused on controlling. Level two is focused on the activities which are used for monitoring the physical process, but also controlling. Level three consist of the operation management, while level four is concerned with the upper business level of plant scheduling and logistics [38].

4.3 Components

This section contains a description of some of the components which can be present in a ICS, beginning from the lower process level and up to the operation level in the reference architecture.

4.3.1 Actuators

The actuators are the components which interact with the physical assets in level 1. An example of an actuator is a valve. A PLC analyzes the output from a sensor which can measure the level of water in a tank and then the PLC decides if the valve should be opened or closed.

4.3.2 Sensors

A sensor is a component which can read a predefined set of data from a processing device in level 1. An example of this can be a pressure gauge which measures the pressure in an oil or gas pipeline.

In [39] different methods of categorizing sensors is presented. One of these categorization methods is to categorize sensors by the physical variable the sensors measures. Examples for this can be a magnetic, thermal or electric field sensor. In this project a thermal sensor is chosen. A thermal sensor can be a thermometer which meassures the temperature. The usage of thermometer may require a transmitter between the temperature sensor and a controller entity. It exsist three different transmitters;

- head transmitter
- field transmitter
- rail/panel transmitter

4.3.3 Programmable logic controller

PLC is a component placed in the production and control level that can control underlying devices such as sensors and actuators. The IEC 61131 is the standard for PLCs. A PLC take sensor data as input, and with this data, the controller processes it through a controller processor which will

generate an output. The PLC processes information with the use of ladder logic, which is a programming language [23]. Common Fieldbus protocols are used for communication. Example of these is the Modbus, EtherNet/IP and Profinet protocols [23].

4.3.4 Remote terminal unit

A Remote Terminal Unit (RTU) is a component which is usually placed in level 2 or 1 of the reference architecture. The capability and functionality between a PLC and a RTU is similar, however, a RTU have the possibility to communicate to remote locations through telecommunication [23]. The RTU can be used to connect a remote station at level 2 up to a control station at level 3. This is not the case in the reference architecture, but it's essential to highlight that an RTU have more communication capabilities than a PLC.

4.3.5 Human machine interface

A HMI can be categorized into two areas; local and centralized HMI [1].

Local human machine interface

A local HMI is focused on controlling and monitoring a process in the production and control level (level 2 of the reference architecture). At this level the HMI can communicate with PLCs, RTUs and SCADA servers.

In this thesis, references to a HMI refer to a local HMI.

Centralized human machine interface

A centralized HMI is focused on controlling the production systems in the operation management level [1]. That is level 3 of the reference architecture. A HMI at this level is software on a PC with an operating system such as Windows.

4.3.6 Supervisory Control And Data Acquisition server

SCADA is a type of control system architecture, however, the term *SCADA server* is used for the supervisory workstation inside the SCADA architecture. SCADA servers can be located in level 2 and 3 of the architecture. The purpose of the SCADA server is to display the current state from collected information from the different components in the architecture. The SCADA server usually limits the user to read-only. However, it can be possible to open up for making small changes, such as revise the set points [23].

4.3.7 Historian

A historian is a software system which is usually run on a Windows machine in level 3, Operation Management. The historian collects and archives data from the components in the ICS. Specifically, this can be events in the system or point values [23]. It exists multiple proprietary and third-party software systems.

4.3.8 Engineering workstation

An Engineering Workstation (EWS) is a component located in level 3. The EWS is used to configure and maintain the ICS components, and for running diagnostics of the system [40]. An EWS can also

be found in level 2 of a remote station. An example of use, can, for example, be an engineer using the EWS to configure one or more HMIs for the ICS. EWSs are discussed further in this thesis; an important note is that EWSs have more control over the overall process than the SCADA server.

4.4 Relevant standards

This section contains the relevant standards for this project.

4.4.1 ISO/IEC 27001

The ISO/IEC 27001 standard is a Information Secuirty standard, that contains requirements for an Information security management system (ISMS) which organisations should comply with [41]. When an organization is in compliance with the standard, the organization can be certified.

4.4.2 ISO/IEC 27002

The ISO/IEC 27002 standard is a Information Secuirty standard which concerns security controls in Information Technology (IT). Information Security Policises and Access control are two of the security controls which the standard include [42].

4.4.3 IEC 60870

The IEC 60870 standard is a defintion of SCADA system(s). Part 5 of the standard (IEC 60870-5) include two transmission protocols, IEC 60870-5-101/104, which is described in more detail in section 4.5.7 [7].

4.4.4 IEC 61850

The IEC 61850 standard contain a specification over the usage of a number of communication protocols for ICS. The standard includes how the communication protocols exchange data, what data to exchange and also an engineerings approach to confifure the components data model. Three of the included protocols which can be used on the station bus of the reference architecture is a client/server protocol (which utilizes the Manufacturing Message Specification (MMS) protocol in the application layer), the Generic Object Oriented Substation Event (GOOSE) protocol and the Time Sync (SNTP) protocol. Further, the standard includes three protocols which are used on the process bus of the reference architecture, which is the Sampled Measured Values (SMV), the GOOSE and the SNTP protocol. Figure 4 below, contain an architecture of the protocols included in the standard [3].

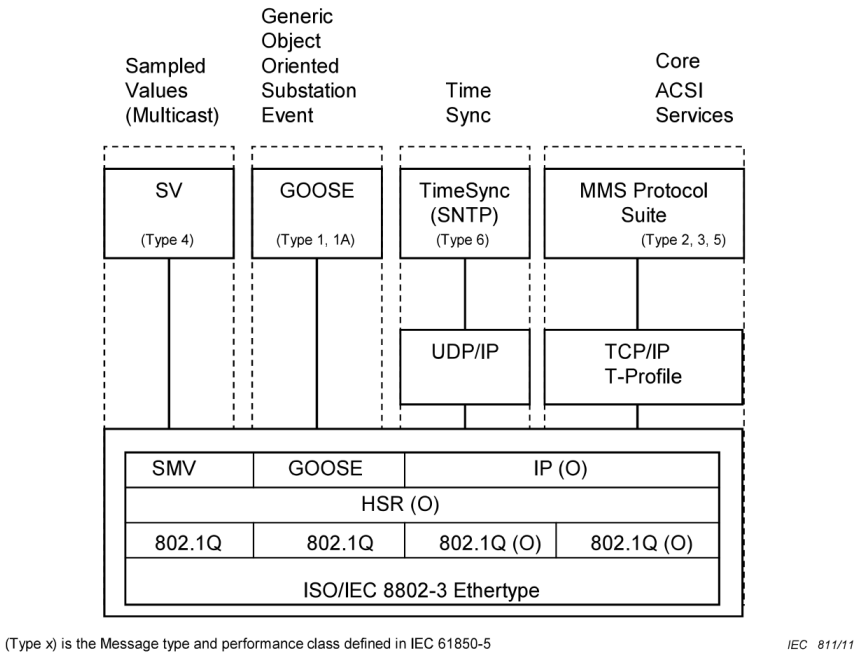

Figure 4: Architecture of the protocols used in IEC 61850 [3]

The GOOSE and MMS protocol is described in section 4.5 below. In section 5.1.2 the client/server protocol are discussed in more detail.

4.4.5 IEC 62351

The IEC 62351 standard defines security for a set of communication protocols used in power systems. The standard is developed by a technical committee named the *TC 57*. The committee is also responsible for the development of standardized communication protocols that are mapped to the IEC 62351 standard. These protocols include the following communication protocols: IEC 60870-5, IEC 60870-6, IEC 61850, IEC 61970 and IEC 61968 [4].

To give a clearer view of the mapping of the IEC 62351 and the communication protocols listed above, figure 5 below is included. This figure is retrieved from the IEC 62351 standard [4].

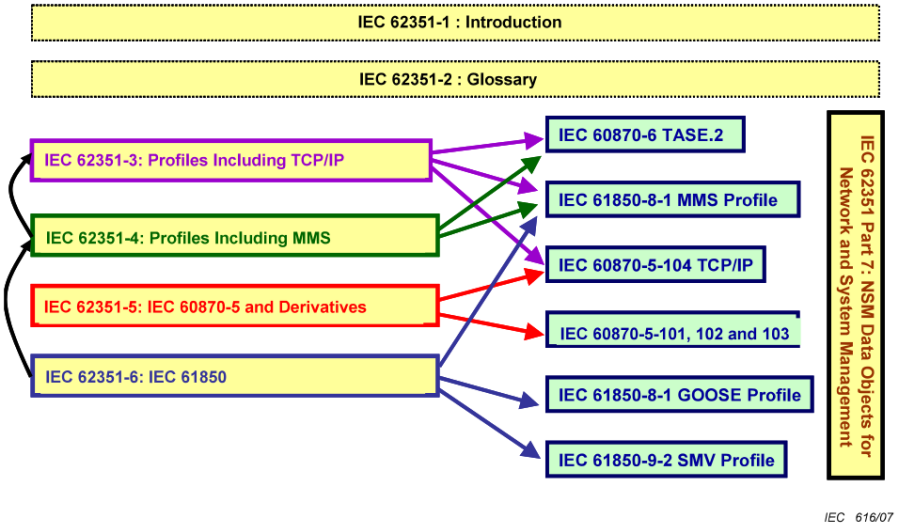

Figure 9 – Correlation between the IEC 62351 series and IEC TC 57 profile standards

Figure 5: Mapping of the IEC 62351 to the IEC 60870-5 & -6, IEC 61850, IEC 61870 and IEC 61968 protocol [4]

Part 5 of the IEC 62351 standard include a secure authentication mechanism for the 60870-5-104 protocol, which is described further in section 5.1.1.

4.4.6 IEC 62443

The IEC 62443 standard is concerned with the security ascpect of ICS [20]. The standard is a collection standards which is divided into four groups, where group 3 is the most relevant for this thesis since it concerned with the system security technologies for the ICS [23].

4.5 Relevant protocols

4.5.1 Highway addressable remote transducer

The Highway Addressable Remote Transducer (HART) protocol is a communication protocol used at layer 1 between sensors/actuators, and controller devices such as PLCs or RTUs. The protocol is used to send digital signals over a current-loop 4..20mA analog wire. Such that both analog and digital signals can be transmitted and received over the analog line at the same time. Both the sensor and controller device need to have support for the HART protocol such that a digital signal will be transmitted, received and parsed correctly. The connected PLC or other controller devices need an input source for analog input that supports the HART protocol. To be able to transmit digital signals over the analog line, the binary value *1* is represented with a 1200Hz signal and the binary value *0* is represented with a 2200Hz signal.

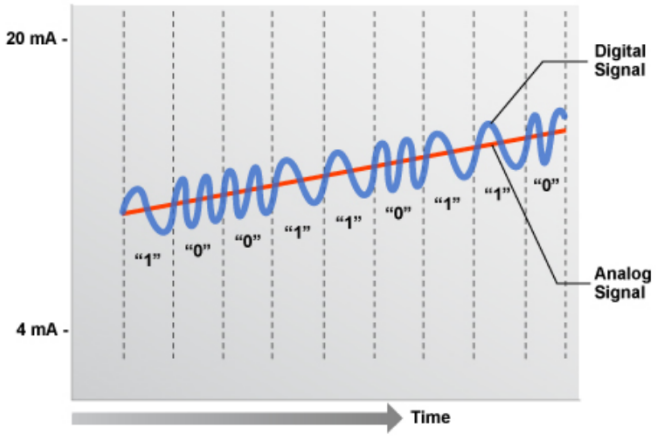

Figure 6: A figure illustrating the HART communication channel [5, p. 11]

In figure 6, the communication channels between the controller device and field devices are presented [5]. In the figure, the analog and digital signal are illustrated as described above, where the blue line is the digital signal and the red line is the analog signal.

The HART protocol is also the foundation for the wireless communication protocol WirelessHART which can be used to communicate wirelessly between components.

4.5.2 Foundation fieldbus

The Foundation Fieldbus (FF) protocol is a communication protocol used at layer 1 of the reference architecture, usually between field instruments (sensors/actuators) and controller components such as PLCs. Compared to the HART protocol described above, the FF protocol transmits data digitally between components. It exist two different types of the protocol;

- Foundation Fieldbus H1
- Foundation Fieldbus High-Speed Ethernet (HSE)

4.5.3 Modbus

The Modbus protocol is a communication protocol, which can be used in multiple positions in a SCADA system. The protocol can be used as the communication protocol between different PLCs and HMI, but also between PLCs slaves and a master PLC in the SCADA system. It exsist multiple variations of the Modbus protocol; Modbus, Modbus RTU, Modbus ASCII and Modbus TCP. The protocol is also upgraded to the Modbus plus protocol [23]. The data frame of the Modbus RTU protocol is included in the figure below.

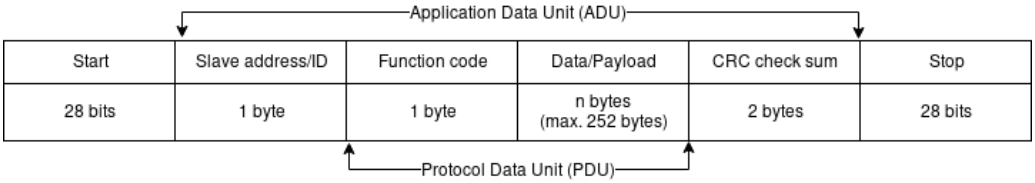

Figure 7: Modbus RTU data frame.

4.5.4 Distributed Network Protocol

The DNP3 protocol is a communication protocol between *master stations* and *slave devices* [23]. The DNP3 protocol can be used between remote slave RTUs and centralized master controller servers, like a SCADA server.

4.5.5 Profibus

The Profibus protocol is a filedbus communication protocol suite, which operate between two types of components; Master and slave component. Where the master is commonly the controller component such as a PLC or RTU and the slave is an actuator or a sensor. In Profibus the master station can also be called an *active* station, and the slave station is called the *passive* station [43]. It exsist multiple variations of Profibus, two of these are Profibus PA and Profibus DP:

- **Profibus Process Automation (PA)** The Profibus PA protocol support the IEC 1158-2 standard on the physical layer.
- **Profibus Decentralised Peripherals (DP)** The Profibus DP protocol support the use of wired transmission such as RS-485, Manchester Coded, Bus Powered (MBP) and ETHERNET, optical transmission such as glass, and wireless transmission on the physical layer [44].

4.5.6 Profinet

Profinet is an Industrial Ethernet Fieldbus communication protocol. The protocol contains three different implementations. The first implementation is used for standard data sent over TCP/UDP. However, the second and third implementation is real-time data protocols, which is used in time-critical environments. The two implementations are mapped directly to the Data-Link Layer of the TCP/IP model architecture [23, p.146]. The protocol can be used between the controllers such as the PLCs and RTUs of the reference architecture.

4.5.7 IEC 60870-5-101/104

The IEC 60870 standard include two transmission protocols for SCADA systems; 60870-5-101 and 60870-5-104. Both of the protocols is used between a *controlling station* in layer 3 and a *controlled station* in layer 2 of the reference architecture. The protocols support the use of multiple controlling stations and multiple controlled stations, in both multipoint point-to-point and multpoint architectures [7].

IEC 60870-5-101

The IEC 60870-5-101, hereby refered to as the -101 protocol, is a serial communication protocol for SCADA systems. The protocol consist of three layers; Application, Data Link and Physical.

IEC 60870-5-104

The IEC 60870-5-104, hereby refered to as the -104 protocol, is a TCP/IP based communication protocol for SCADA systems. The protocol is built upon the -101 protocol to support TCP/IP communication. The protocol consist of five layers; Application, Transport, Network, Data Link and Physical.

4.5.8 Generic object oriented substation event

GOOSE is a communication protocol defined in the IEC 61850 standard [3]. The protocol can be used in level 1 and 2 between field devices of the reference architecture. The protocol is a broadcast and multicast protocol, which means that there are at least one publisher node and one or more subscriber nodes. The transfer time of the protocol is low because it is used for multicasting and broadcasting of events and alarms. The transfer time type is set to 1A and 1 which is defined in IEC 61850-5 standard. The protocol maps data packets from the application layer to the Internet layer of the OSI model which means each data packet are embedded to Ethernet frames. The packages contain the MAC address of the publisher and subscriber.

4.5.9 Manufacturing message specification

MMS is a application layer protocol defined in the ISO 9506 standard. The protocol is used for real-time data exchange between network components or applications [45]. The MMS protocol is considered as a non time-critical protocol, and the transfer type time is set to 2, 3 and 5 which is defined in the IEC 61850-5 standard.

In the IEC 61850 standard the MMS protocol is used in the application layer of the Client/Server communication protocol defined in the standard [3].

4.6 Attack Scenario

A reference attack scenario was made in collaboration with Niclas Hellesen in this project [6]. The objective behind this is to have a base attack scenario for the full project such that thesis' after this in the same project can follow the same test attack scenario.

4.6.1 Description of the attack

This section starts with a description of the threat actor's overall objective of the attack and follows up with the objective of the defender. The section is closed with an attack tree of the attack, where both the threat actor and defenders actions are presented.

Obejctive of threat actor

The objective of the threat actor is to change the value of a set point variable in the configuration of a RTU, bypass alarms and try to avoid detection. The attack is a part of a more significant attack on a power substation. The change of this variable can, for example, disable alarms of future values

that normally are outside of its range.

The objective of the defender

The objective of the defender is to analyze real-time network traffic to detect dangerous commands/changes to the configuration and create an alarm or event to a monitoring system for the operator. The second objective is to be able to gain an understanding of the situational awareness of the ICS based on this network traffic. The monitoring architetcure presented in section 5.1.4 can be used as a reference for the defender in this attack.

Attack tree

In figure 8, the attack is presented in an attack tree. Each of the steps of the threat actor represents the square rounded nodes. The root node contains the objective of the threat actor, which was discussed above. Each hexagon represents the defender's action to detect or prevent the threat actor's movements. Both node 1.1 and 1.2 must be reached to be able to achieve the overall objective of the attack.

The steps of the attack scenario are as follows. First of all, the attack scenario assumes that the threat actor has access to the EWS. The threat actor deactivates the alarms in the application server, and then the modification of the setpoint variable is conducted. Lastly, the threat actor modifies the process data in the historian. Each of the steps has actions which are done to be able to get access to the historian and application layer, but these are described in the tree below.

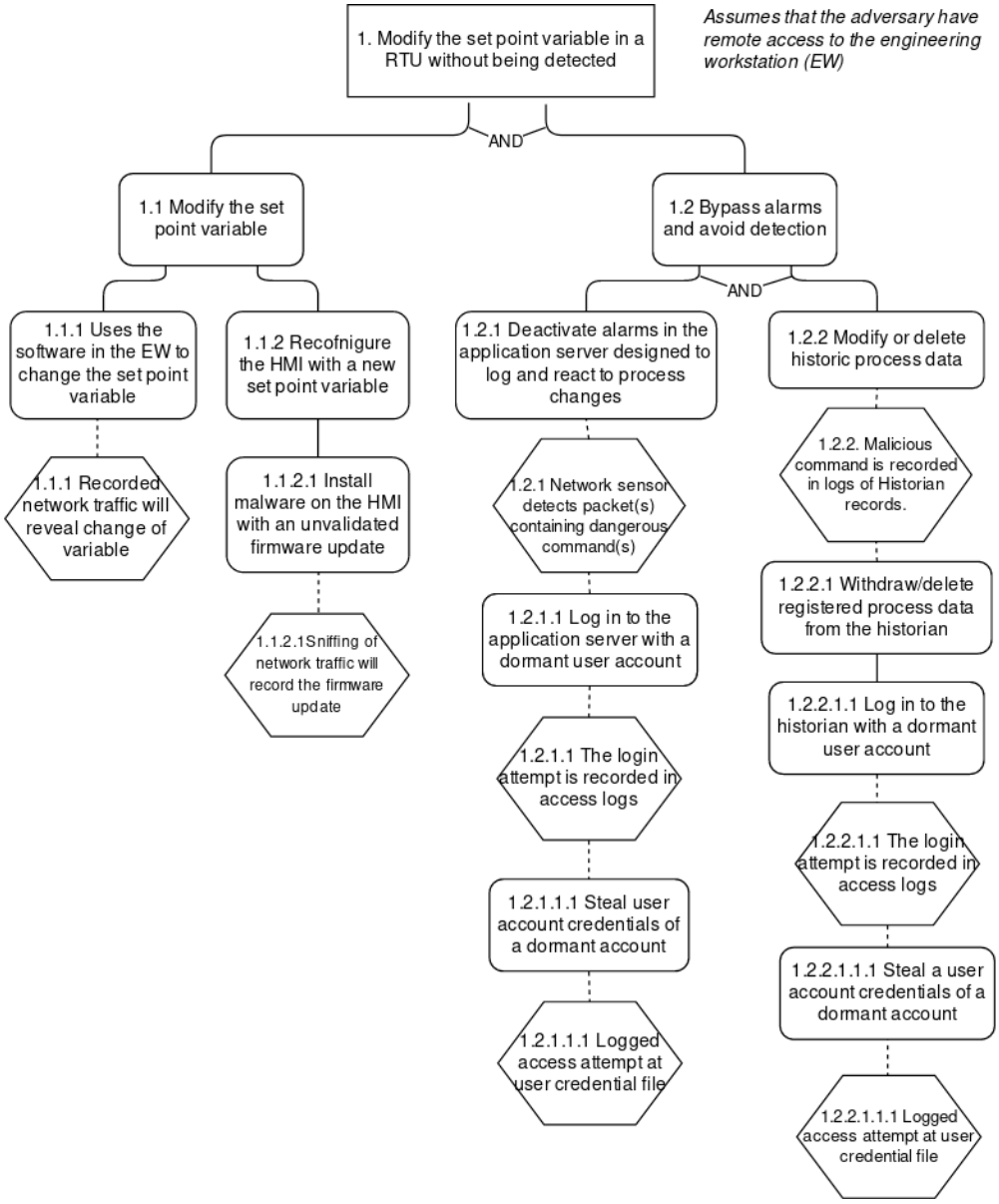

Figure 8: Attack tree of attack scenario, made in colaboration with Niclas Hellesen [6]

Visualization of the attack

In figure 9 below, the attack is visualized in the reference architecture. The numbers in the figure correlates with number of each subnode of the attack tree showed in figure 8.

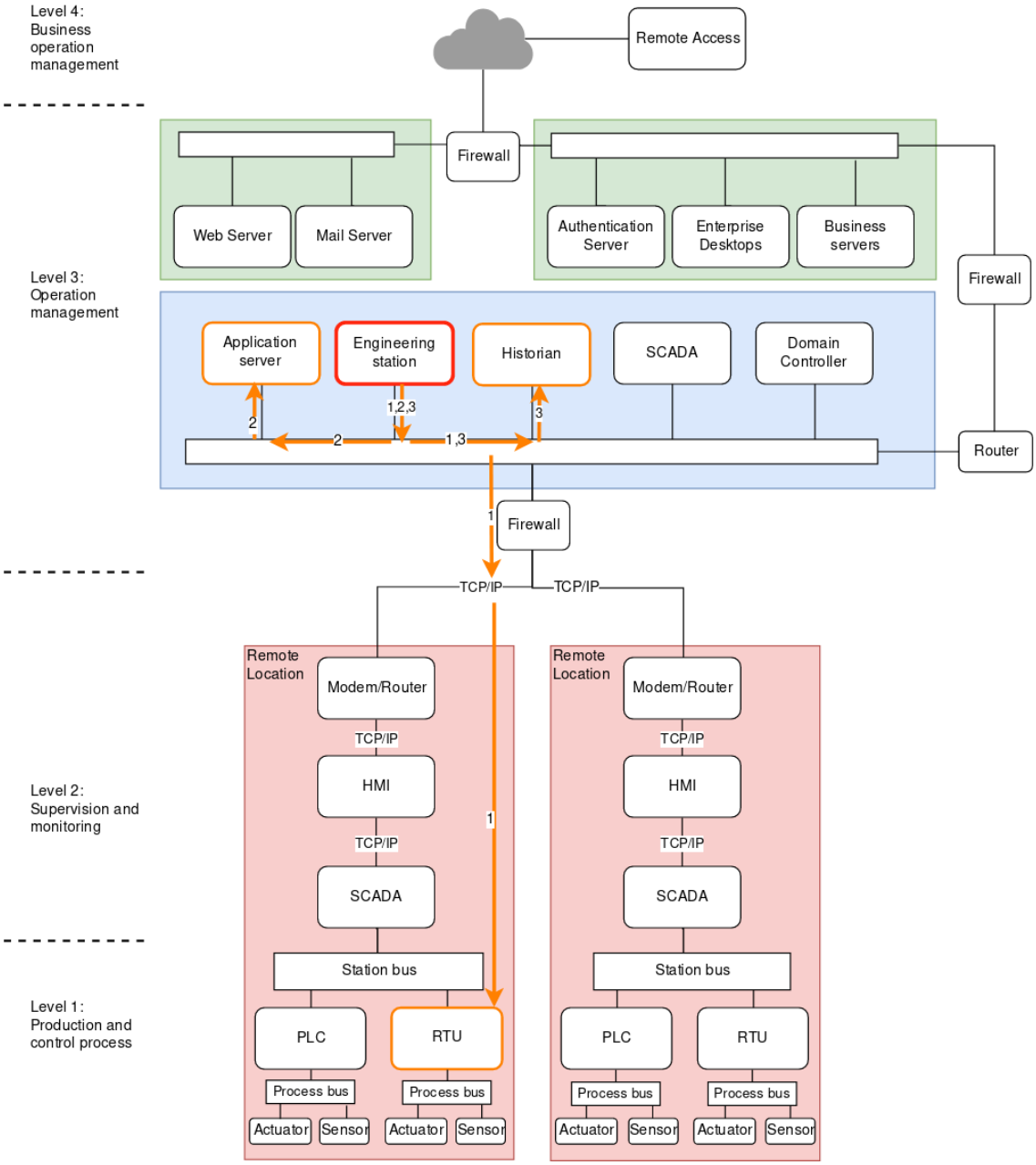

Figure 9: Attack tree of attack scenario, made in colaboration with Niclas Hellesen [6]

5 Incident Handling, Forensics Sensors and Information Sources in Industrial Control Systems

This chapter contains a section for each of the research questions defined in section 1.6. The first section is concerned with the collection of information flow in the ICS. The second section discusses four components related to retaining state after an incident occur. The third and last section is concerned with the aggregation of the collected network traffic, and briefly discusses the possible filtering method.

5.1 Study of control systems architectures to capture control and information flows in a semi-formal model capturing high-level semantics

It exists numerous communication protocols for ICSs, vendor specific, but also open standardized protocols. Because of the diverse amount of protocols, this thesis will not cover them all. One of the most known ICS protocols today is the Modbus protocol. However, because of its simplicity, it does not include a substantial amount of features which can be valuable in an analysis of the network flow. Therefore, the Modbus protocol is put aside in this thesis. A small description of the protocol and other established protocols are placed in section 4.5.

The IEC 60870-5-104 communication protocol can be used between layer 1, 2 and 3 of the reference architecture(described in section 4.1), between what the standard refers to as *controlling* and *controlled* components. The standard IEC 61850 defines a Client/server communication protocol, usually referred to as the MMS protocol. The protocol can also be used between layer 1 and 2 of the reference architecture.

Both of these communication protocols are open, support communication over TCP/IP and can be valuable in an analysis of the network flow. Therefore, these two protocols are covered in more detail through this section. The angle of the description of the two protocols are not the entirely the same, and this is to highlight both how the overall picture of a network packet, but also the information object used in these systems.

5.1.1 IEC 60870-5-104

The -104 protocol is usually used between the controller station in layer 3 and the remote stations in layer 2 of the reference architecture defined in section 4.1.

The -104 protocol is built on the -101 serial communication protocol, as described in the overview chapter, 4.5.7. In the application layer, the -101 protocol contain Application Service Data Units (ASDUs). An ASDU can be separated into two parts where the first part provides the type of data being sent, the number of information objects, the cause of transmission, and the common address of the ASDU. The second part provides the information objects being sent, which include the information

element(s) and optionally an address of the information object and a time tag. With the -104 protocol an Application Protocol Control Information (APCI) header is added to the ASDU. The APCI contain information about the ASDU, such as how long the ASDU is and where the message starts and stop. When the APCI are attached to the ASDU, the full unit is called an Application Protocol Data Unit (APDU). Figure 10, present an overview of this information, the size of each part of the APDU, and how these parts are connected.

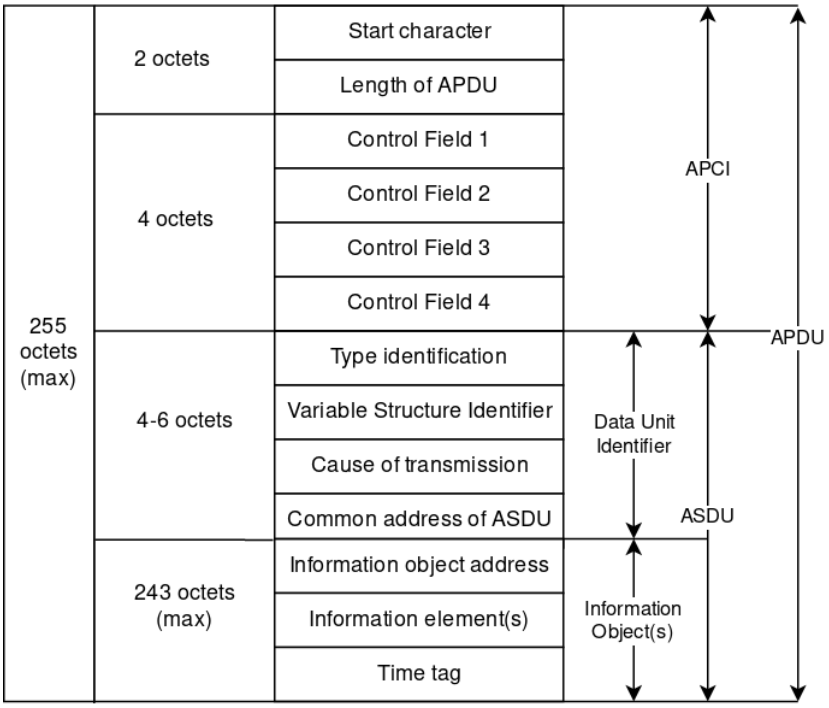

Figure 10: Overview of the APDU of the 60870-5-104 protocol, based on [7]

IEC 62351-5 - Secure Authentication

The IEC 62351-5 include application layer security mechanisms for the -101 and -104 protocol [4]. The standard defines *Secure Authentication*. This security mechanism can, therefore, be implemented next to the -101 and -104 protocol if wanted.

Secure authentication involves the use of a Message Authentication Code (MAC) algorithm. By default, the protocol supports the use of preshared keys to use to the MAC algorithm. However, it does also support the use of symmetric and asymmetric crypto, for example, with the use of asymmetric crypto where each station has its public and private key. The ASDUs are sent between a controlling and controlled station, where both of the stations can initiate the communication. The sender station is usually called the *Responder*, and the receiver is called the *Challenger*.

It exists three different modes for sending and receiving the ASDUs, and these are described in

more detail below.

Non-Critical ASDU

In this mode, the Challenger will immediately process the received ASDU without initiating a challenge. The reason for this is because the Challenger does not find the received ASDU to contain any critical information that must be authenticated.

Critical ASDU

This mode is used if a Challenger finds the received ASDU from a Responder station as critical. The Challenger will then initiate a challenge to authenticate the received ASDU. The challenge involves the Challenger station to create a *Challenge message* which contain the following:

- Challenge Sequence Number (CSQ),
- User Number (USR),
- MAC algorithm (MAL),
- Reason for challenge (RSC),
- Challenge data length (CLN), and the
- Pseudo-random challenge data (CHD).

To authenticate the ASDU, the Responder must answer by including the CSQ, USR, the MAC length, and the MAC value, which is the output value of the chosen MAC algorithm. The MAC value constitute of the *Challenge message*, *Addressing Information*, *Challenge ASDU* and the *Padding data*. When the Challenger has received the response from the Responder and has authenticated the response, the ASDU can be processed. If the authentication challenge is not passed an error message can be transmitted.

Aggressive ASDU

In this mode, the Responder is anticipating that the ASDU is critical. The Responder will, therefore, use a MAC algorithm to calculate the MAC value and include this next to the *CSQ*, *USR* and the critical ASDU in the first transmission to the Challenger. The MAC value will here constitute of the last *Challenge message* received, *Addressing information*, *Authenticated Data* and *Padding Data*. The Challenger can, therefore, authenticate the received ASDU without initiating a challenge. The Responder has sent which type of MAC algorithm which is annotated with the MAL value and the Challenger, therefore, has everything it needs to calculate the MAC Value of the received ASDU. This reduces the bandwidth usage between the two stations. However, it will also reduce the security level, which is discussed below.

Vulnerabilties of the -104 protocol

Without the secure authentication security mechanism, the -104 protocol is vulnerable to numerous attacks. This includes the following attacks: sniffing, man-in-the-middle with modification possibilities, replay, spoofing, and non-repudiation [46, 47].

Vulnerabilities with a IEC 62351-5 implementation

The Secure Authentication concept ensures that an attacker cannot modify the data between a controlling and controlled station without being detected unless the Non-Critical ASDU mode is used. Where the ASDUs aren't authenticated. However, the protocol does not include a solution for encrypting the ASDUs. Therefore, can an attacker be able to eavesdrop unless other security mechanisms are in place, such as Transport Layer Security (TLS) encryption and authentication [48, p. 73]. Replay attacks are mitigated because of the use of Challenge Sequence Numbers and pseudo-random data in each calculation of the MAC Value. Spoofing and non-repudiation attacks are mitigated by the secure authentication challenge between the Responder and Challenger. Where each station holds either a pre-shared key or uses asymmetric crypto where both stations have its pair of private and public keys.

Next, to these mitigations, the standard also mitigates the threat actors ability to eavesdrop when the shared keys are updated.

5.1.2 IEC 61850 Client/server protocol with MMS

The IEC 61850 standard and its client/server protocol were introduced in the overview chapter, section 4.4.4. In this section, this protocol is discussed further, and are form now referenced to as the client/server protocol. Figure 11 below include the parts specified in the IEC 61850 standard and how it maps to the OSI model. On OSI-layer 7, the application layer, the MMS protocol is utilized. Introduction to the MMS protocol can be found in section 4.5.9. Then above the application layer, the Specific Communication Service Mapping (SCSM 2) is utilized. The SCSM 2 contains a specification of how the MMS protocol is mapped to the Abstract Communication Service interface (ACSI) defined in the standard. The next layer containing the ACSI defines the communication between server and clients of the ICS. The upper layer contains the definition of the data objects and classes used in the communication between the nodes.

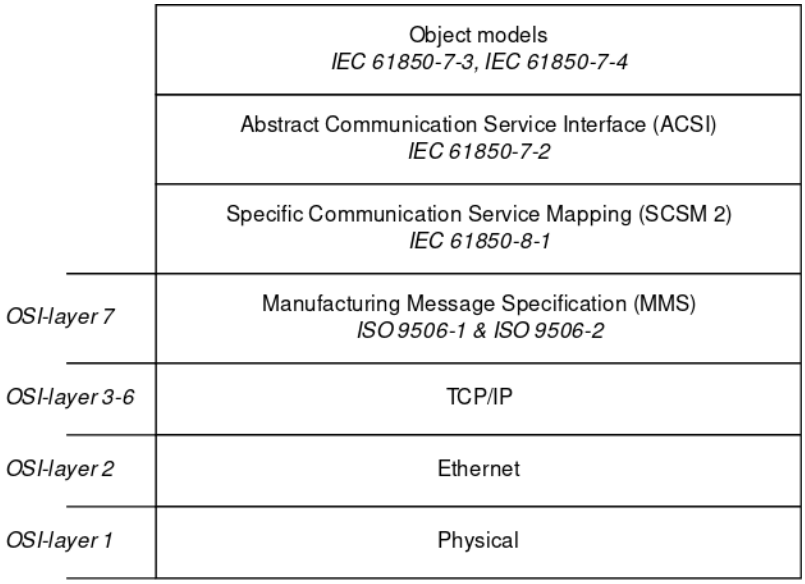

Figure 11: Overview of Client/server 61850 protocol with mapping to the OSI model, based on the IEC 61850 [3]

ACSI

In the standard, the ICS is modeled into information classes [3]. To understand the structure of variables included in data packets that identify a specific components of ICS, these information classes are esential. The object classes consist of the following:

Table 1: Information classes of IEC 61850, based on the IEC 61850-7-2, figure 2 [3]

Class	Description
Server	The server represent the behaviour of a physical component.
Logical Device (LD)	A LD represent the grouping of logical nodes which is associated with the same server.
Logical Node (LN)	A LN can represent one function in a component.
Data object	The data object contain typed information of a LN, such as the timestamp or a set point value.

MMS services

The MMS protocol, which is used on the application layer have multiple services, and some of these were included above. Two of the most used services in the MMS protocol is the *read* and *write* request.

When a *read* service is used the read request must include two arguments, which is defined in section 14.6.1 of the ISO 9506-1 standard [49]. The two arguments are described below.

- *Specification with Result*: The variable is a bool value that will tell if the response of the request should include the value of the *Variable Access Specification* value.
- *Variable Access Specification*: The variable consist of the name of the variable that will be read.

When a *write* service is used the write request must include two arguments described below, which is defined in section 14.7.1 of the ISO 9506 standard [49].

- *Variable Access Specification*: This variable consist of the name of the variable that will be changed, and can be considered as the address of the write request.
- *List of Data*: The variable consist of the new value of the variable specified in above.

SCSM2

In section 5.4 of the IEC 61850-8-1 the mapping of MMS objects and services to the objects of IEC 61850 is defined. Two examples of this is listed below:

- The MMS object *Named Variable Objects* is mapped to the IEC 61850 objects *Logical Nodes and Data*, which is used in the MMS services above
  - *Read*,
  - *Write*,
  - *InformationReport*,
  - *GetVariableAccessAttribute* and
  - *GetNameList* [3].
- The MMS object *Named Variable List Objects* is mapped to the IEC 61850 *Data Sets* object, which is used in the following MMS services:
  - *GetNamedVariableListAttributes*,
  - *GetNameList*,
  - *DefineNamedVariableList*,
  - *DeleteNamedVariableList*,
  - *Read*,
  - *Write*, and
  - *InformationReport* [3].

Each of these MMS objects are mapped to IEC 61850-7-2 services. In section 6.2.1 of the IEC 61850-8-1 these are defined. The *Data Set* object mentioned above are mapped to services such as *GetDataSetValues* and *SetDataSetValues*. (See the standard for the full list)

The *SetDataSetValues* service is mapped to the *write* MMS service. The full mapping of the service can be found in section 14.3.2 of the IEC 61850-8-1 [3]. Above the MMS parameters for the *write* request was presented. In the service mapping the *variableAccessSpecification* is mapped to the *SetDataSetValues* parameter *DataSetReference* attribute and the *lisOfData* MMS parameter is mapped to the *DataAttributeValue[1..n]* of the *SetDataSetValues* service.

IEC 62351 - Security mechanisms

The 61850 protocol does not include any security mechanisms. However, it cites to the IEC 62351 standard, part 4, which is involving security mechanisms for the protocol. The Client/server protocol can implement security mechanisms through TLS.

5.1.3 Network flow on level 1

The IEC 60870-5-104 and IEC 61850 client/server protocol, which introduces this section are primarily not used on a lower level such as level 1. The IEC 61850 standard does, however, define two different protocols that can be found in use on level 1; GOOSE and SMV, which is introduced in the overview chapter. Sensor and actuators can be compatible with multiple communication protocols. Next, to this, there is no clear way of how the sensor or actuator is physically assembled. Modern sensors can be stand-alone devices, or it can be a small component such as a thermometer that requires an intermediary transmitter between the sensor and the controller. The type of sensor used can be strongly dependent on the environment it will be placed in and the distance to the operation management.

It can, therefore, be a problem of listing every possible protocol at this level, and discuss the network flow with its possible vulnerabilities. The list below includes some of the protocols which a sensor or actuator can support. These protocols are described in the overview chapter of this thesis.

- Highway Addressable Remote Transducer (HART) (subsection 4.5.1)
- Foundation Fieldbus (FF) (subsection 4.5.2)
- PROFIBUS PA & DP (subsection 4.5.5)
- Generic Object Oriented Substation Event (GOOSE) (subsection 4.5.8)
- Sampled Measured Values (SMV)
- 4...20 mA and FSK - Frequency Shift Keying (not protocols)

The different transmitters are briefly described with the sensor description in the overview chapter, section 4.3.2. Three different transmitters are described; Field, Head, and Rail Transmitters. All three of the transmitters can transmit data over a current-loop 4..20mA analog wire with or without the HART protocol. Field and Head transmitters can also transfer data with protocols such as Profibus PA, Profibus DP, and FF. Field transmitters can use the wireless protocol called *WirelessHEART*.

Challenges of level 1

The challenge of the diverse use of protocols was briefly discussed earlier. However, it's a significant challenge. From the reference architecture, it's known that the sensors and actuators are placed in level 1 of the ICS, and from the introduction above it's known that the information that flows between the sensor and RTU uses particular protocols which can be complex to monitor at this low level. One way of solving this is to use a sniffer which can aggregate the information to a higher level for analysis. This is discussed further in the next section.

5.1.4 Monitoring of architecture

This subsection presents the proposed architecture for monitoring. The figure below presents the reference architecture with monitoring sensors.

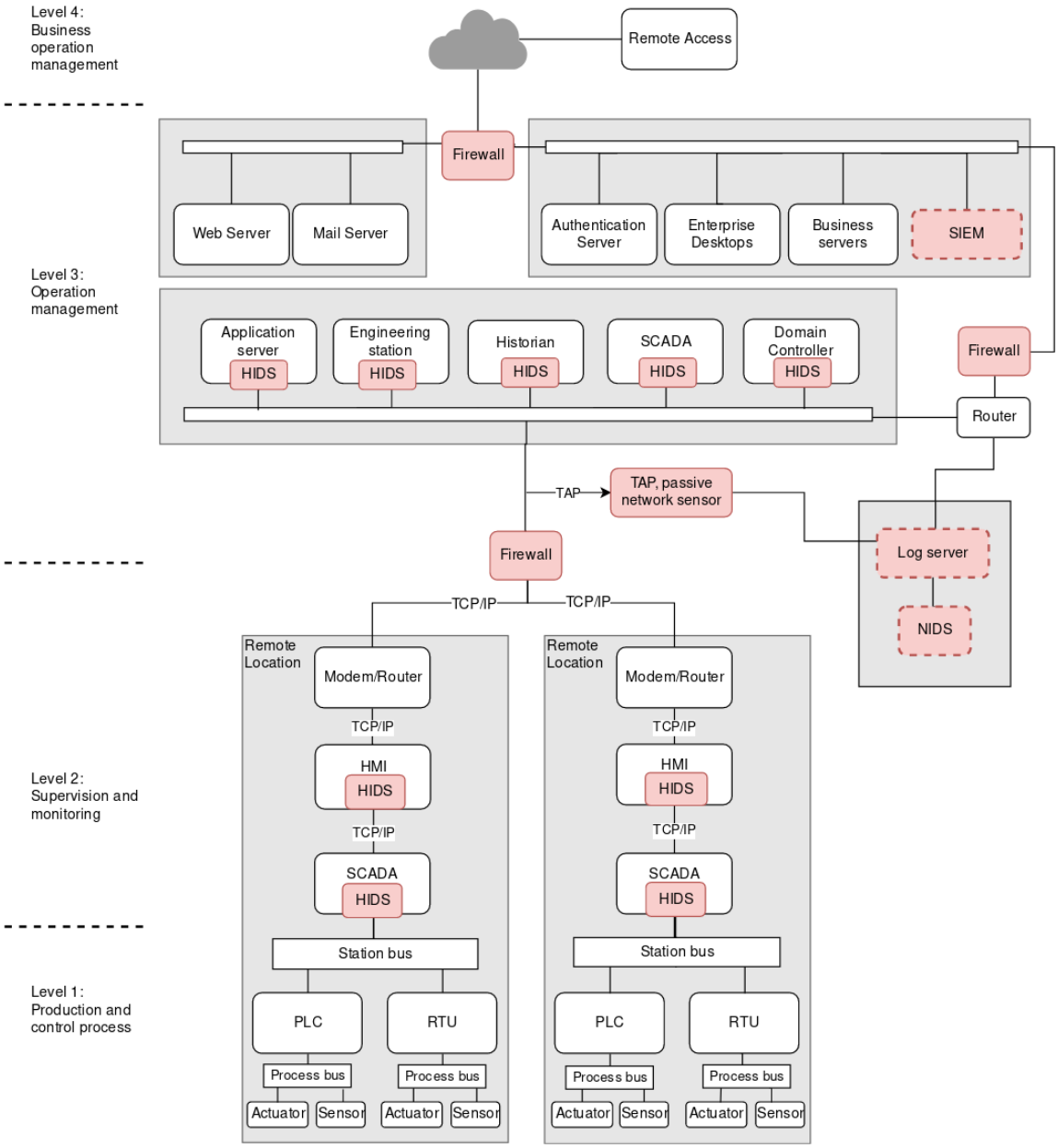

Figure 12: Proposed architecture of monitoring sensors

Firewall

Firewalls are used for controlling incoming and outgoing network packages in the different zones of the ICS. The firewalls need to be configured with a set of rules which will decide if a packet will be dropped or not. In the proposed architecture above a network-based firewall is placed between each of the zones of the ICS, and therefore work as an endpoint security measure of each zone. These firewalls are already in the reference architecture presented in figure 3, and is proposed from the original architecture by ENISA [1].

Control traffic consist of commands from one controller device to another, this can be between zones such as a *write* command from the level 3, EWS, down to the level 2, HMI or SCADA server. A *write* command performs a change to the system configuration, such as changing a temperature limit, or open or close a specific connector in the architecture. Because of the consequence of a *write* request, it is valuable to have a set of rules which can decide if a network packet that contains the command should be dropped or not. To be able to identify which network packet that contains a *write* command, the firewall must be able to interpret each packet on the application level. In section 5.1.1, the architecture of the application layer of the -104 protocol was briefly presented. The ASDU contain a variable named *cause of transmission*, and it's this variable that will tell a firewall if the request is a *write* request or not. In the -104 protocol the write command is called a control command, or more specifically an activation command. Firewall logs are therefore a valuable source of information, for example, if an attacker has tried to request a change to the system and the firewall has dropped the request.

In [23], application layer firewalls are presented and point out the same as stated above about the importance of inspecting packages based on the application layer data. Further, the difference between a *read* verses a *write* request.

Until now, only the use of network-based firewalls has been discussed. However, host-based firewalls are also used in ICSs and should therefore also be considered as a source of information. Host-based firewalls can be found in almost every component of the architecture, but are usually not found in field devices such as sensors and actuators.

Anti Virus

AV software is used for scanning of files for malware, with the use of signatures. Because AV is signature based, it is a necessity to update the signature database regularly such that newer attacks can be detected. Depending on the frequency of the scanning, AV can be resource intensive for a component. Therefore specific field components on layer 1 and 2 must consider the frequency of scanning files and updates to the database. Components which uses universal operating systems (OSs) such as Linux or Windows can use publicly approved AV software, while field devices such as PLCs and RTUs are more troublesome because of the OSs usually are not universal, and the components are more resource constrained than servers.

The events and alarms that the AV software will generate are of interest.

Vulnerability scanners

Vulnerability scanners can be used whenever there is a change in the infrastructure, periodically or when something has happened. Because the field components in level 1 and 2 are sensitive to time and resource constrained, it is advised to be very careful with the use of vulnerability scanners. If a heavy vulnerability scan is used, it may be a risk of one or more components shutting down, which means that some data can be lost. If the low-level components use a vendor-specific or unfamiliar communication protocol, it can be challenging to use vulnerability scanners at this level. However, in level 3 of the architecture, components use universal OSs and communication protocols, are not as resource constrained and time sensitive. A vulnerability scan at or around level 3 will, therefore, generate more useful data and not do any harm to the system.

Network TAP - Sniffes

Sniffers are used for capturing data packets between nodes of the architecture. It is especially interesting to capture traffic that contains multiple features such that more contextual information is in place. The placement of network tap(s)/sniffers must be considered before implementation. Variables such as encryption between components or zones and bandwidth usage are relevant. In the reference architecture above the network, a tap is placed between layer 2 and layer 3 where all of the network traffic between the SCADA control center and remote locations transmit.

Host-based Intrusion Detection System

Host-based Intrusion Detection System (HIDS) software used to monitor an individual component for malicious activity. The HIDS will generate an alarm or event if something malicious is detected [50]. In the reference architecture HIDS can be placed in every server in level 3, and the SCADA servers and HMIs in level 2. The placement of HIDS in HMIs is proposed in multiple works on IDS in ICS [50].

The number of sensors used in the architecture will affect the bandwidth costs. It is therefore essential to consider this before the sensors are placed through the architecture.

Network-based Intrusion Detection System

Network-based Intrusion Detection System (NIDS) is used to monitor the network traffic between two or more nodes in the architecture for malicious traffic.

In the reference architecture, a passive NIDS are placed between the control center in level 3 and the remote stations of level 2. A passive NIDS is used because of the time sensitivity of the SCADA architecture.

It exists multiple NIDS software on the market; however, it is a major shortcoming of NIDS software which supports communication protocols specific for ICS. This is, therefore considered a challenge. The NIDS Suricata supports two ICS communication protocols when this thesis is written, the DNP3 and Modbus communication protocol [51].

Security information and event management

A Security Information and Event Management (SIEM) is used to analyze all the real-time events and alarms other monitoring sensors in the infrastructure has generated. The SIEM is configured

with specific rules for detecting compound events, and therefore when a rule is met, an event is generated by the SIEM.

The SIEM can be placed in multiple areas of the reference architecture and is therefore considered as a logical component. The placement of the SIEM will depend on the monitoring sensors and the log server(s) in the architecture, but also variables such as bandwidth costs and security. The SIEM can be placed in a the SCADA network in level 3, or it can be placed in a separate demilitarized zone (DMZ) with or without a log server in level 3.

Today it exists multiple SIEMs. However, not many of these have support for SCADA protocols and operators, therefore, need to write their own rules to be able to detect events. Splunk is a software product able to work as a SIEM. It exist a plugin for Splunk called *kepware* that work as a *data forwarder* between the protocol data and Splunk [52].

Log server

The log server is a logical component. By this, it means that one or multiple log server(s) can be placed in numerous positions in the architecture. The log server stores data from the monitoring sensors of the infrastructure. In figure 12 the log server is placed in a separate network zone. However, the placement of the log server can also be in the SCADA network in level 3, in the same zone as the SIEM or it can be placed in a separate DMZ between the SCADA and operational business zone in level 3. The remote stations are a possible location for a secondary log server.

It exists advantages with the usage of both one centralized log server and multiple log servers. In [53], R. Bejtlich highlights the benefit of retrieving only logs from one log server instead of numerous during an incident, and how an attacker must be able to compromise both the log server and the actual component to completely hide their traces, if not it will exist reliable information in either the component logs or in the log server. Bejtlich also, includes for an attacker the log server is a higher target than a single component. Therefore, the security of the log server should be considered.

Next, to storing events from the sensors, a log server can also be used to correlate events from the architecture. In [54], Lemay, Sadighian and Fernandez has integrated journaling for SCADA systems talking the Modbus protocol by correlating the events of the network. This is done by correlating controller commands sent from an operational workstation to a controller component, the identified user logged into the windows system and the sender IP address of the controller command.

5.2 Identification of entities retaining relevant state for incident response and analysis within said architecture

The components of the architecture contain information which can be relevant if an incident occurs. It is therefore essential to know which of the components that can retain relevant state for analysis during or after an incident occur. In this section, the four components; thermal sensor, RTU, HMI, and the SCADA server is discussed in more detail concerning the state of the data storage of the different components.

5.2.1 Thermal sensor

A thermal sensor exists in different formats, from small analog sensors that output direct signals to more sophisticated digital sensors that output in wireless protocol formats such as WirelessHART discussed in the previous section. In this subsection, digital thermal sensors are the focus. These type of sensors can output a range of different protocols; HART, FF, and Profibus PA.

Architecture

An architectural overview of a digital sensor is included below, in figure 13. The figure is mainly based on the four references, [8, 9, 10, 11]. The digital temperature sensor consists of a converter between analog and digital signals, a CPU, RAM, EEPROM, physical buttons, an LCD indicator, power supply, input and output ports. As indicated in the introduction to this subsection, thermal sensors exist in different formats, and this is also the case for digital thermal sensors. e.g., not all digital sensors have an LCD indicator and physical buttons.

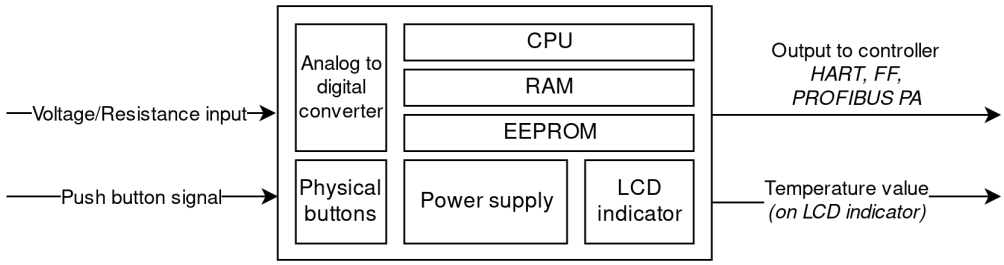

Figure 13: Architectural overview of a digital thermometer sensor, based on [8, 9, 10, 11]

To be able to collect data from the thermal sensor during or after an incident, its information sources should be identified.

Memory

The digital temperature sensor consists of volatile and non-volatile memory, where the volatile memory unit format is *Random-Access Memory (RAM)* and the non-volatile memory unit used is the *Electrically Erasable Programmable Read-Only Memory (EEPROM)*.

EEPROM is used to store collected data from the sensor, and the firmware of the sensor. EEPROM

can only hold a small amount of data. However, the sensor will retain the data stored in EEPROM after a power outage. This is not the case with the data stored in RAM. This data will be lost after a power outage takes place. In [10] this is highlighted.

As mentioned is the sensor discussed in this section a digital sensor; however, as stated, it also exists sensors that are analog and communicate over an analog wire. These types of sensors should not be considered as a location for relevant information after an incident has occurred since it does not contain any digital memory.

5.2.2 Remote Terminal Unit

A description of a RTU are included in section 4.3.4. In this section the architecture and the Memory capabilities of the RTU is discussed.

Architecture

A RTU consist of analog input and output module, a digital input and output module, a CPU, non-volatile memory such as RAM, volatile memory, a programming interface, status lights, power supply and optionally a UMTS modem. This is summarized in figure 14 below.

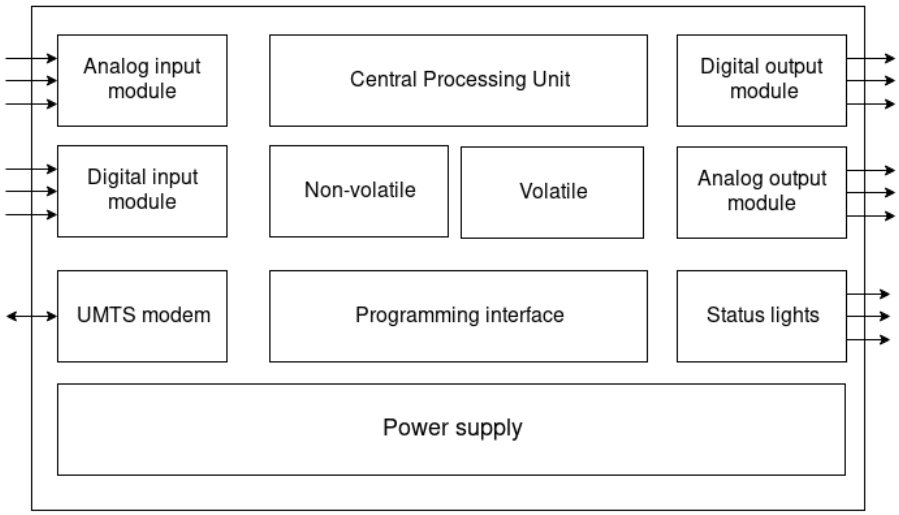

Figure 14: Architectural overview of an RTU, based on [12, 13]

Memory

In [13], Ahmed, Obermeier, Sudhakaran, and Roussev, perform a forensic analysis of two different PLCs. Because of the similarities between a RTU and PLC this paper is of relevance also for a RTU. In this analysis, both volatile and non-volatile data was extracted from the PLC. The results showed that the two different PLCs had different architecture, where one of these stored everything except the firmware in RAM. Therefore, if the type of PLC were turned off, relevant information such as

the file system and log files would be lost. This PLC require a forensic analysis without turning the PLC off.

During the thesis period, an interview with a power utility company was conducted. From this interview, it was informed that this is an issue with older PLCs. Even though it appears to only apply for older PLCs, doesn't mean that it's no longer an issue. From the interview, it was learned that organizations today still have these PLCs in their network. However, they're being faced out. This problem highlights the importance of knowing the architecture of the PLC, and other components of the architecture, such as HMIs and RTUs.

5.2.3 Human Machine Interface

The Human Machine Interface (HMI) was introduced in section 4.3.5. The hardware architecture and memory capabilities of the HMI are discussed further in this section. Technical information of the HMI is based on vendor's data sheets, such as [55, 56, 57].

Architecture

The HMI consist of digital input and output module, a CPU, a monitor/screen, non-volatile memory, volatile memory, physical buttons and or a touch screen and a power supply. All of these hardware components are included in figure 15 below.

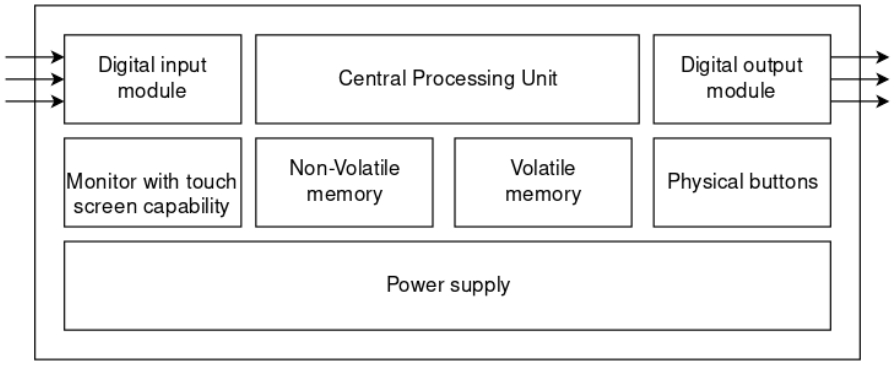

Figure 15: Architectural overview of an HMI

The HMI is used to control and monitor the physical process. It has a monitor which will display real-time process data from the field devices, such as temperature values from a sensor or generated alarms from a component. With buttons and or touch screen, the HMI is used to make control changes to the configuration of the field devices.

Memory

The HMI consist of volatile, non-volatile, and external memory storage. Where the non-volatile memory is flash storage, which ranges in capability from 12MB up to 4GB depending on HMI. The volatile memory is RAM; here also, the range of capability depends on the model of HMI. Next, to

this, the HMI has an input slot for SD cards.

The OS of the HMI depends on the version. OSs such as Windows Embedded and RTLinux is used on these devices. Vendor-specific software usually runs on HMIs to be able to communicate with other components and to be able to monitor the physical process data and control the field devices.

5.2.4 Supervisory Control And Data Acquisition server

A SCADA server or an operator station is a component which can be used by the operator or engineer to get an overview of the state of the system and make small configuration changes to the system components. The SCADA server was briefly introduced in the overview chapter, section 4.3.6. A SCADA server usually consist of a PC running a Windows OS with vendor-specific software for operator stations.

Memory

The SCADA server has volatile, non-volatile memory and the possibility for external memory storage. The server can be considered to have the same storage capacity as a workstation at level 3 and 4 of the reference architecture.

Historical data and logs

The SCADA server run vendor-specific software which generates files which can be viewed by a user on the computer [58, p.326]. These files can be collected after a reboot of the server and usually consist of the following:

- System logs such as events with logged in user and timestamp
- Firmware information
- Session information
- Startlog of controller components
- Crash logs

IEC 61850 descriptive configuration files

In part 6 of the IEC 61850 standard, the System Configuration description Language (SCL) is defined [3]. This language is XML based and is used in files containing the configuration of the components, the structure, and placements of the information objects and networking specifications. These files are referred to as *SCL configuration files* and are human readable files, but can also be parsed with an XML parser for easier understanding.

5.3 Identification of network flows augmenting or corroborating information and control flows

The use of a network tap as a monitoring sensor was included in the proposed monitoring architecture described in section 5.1.4. The location of the network tap was recommended between the operator station at level 3 and the remote stations at level 2, of the reference architecture presented in section 4.1. With this location, the network tap collects a significant amount of data. To be able to analyze the data for augmenting and corroborating information and control flows, the captured data must be aggregated to a centralized location and filtered such that only relevant network packages are kept.

5.3.1 Considerations of Aggregation of information

To be able to analyze the collected network data, all of the data need to be aggregated to a centralized server. The aggregation is especially necessary if a network tap has been placed in multiple positions of the architecture. Such that all of the collected data can be analyzed and put together in a broader context. The centralized server needs to have the computational capacity of conducting real-time analysis and filtering of the collected data.

The placement of the server(s) that performs the analysis of the network traffic needs to be considered. It is several points that should be considered before the location of a server is decided. Which include the following points:

Computational capacity

The computation capacity of the aggregation and analysis of real-time network data should be considered such that the server is capable of handling the amount of data.

Bandwidth

The bandwidth capacity can be different from the different levels of the reference architecture. Therefore, when the logs are being aggregated to the central server, the bandwidth capacity is important. If it exists a line in the architecture that does not tolerate high levels of network flow, the central server should not be placed in this part of the architecture.

Storage capacity

The amount of data that are collected in a monitoring system are very large [23, p.382]. Therefore, the storage capability of the server(s) must be considered.

Placement of further processing device

The placement of the central server that will process the captured network flow needs to send the results to another component that will further analyze the results or display the results to an operator such as a SIEM.

Security considerations

A log server or a server that process and analyze this type of information can be objects of interest for an attacker. Therefore, considerations such as placing the server into a dedicated zone or access control should be considered.

5.3.2 Aggregation of information in the reference architecture

In the last section, relevant aspects of aggregation of data were discussed. In this section, the aggregation of data is discussed with empathizing on the monitoring architecture presented in section 5.1.4.

A network tap is placed between layer 2 and 3 of the reference architecture. The network tap can be placed at other locations, as discussed in section 5.1.4 as well. However, now the tap is only added between level 2 and 3. The data is collected and stored in a log server between level 2 and 3. The location of this log server is not optimal when operators need to access the information. Access to the SCADA part of level 3 should be very restricted to only operators that are controlling the ICS. Therefore, the collected network data should be aggregated to a higher level in the architecture. However, another aspect of deciding the placement is that the data contain a lot of sensitive data for the business, and it should, therefore, not be accessible for every employee in the organization. A log server and an analyzer are therefore placed in its dedicated zone between the SCADA zone and the organizational management zone. A figure that illustrates this description is included below in figure 16.

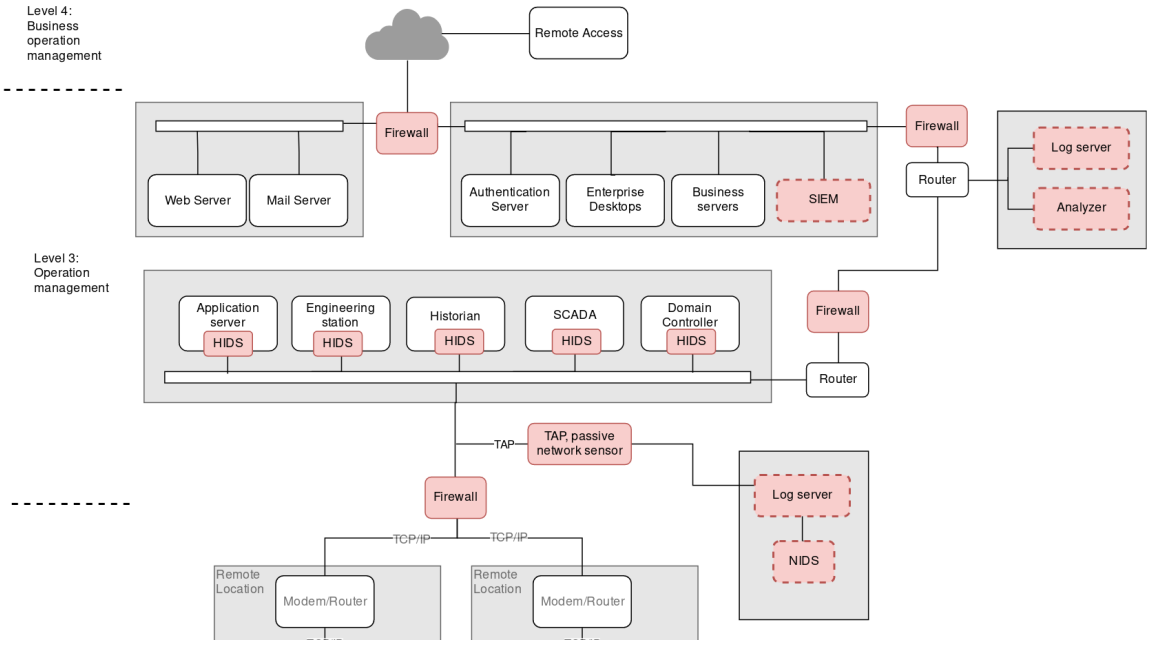

Figure 16: Overview of monitoring architecture, including the placement of server used for pre-processing of collected network data.

A discussion regarding the aggregation of information was done in the unstructured interview with a utility company. In this interview, it emerged that the company aggregated collected network

traffic to a DMZ zone in level 3.

5.3.3 Filtering of collected network traffic

In the section above an analyzer with a log server was placed into the reference architecture to be able to analyze the collected network traffic between the operation station and the remote stations. In this section the type of analyzing method for the network flow is discussed.

Since an ICS generate big amounts of data, it can be a challenge to identify unnormal behaviour of the system. Different techinques for analyzing network flow in SCADA networks have been conducted in earlier research [31, 32, 59, 33]. These focuses on anomaly detection, where the normal peridodic behaviour of the system is used to be able to identify behaviour in the network traffic which is not normal. Unnormal behviour can be the identification of unwanted events or it can be just noise from the system. In [31], two important characteristics are used to analyze the captured network traffic. The frequency of packages and the size of each data package are used in the analysis.

In section 2.5 data forms used in NSM was described; one of these data forms was metadata and statistical data. In this section, metadata and statistical data are discussed further. However, metadata discussed in this section is more related to the data that already exists in the collected network traffic, such as size, timestamps, sender, and receiver IP and port.

To be able to analyze the collected network traffic to identify data packets that can be interesting an anomaly detection method can be used. Where statistical data such as frequency of data packets, and the metadata such as the size can be considered. Which is done in the research papers listed above. However, the use of a different protocol must be analyzed to check if it fits or not.

6 Physical testing

In the last chapter, different monitoring sensors were proposed to be able to capture the control and information flow of the ICS. In this chapter, a physical lab is used to test the network tap that was presented as a monitoring sensor for the architecture, and the data from this test is analyzed to map to the IEC 61850 Client/server communication protocol. This is done by executing the attack scenario defined in section 6.2 below.

6.1 Lab setup

A remote station was used as the base for the lab setup. The full setup is included in figure 17 below.

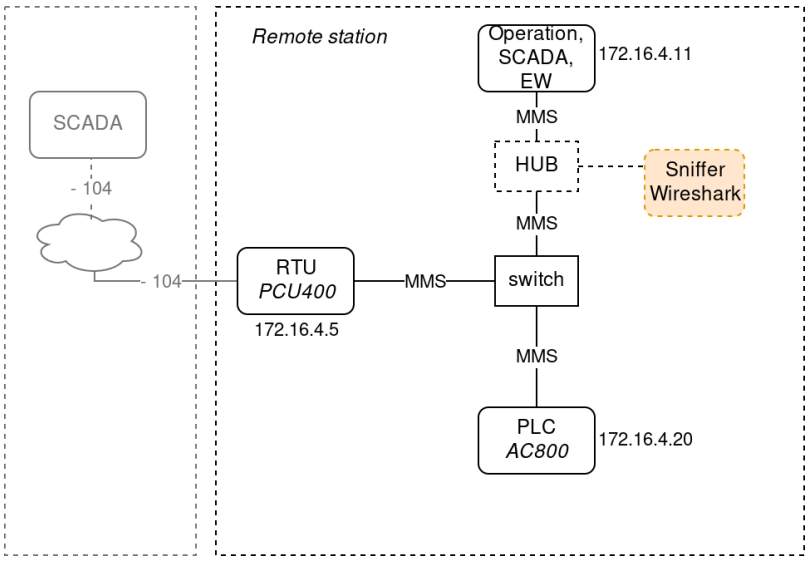

Figure 17: Lab setup of remote station for testing phase

The lab consists of a RTU which serves as a gateway for remote configuration, a PLC which serves as a slave, an operator workstation that serves as a master, and networking equipment consisting of a switch to connect the components. A hub and a PC running Wireshark are used to capture the traffic between the operator PC and the PLC.

Protocols

The MMS protocol, described in section 4.5.9 and 5.1.2, is used as the base client/server communication protocol between the components. The RTU communicate over the -104 protocol, described

in 4.5.7 and 5.1.1, out to the level 3 controller station's SCADA server. The physical lab does not include the actual level 3 station, however, it is included in the figure to illustrate the full ICS.

Components & vendor

The RTU is a *PCU400* model from ABB, the PLC is a *AC800* model from ABB, and the operator station also run ABB software. The operator station include both software for an engineering workstation and SCADA server. The name of the software is *Engineering Station*, *Operator station*, *OPC server* and *Control Builder*. The PC uses Fedora as OS and contain Wireshark version 3.0.1.

6.2 Attack Scenario 1 - Physical damage

An attack scenario was created in collaboration with Niclas Hellesen as described in section 4.6. In the planning of this thesis, this attack scenario was proposed as the base for the testing of this thesis' and later studies of the same project. The equipment that was necessary to use this attack scenario was not available at the time of the work of this thesis, and therefore, the attack scenario was reshaped to fit the lab environment. An attack is described in this section, which is based on the joint attack scenario. Both of the attacks involve a threat actor changing the value of a variable in the configuration of a RTU.

6.2.1 Attack narrative

The attack concerns a disgruntled employee that wants to create damage to its employer by creating physical damage at a remote station. The employee has physical access to a remote station and decides to use this as an entry point. With the operator station at the remote station, the employee applies a change to the configuration of the PLC. The configuration change involves an alarm limit for a temperature sensor, such that when the physical process reaches a breaking point the sensor will not output an alarm to the operator at the control station at level 3 of the reference architecture.

6.2.2 Attack Tree

This subsection presents the attack in an attack tree, in figure 18. The root node of the attack is the objective of the attack, each square node represent a step of the attack scenario, and the hexagon represents the detection method of the attack step.

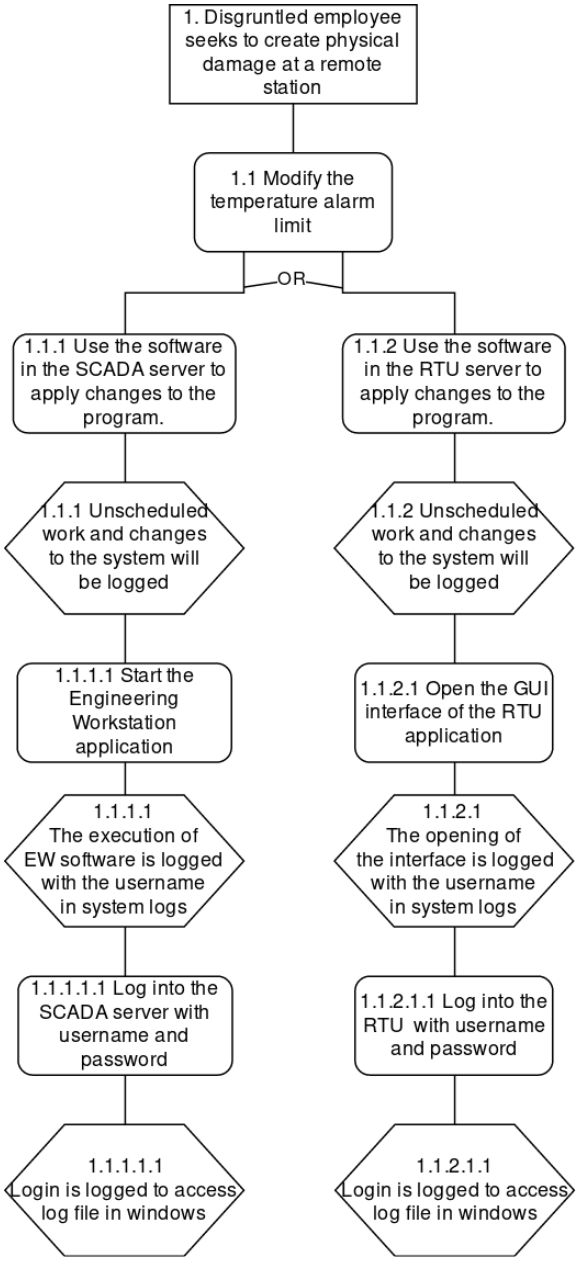

Figure 18: Attack tree of attack scenario 1

The attack tree presents two different ways of achieving the attacker's objective. In the first alternative the disgruntled employee logs into the operator station, open the *Control Builder* software, identify the correct sensor in the architecture and apply a change of the *max* variable of the alarm

limit of the temperature sensor. When the attacker has fulfilled this a *write* request should have been sent from the operator station to the PLC and the PLC have answered with a *success* response.

The second alternative is equal to the first. However, the disgruntled employee chooses the RTU as an entry point instead and uses the interface on the RTU to conduct the attack.

6.2.3 Attack steps in the lab

To be able to simulate the attack presented in the section above, a set of steps was made.

1. Log into the control station
2. Open the *Control Builder* software and the *Engineering Station*
3. Find and open the temperature sensor wanted
4. Apply a change to the *max* variable of the alarm limit of the sensor. Make the new value of *max* to *750* degrees Celsius.
5. Open up the view of the engineering station to see the change.

6.2.4 The objective of the attack

The objective of this attack scenario is to check the possibilities of capturing the *write* command sent over the network and analyze the transparency of the captured packet. With this is meant, if it's possible to identify what type of command that was sent, the variable of a *write* command, which function or component that is the receiver of the command and the source of the command.

The operator station is also checked for systems event logs that can identify which user that logged into the operator station and open the application of the station.

6.3 Analysis of the attack

6.3.1 Analysis of captured traffic

The hub that was placed between the operator station and the switch forwarded all network traffic to the pc running Wireshark simulating a network tap. From the packet capture, the request that was sent from the operator station to the PLC with the *write* request was identified. The packet is included in figure 19 below. The figure presents the MMS structure of the message and the full hex & ASCII content of the packet through the use of Wireshark.

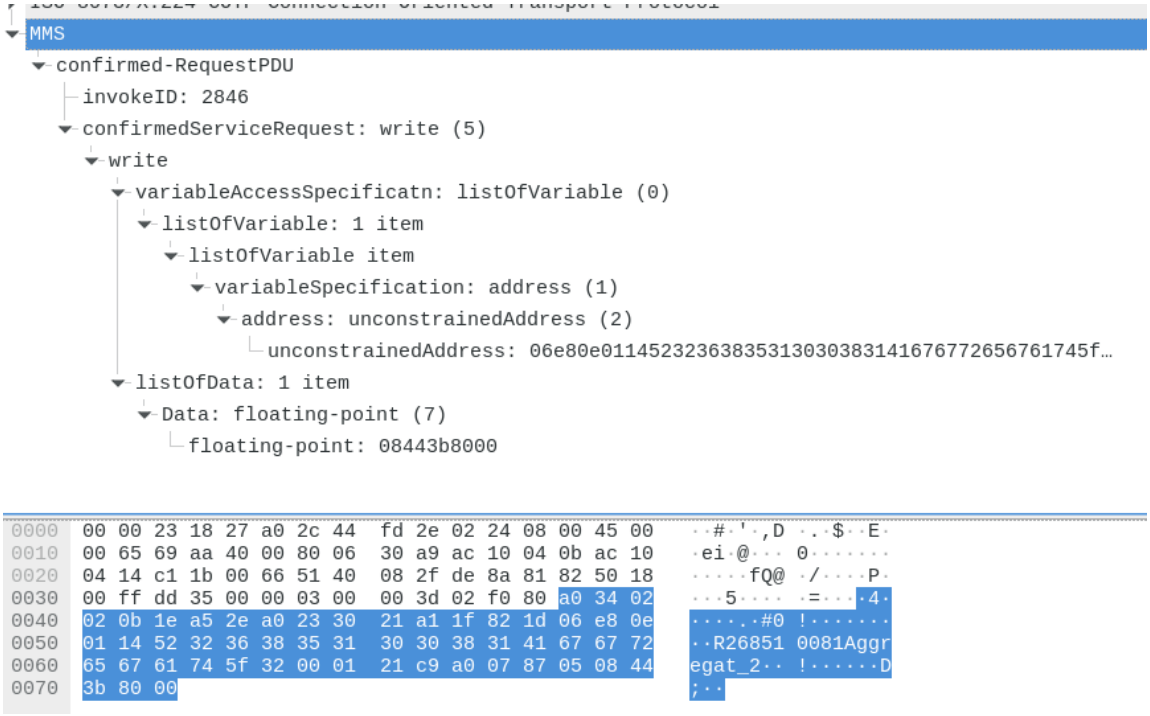

Figure 19: Captured packet of writing request from operator station to PLC

From figure 19 above the use of the MMS application layer protocol is present. The request is of the type *confirmed-RequestPDU* and the MMS service *write* is used. In section 5.1.2 the parameters of the *write* service was presented. The *write* request consist of the two parameters; *Variable Access Specification* and *List Of Data*. The first varible consist of a *unconstrainedAddress* and the second variable consist of a *Floating-point*. The two parameters are discussed in more detail in the following two parts.

unconstrained Address

The *unconstrainedAddress* is the content of the *variableAccessSpecification* as pointed out above. From section 5.1.2, it's known that the variable contain the address to the variable that will be changed. In table 6.3.1 below, the value of the *unconstrainedAddress* variable is included in both hex format and in ASCII format.

Table 2: The unconstrained Address value of a write request form collected network traffic.

Type of format	Value
HEX	*06e80e011452323638353130303831416767726567617 45f32000121c9*
ASCII	*.e...R268510081Aggregat_2..!E*

The ASCII representation of the value confirms that the *unconstrainedAddress* variable contain the address of the variable that is changed.

Floating-Point

The next parameter, the *Floating-Point* is the content of the *listOfData* variable. The content of the *floating-point* is presented in table 6.3.1 below. In the figure both the IEC 754 hex representation value of the variable and the decimal value is included.

Table 3: The floating-point variable value of a write request form collected network traffic.

Type of format	Value
IEC 754 hex	*08443b8000*
decimal	*750*

The decimal value representation of the value shows that the variable contains the new value of the variable addressed to in the variable above. In the attack scenario, the threat actor changed the value of a variable containing the max temperature of a thermometer, the value set in the attack scenario was 750. Therefore, the output from the network packet above reflects the action of the threat actor.

6.3.2 Access logs

In 5.2.4, the different log files that can be collected from a SCADA server was presented. The access log file was one of these and was of interest for this analysis.

In the event properties of Windows, the security logs were located. Here events such as login attempts were logged. Therefore, after the attack, these logs where analyzed and the results are included in figure 20 below. The figure shows the username that the attacker used to gain access to the system, the timestamp of the login and that the login attempt was successful.

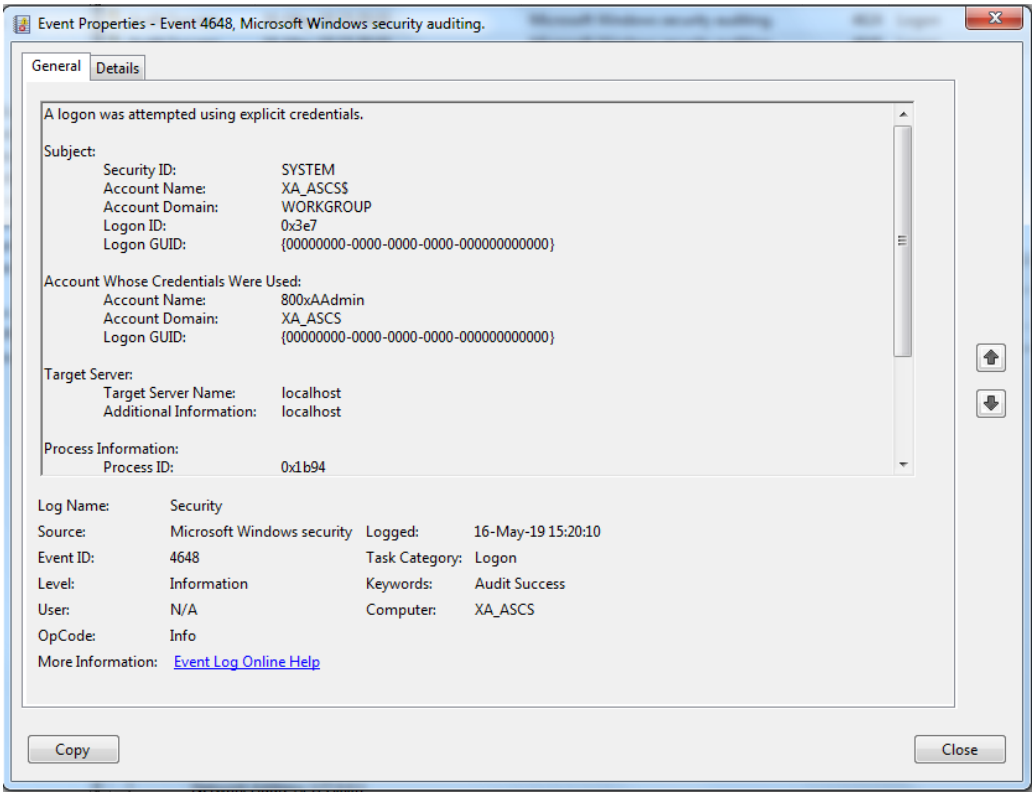

Figure 20: System log event of user login success

With the information in the figure above, the username and the timestamp could be used further to analyze system logs to identify which application the user opened. This was not done in this lab, but it's a possible step. Though this was not done, the user that logged in to the system has been identified with this log. If the username that was used for logging into the system were not connected to one specific person, the analysis of the log would not be efficient to identify the attacker.

7 Discussion

To be able to have real-time awareness over an ICS, it's essential to understand what is going on in each level of the architecture, between each component and on the component on its own. To be able to achieve this it's a need to collect and analyze the information flow in the system, but also have an understanding of the operation of the physical process that the ICS is controlling.

The overreaching objective of this research was to create a semi-formal model for monitoring purposes of ICS, identify components in the ICS that retain a relevant state after an event or incident has occurred, and lastly discuss the ability to identify augmenting and corroborating information from network flow.

The outcome of the study was a presentation of some of the protocols used in an ICS, which was the groundwork/discussion for the proposed monitoring architecture. Next, to the monitoring sensors which can analyze and aggregate information in real-time, the four components sensor, RTU, HMI, and SCADA server was described to give an understanding of which component that would be interesting in forensic analysis. The proposed monitoring architecture used a network tap to capture network flow between the operator station and the remote stations of the ICS. The need for aggregating this information was managed by a small discussion of the considerations that should be taken, and also a filtering method for the network traffic was proposed. However, the filtering of network flow did not get the full attention it needed before the delivery date and it's therefore also added into future work. A little lab experiment was conducted to analyze 61850 client/server protocol traffic to analyze how much information that could be collected from network traffic.

The research study had a few limitations. Specifically, when it came to the use of the physical lab. In the planning of the project, it was established that the use of a realistic to real-life lab environment was to be used for the physical testing part of the project. This plan fell through, and a lab environment close to a remote station was used instead. Therefore, it was not possible to conduct a full analysis of the whole architecture.

Next, to this, it is relevant to comment that a reference architecture was defined and used as a reference for an ICS through the project. Therefore, it can be small differences between the results of this study next to other ICS architectures. However, in the decision of the reference architecture, various architectures form different sources where analyzed, and the conclusion was that there is no standardized architecture for ICS. Each architecture depends on different factors of the organization such as eth environment, the physical process to control and location of the stations.

The last limitation, but also an important one, is the researcher's lack of experience from work in ICSs. Experience from the field is vital to understand the processes in an ICS. Both from an engineers point of view, but also from an operator's point, where knowledge about the system, the

process and the information flow between the components get essential. As discussed in earlier chapters, one problem is the use of multiple communication protocols. Beginners in this field can easily get overwhelmed with the number of different protocols and the operational understanding of what is normal and unnormal in ICSs.

From the theoretical parts of the study, it was discussed the problem of interpreting captured network flow. The IEC 60870-5-104 protocol and the Client/server protocol of IEC 61850 were presented, and a lab experiment using the Client/server protocol was conducted. The results from this showed the problem of interpreting the full context of each network packet.

Various research papers have been cited concerning analyzing and monitoring ICSs. Most of these focus on the use of Modbus protocols and the DNP3 protocol. This research study presents protocols at different levels of the architecture, including protocols used between sensors and controllers such as HART. Further, it presents the use of the IEC 61850 Client/server protocol and IEC 60870-5-104 at higher levels. This gives a new interpretation of systems utilizing these protocols and monitoring sensors. The study proposes a monitoring architecture of the ENISA reference architecture. The monitoring architecture contributes to a broader architecture for monitoring of ICSs.

The study needs more work on the filtering part of the network flow. Further, an analysis of the Client/server protocol and captured network traffic must be conducted to map the protocol to collected network traffic in more detail. This is specifically important when it comes to being able to identify the receiver in the ICS of a command. In the lab experiment in this study, it was not possible to determine precisely which function or variable that was changed. The author of this thesis will not work on related work at this time. However, it is up to the community and the overall project at NTNU to go further with this process.

8 Conclusion

The main contribution of this research study is the proposed monitoring architecture for ICS. The IEC 60870-5-104 protocol and the Client/server protocol of the IEC 61850 standard have been presented in more detail to understand the network flow between components with the monitoring sensors. Further, the research highlights four components in the reference architecture to discuss the architecture of the components, which can say something about the state if an incident occurs. Lastly, the aggregation and filtering of collected network flow are proposed for the proposed monitoring architecture. Therefore, the contribution of this thesis is a model for collecting and analyzing information flow in a ICS.

The model for monitoring in this research study will hopefully bring more awareness around the process of monitoring critical infrastructure, and further studies on the IEC 61850 client/server protocol. And also, bring more motivation for the operators to know the architecture of each component, such that they know which component that is relevant for analysis if something happens.

8.1 Future Work

This study is a part of a bigger project concerning a Cyber Situational Security Awareness Architecture for Industrial Control Systems, as stated earlier. The future work is concerned with the pre-processing of sensor data that are used further in the architecture. This includes the normalization of the collected data from all of the monitoring sensors placed in the proposed monitoring architecture.

Besides the complete project mentioned above, this research paper on its own brings light to future work that should be discussed on its own. Especially the analysis of captured network traffic should be evaluated in more depth to the IEC 61850 Client/server and the IEC 60870-5-101/104 communication protocols. As mentioned in the discussion, most research studies focus on other protocols like Modbus and DNP3, and it's, therefore, a need to find what normal behavior can be categorized as in the IEC 60870-5-104 protocol and Client/server protocol of IEC 61850 standard.

Further, a lab experiment should be conducted to ensure that all relevant data are collected with the proposed monitoring architecture. The architecture needs to be validated by a physical experiment.

Bibliography

[1] Communication network dependencies for ics/scada systems. Technical report, European Union Agency for Network and Information Security (ENISA), 2 2017.

[2] Waagsnes, H. Scada intrusion detection system test framework. Master's thesis, University of Agder, 9 2017.

[3] Iec 61850 - communication networks and systems for power utility automation. Standard, International Electrotechnical Commission (IEC), 09 2011.

[4] Iec 62351 - power systems management and associated information exchange - data and communications security. Standard, International Electrotechnical Commission (IEC), 05 2007.

[5] HART Communication Foundation. Hart communication application guide. Last accessed 24.05.19. URL: https://www.fieldcommgroup.org/sites/default/files/technologies/hart/ApplicationGuide_r7.1.pdf.

[6] Hellesen, N. Cyber situational security awareness architecture (cssa) for industrial control systems. Master's thesis, Norwegian University of Science and Technology (NTNU), Gjovik, 05 2019. Not published.

[7] Iec 60870 - telecontrol equipment and systems. Standard, International Electrotechnical Commission (IEC), 06 2006.

[8] ABB Automation Products GmbH. 2013. Industrial temperature measurement basics and practice. URL: http://search-ext.abb.com/library/Download.aspx?DocumentID=03/TEMP-EN&LanguageCode=en&DocumentPartId=&Action=Launch.

[9] Bolton, W. 2015. Chapter 10 - microprocessors and microcontrollers. In *Mechatronics Electronic control systems in mechanical and electrical engineering*, Bolton, W., ed, 241 – 277. Pearson, sixth edition edition.

[10] Smith, D. W. 2013. Chapter 13 - eeprom data memory. In *PIC Projects and Applications using C (Third Edition)*, Smith, D. W., ed, 135 – 138. Newnes, Oxford, third edition edition. URL: http://www.sciencedirect.com/science/article/pii/B9780080971513000131, doi:https://doi.org/10.1016/B978-0-08-097151-3.00013-1.

[11] van der Knijff, R. 2014. Control systems/scada forensics, what's the difference? *Digital Investigation*, 11(3), 160 – 174. Special Issue: Embedded Forensics. URL: http://www.sciencedirect.com/science/article/pii/S1742287614000814, doi:https://doi.org/10.1016/j.diin.2014.06.007.

[12] Clarke, G., Reynders, D., & Wright, E. 2003. 2 - fundamentals of scada communications. In *Practical Modern SCADA Protocols*, Clarke, G., Reynders, D., & Wright, E., eds, 12 – 62. Newnes, Oxford. URL: http://www.sciencedirect.com/science/article/pii/B9780750657990500188, doi:https://doi.org/10.1016/B978-075065799-0/50018-8.

[13] Ahmed, I., Obermeier, S., Sudhakaran, S., & Roussev, V. November 2017. Programmable logic controller forensics. *IEEE Security Privacy*, 15(6), 18–24. doi:10.1109/MSP.2017.4251102.

[14] Whitman, M. E., Mattord, H. J., & Green, A. 2014. *Principles of Incident Response & Disaster Recovery*. Cengage Learning, 2 edition.

[15] Threat landscape for industrial automation systems. Technical report, Kaspersky Lab ICS CERT, Sep 2018. URL: https://ics-cert.kaspersky.com/media/H1_2018_ICS_REPORT_ENG.pdf.

[16] Internet security threat report. Technical report, Symantec, 4 2018. URL: https://www.symantec.com/content/dam/symantec/docs/reports/istr-23-2018-en.pdf.

[17] Lopez, J., Alcaraz, C., & Roman, R. 2007. On the protection and technologies of critical information infrastructures. In *Foundations of Security Analysis and Design IV*, Aldini, A. & Gorrieri, R., eds, 160–182, Berlin, Heidelberg. Springer Berlin Heidelberg.

[18] Alcaraz, C. & Zeadally, S. 2015. Critical infrastructure protection: Requirements and challenges for the 21st century. *International Journal of Critical Infrastructure Protection*, 8, 53 – 66. URL: http://www.sciencedirect.com/science/article/pii/S1874548214000791, doi:https://doi.org/10.1016/j.ijcip.2014.12.002.

[19] Erol-Kantarci, M. & Mouftah, H. T. January 2013. Smart grid forensic science: applications, challenges, and open issues. *IEEE Communications Magazine*, 51(1), 68–74. doi:10.1109/MCOM.2013.6400441.

[20] Security for industrial automation and control systems. Standard, International Electrotechnical Commission (IEC), 2009.

[21] Information technology – security techniques – information security controls for the energy utility industry. Standard, International Electrotechnical Commission (IEC), 10 2017.

[22] Guide to industrial control systems (ics) security. Technical report, National Institute of Standards and Technology (NIST), May 2015.

[23] Knapp, E. D. & Langill, J. T. 2015. *Industrial Network Security (Second Edition)*. Syngress, Boston, second edition.

[24] Senthivel, S., Ahmed, I., & Roussev, V. 2017. Scada network forensics of the pccc protocol. *Digital Investigation*, 22, S57 – S65. URL: http://www.sciencedirect.com/science/article/pii/S1742287617301998, doi:https://doi.org/10.1016/j.diin.2017.06.012.

[25] Wakchaure, M., Sarwade, S., & Siddavatam, I. March 2016. Reconnaissance of industrial control system by deep packet inspection. In *2016 IEEE International Conference on Engineering and Technology (ICETECH)*, 1093–1096. doi:10.1109/ICETECH.2016.7569418.

[26] Årnes, A., ed. 2018. *Digital Forensics*. Wiley, 2 edition.

[27] Salater, T. 8 2016. Iec 62443-serien. URL: https://www.nek.no/manedens-standard-august-2016.

[28] Sanders, C. 2017. *Practical Packet Analysis - Using Wireshark to solve real-world network problems*. no starch press, 3 edition.

[29] Cruz, T., Rosa, L., Proença, J., Maglaras, L., Aubigny, M., Lev, L., Jiang, J., & Simões, P. Dec 2016. A cybersecurity detection framework for supervisory control and data acquisition systems. *IEEE Transactions on Industrial Informatics*, 12(6), 2236–2246. doi:10.1109/TII.2016.2599841.

[30] Wang, K., Du, M., Sun, Y., Vinel, A., & Zhang, Y. November 2016. Attack detection and distributed forensics in machine-to-machine networks. *IEEE Network*, 30(6), 49–55. doi:10.1109/MNET.2016.1600113NM.

[31] Barbosa, R. R. R., Sadre, R., & Pras, A. Sep. 2012. Towards periodicity based anomaly detection in scada networks. In *Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies Factory Automation (ETFA 2012)*, 1–4. doi:10.1109/ETFA.2012.6489745.

[32] Markman, C., Wool, A., & Cardenas, A. A. 2017. A new burst-dfa model for scada anomaly detection. In *Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and PrivaCy*, CPS '17, 1–12, New York, NY, USA. ACM. URL: http://doi.acm.org/10.1145/3140241.3140245, doi:10.1145/3140241.3140245.

[33] Haller, P., Genge, B., & Duka, A.-V. 2019. On the practical integration of anomaly detection techniques in industrial control applications. *International Journal of Critical Infrastructure Protection*, 24, 48 – 68. URL: http://www.sciencedirect.com/science/article/pii/S1874548218301021, doi:https://doi.org/10.1016/j.ijcip.2018.10.008.

[34] Bejtlich, R. 2013. *The practice of network security monitoring*. No starch ppress.

[35] Leedy, P. D. & Ormrod, J. E. 2015. *Practical Research Planning and Design*. PEARSON, 11 edition.

[36] Nov 2018. Iso/iec/ieee international standard - systems and software engineering – life cycle processes – requirements engineering. *ISO/IEC/IEEE 29148:2018(E)*, 1–104. doi:10.1109/IEEESTD.2018.8559686.

[37] Ross, R., McEvilley, M., & Oren, J. C. November 2016. *Systems Security Engineering - Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems*, volume 1. National Institute of Standards and Technology. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-160v1.pdf.

[38] Iec 62264 - enterprise-control system integration. Standard, International Electrotechnical Commission, Geneva, CH, sep 2013.

[39] Cecílio, J. & Furtado, P. *Distributed Control System Operations*, 19–26. Springer International Publishing, Cham, 2014. URL: https://doi.org/10.1007/978-3-319-02889-7_3, doi:10.1007/978-3-319-02889-7_3.

[40] ICS-CERT. Control system engineering workstation. Accessed: 27.01.19. URL: https://ics-cert.us-cert.gov/Control_System_Engineering_Workstation-Definition.html.

[41] Information technology – security techniques – information security management systems – requirements (iso/iec 27001:2013 including cor 1:2014 and cor 2:2015). Standard, International Organization for Standardization, Geneva, CH, feb 2017.

[42] Information technology – security techniques – code of practice for information security controls (iso/iec 27002:2013 including cor 1:2014 and cor 2:2015). Standard, International Organization for Standardization, Geneva, CH, feb 2017.

[43] Reynders, D., Mackay, S., Wright, E., & Mackay, S. 2004. 14a - profibus pa/dp/fms overview. In *Practical Industrial Data Communications*, Reynders, D., Mackay, S., Wright, E., & Mackay, S., eds, 303 – 316. Butterworth-Heinemann, Oxford. URL: http://www.sciencedirect.com/science/article/pii/B978075066395350022X, doi:https://doi.org/10.1016/B978-075066395-3/50022-X.

[44] PROFIBUS and PROFINET International (PI). 2016. Profibus system description technology and application. URL: https://www.profibus.com/index.php?eID=dumpFile&t=f&f=52380.

[45] Sørensen, J. T. & Jaatun, M. G. 2008. An analysis of the manufacturing messaging specification protocol. In *Ubiquitous Intelligence and Computing*, Sandnes, F. E., Zhang, Y., Rong, C., Yang, L. T., & Ma, J., eds, 602–615, Berlin, Heidelberg. Springer Berlin Heidelberg.

[46] Pidikiti, D. S., Kalluri, R., Kumar, R. K. S., & Bindhumadhava, B. S. Jun 2013. Scada communication protocols: vulnerabilities, attacks and possible mitigations. *CSI Transactions on ICT*, 1(2), 135–141. URL: https://doi.org/10.1007/s40012-013-0013-5, doi:10.1007/s40012-013-0013-5.

[47] Kalluri, R., Mahendra, L., Senthil Kumar, R. K., Ganga Prasad, G. L., & Bindhumadhava, B. S. 2018. Analysis of communication channel attacks on control systems—scada in power sector.

In *ISGW 2017: Compendium of Technical Papers*, Pillai, R. K., Ghatikar, G., Seethapathy, R., Sonavane, V. L., Khaparde, S. A., Yemula, P. K., Chaudhuri, S., & Venkateswaran, A., eds, 115–131, Singapore. Springer Singapore.

[48] ABB. Rtu500 series - remote terminal unitshost communication interface with iec60870-5-104 protocol description. Last accessed 12.05.19. URL: https://library.e.abb.com/public/8b837541d31b48baa68fdabf97be9e79/HCI_IEC60870-5-104_en.pdf.

[49] Iso 9506 - industrial automation systems - manufacturing message specification. Standard, International Organization for Standardization (ISO), 07 2003.

[50] Horkan, M. 2015. Challenges for ids/ips deployment in industrial control systems. URL: https://www.sans.org/reading-room/whitepapers/ICS/challenges-ids-ips-deployment-industrial-control-systems-36127.

[51] Suricata. All features - complete list of suricata features. URL: https://suricata-ids.org/features/all-features/.

[52] Kepware. Product overview. Last accessed 05.05.19. URL: https://www.kepware.com/en-us/products/kepserverex/advanced-plug-ins/industrial-data-forwarder-splunk/.

[53] Bejtlich, R. November 2005. *Extrusion Detection Security Monitoring for Internal Intrusions*. Addison Wesley.

[54] Lemay, A., Sadighian, A., & Fernandez, J. 2016. Lightweight journaling for scada systems via event correlation. In *Critical Infrastructure Protection X*, Rice, M. & Shenoi, S., eds, 99–115, Cham. Springer International Publishing.

[55] Siemens. 2019. Simatic hmihmi devicesbasic panels 2nd generation. Last accessed: 26.05.19. URL: https://support.industry.siemens.com/cs/attachments/90114350/hmi_basic_panels_2nd_gen_operating_instructions_enUS_en-US.pdf.

[56] ABB. 2018. Data sheet control panel cp600cp630, cp630-web. Last accessed: 26.05.19. URL: https://search-ext.abb.com/library/Download.aspx?DocumentID=3ADR010243&LanguageCode=en&DocumentPartId=&Action=Launch.

[57] ABB. 2018. Data sheet control panel cp600-ecocp607, cp607-b. Last accessed: 26.05.19. URL: https://search-ext.abb.com/library/Download.aspx?DocumentID=3ADR010240&LanguageCode=en&DocumentPartId=&Action=Launch.

[58] ABB. 04 2013. Compact control builder ac 800mconfiguration. Last accessed: 30.05.19. URL: https://library.e.abb.com/public/0addd95865798de1c1257b66004e24b4/3BSE040935-511_-_en_Compact_Control_Builder_AC_800M_5.1.1_Configuration.pdf.

[59] Nai Fovino, I., Coletta, A., Carcano, A., & Masera, M. Oct 2012. Critical state-based filtering system for securing scada network protocols. *IEEE Transactions on Industrial Electronics*, 59(10), 3943–3950. doi:10.1109/TIE.2011.2181132.