Master's thesis

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

Thomas Johansen

Attack scenarios in critical infrastructure

Remote control of the regional electricity grid

Master's thesis in Department of Information Security and Communication Technology
Supervisor: Prof. Stephen D. Wolthusen Prof. Vasileios Gkioulos
June 2020

NTNU
Kunnskap for en bedre verden

NTNU
Norwegian University of
Science and Technology

Attack scenarios in critical infrastructure – Remote control of the regional electricity grid

Thomas Johansen

02-06-2020

Master's Thesis
Master of Science in Information Security
30 ECTS
Department of Information Security and Communication Technology
Norwegian University of Science and Technology,

Supervisor: Prof. Stephen D. Wolthusen
Co-Supervisor: Prof. Vasileios Gkioulos

Preface

This thesis is the conclusion of my master's degree in Information Security at the Norwegian University of Science and Technology in Gjøvik. It was performed in the fall semester of 2019 and the spring semester of 2020 as it was done in part-time. The thesis gives the reader an introduction to a regional distribution power grid and the components used in remote controlling of it, focusing on building a lab related to the security aspect of this and detailing a set of attack scenarios that can be performed. The work was done with guidance and assistance from my supervisors, Professor Stephen D. Wolthusen and Professor Vasileios Gkioulos.

02-06-2020

Thomas Johansen

Acknowledgment

I would like to thank my supervisors, professor Stephen D. Wolthusen and professor Vasileios Gkioulos for their assistance and advice throughout the process. Their guidance and persistence has been vital to the completion of this thesis.

Further, I would like to thank my wife and kids for the providing me with support and encouragement during this process, granting me time and ability to work unhindered on this thesis and years of study. This work would not have been possible without it.

Thank you.

T.J.

Contents

List of Figures

Abstract

The regional electricity grid is a central and vital component in any national electricity distribution. This grid consist of multiple cyber-physical systems (CPS) and hence is prone to attacks against any of these components, and has in some occurrences been compromised with devastating effects [8]. As the electricity sector has started focusing more on easier controlling, monitoring and in general improving control systems, new technologies will be mixed with the old, legacy systems, mainly implementing the standard TCP/IP stack for this kind of operations. This in turn leads to integration with managerial systems for reporting, which in turn leads to the potential of exposing a critical infrastructure for cyber attacks.

In this thesis, focus will be put on building a lab where attacks against the remote controlling and operations of a regional electricity grid can be performed, and, in time, how a variety of attacks will affect the stability of the regional power grid, potentially revealing how such an attack can affect the delivery of such a critical infrastructure. The result will give a detailed overview of how a simulated environment can be built and a set of attack scenarios that can be reproduced to see what effect the various attacks can have on a real system, building a baseline for a lab environment for further development.

1 Introduction

1.1 Topic

This paper will cover attack scenarios in a cyber-physical system (CPS), which will utilize known attack strategies in the common IT world and be applied to a CPS environment. It will also cover the construction of a lab environment where such attack scenarios can be performed, with the possibility of real simulation data. The lab will initially consist of only simulated data, and all components of the lab has been scaled down to facilitate a educational purpose and the internal logic of some components has been simplified.

1.2 Keywords

Regional Electricity Grid; Attack Scenarios; Cyber-Physical Systems; Industrial Control Systems; Cyber Range; IEC61850;

1.3 Problem description

The energy sector is making advances in their effort to modernize themselves, utilizing remote control management of central control components is one of these steps. The purpose of this modernization is to further implement control systems that can assist in regulating the flow in a regional power grid, optimize paths between supply and demand and ease maintenance operations by not requiring an operator to be physically on site when doing maintenance, to perform daily operations or other mundane tasks. With the introduction of this, one also opens for the possibilities of, remotely, potentially shutting down power supply in large regions, causing critical components in the power grid to seize by manipulating the current or even run high voltage out on a low voltage network. All of these scenarios can cause major physical harm on both equipment, society and people. These are just a few of a large number of scenarios that can occur if a hostile entity is able to get control of these systems. By modernizing and automating tasks, the energy sector is also widely implementing the use of the TCP/IP protocol in remote management tasks and systems and in turn connecting these systems to the administrative network, making them a target for the traditional attack scenarios that is used in traditional IT systems. When it comes to security- and penetration testing of cyber physical systems in the energy sector, there is little publicly available documentation or experience when it comes to this, and the general fear from operators that such testing could impact or even break something during testing can prevent further building of knowledge within the field, other than simple simulations in lab environments on single parts of the system before implementation. By creating various attack scenarios for a system with multiple components, one is able to see the effect the different attacks can have on different parts, or the whole system in all. During this master's thesis, focus will be put on the regional power structure of the power grid

and the remote operations of this. The regional power structure is the link between the producer of electricity and the actual consumer and hence is a critical link in the supply chain, potentially affecting hundreds or thousands of customers. To be able to perform testing and see cause and effect, a miniaturized version of the control system for a regional power structure, based on the IEC 61850 standard, will be created for simulation. Based on this setup, different attack scenarios will be developed and executed on the system, and the effect will in turn be observed and documented. This control system will be a part of the cyber range project established at NTNU.

1.4 Justification, motivation and benefits

The motivation of this thesis is to support further research and knowledge within the field of security- and penetration testing in critical infrastructure, and hence hardening the systems along the way. As the remote control of the regional electric grid is a major critical infrastructure component that affect the society at large, a implementation of a lab environment where attack scenarios can be developed, tested, and later mitigated in a controlled environment is worth the time and effort a Master Thesis will demand.

It is an important task to solve, as there is a lack of knowledge and openness within these systems today. The systems today is considered a "black box" that is best suited to be marked as secret and not tampered with, creating a false sense of security. It is therefore a need to establish a cheap, but reliable, test/lab environment with realistic data and as close to realistic simulations as possible. Some limitations does apply and will be highlighted during this paper.

It is also vital to keep these systems safe from attacks, initially by hardening them so only vital functions is accessible externally, but also to protect them from the ever developing scene of 0-day attacks and customized malware. This is only achievable by exposing a realistic lab environment to the scrutiny of students, researchers, security professionals and the industry itself.

The primary objective of this thesis will be a “live” test-/lab environment for research purposes, down scaled but as close to real as possible, based on set limitations. The secondary objective will be to further knowledge within the field of how different attacks can/will affect remote control systems in a regional power structure.

1.5 Research questions

Based on the previous sections, a set of research questions has been deduced and will be investigated in the report:

1. How can a model of fragments of the regional electricity grid be built and simulated in such a way that it yields a meaningful outcome?
2. How can a reference architecture be modeled in this setup and how will that effect the outcome of the simulations?
3. How will Industrial Control Systems (ICS) and fail-safes interact with measurements during an attack, and can the implemented safeguards be broken?
4. How will a specific attack (or a multitude of them) affect the flow of communication in a

specific architecture?

5. Given that an attacker is inside the control system network and have free access to the IED or ICS, what is needed to fully hinder the normal operations of the deployed architecture and what consequences will that have?

1.6 Contributions

The actual contribution will be to develop a modular, small-scale test-bed/lab environment of a remote control system for a electricity grid, where different attack scenarios can be executed and the behaviour of the system observed during these attacks. This will also contribute to the possibility of further research in the field, to support the security hardening of CPS in the electricity grid and to detect, prevent or be able to mitigate attacks.

2 Related work

In this thesis, focus will be on specific parts of the regional electricity grid in the energy sector and remote controlling of it. It is also simulated how attacks on this can affect both the physical and cyber part of this sector. Based on that selection, focus on existing work will mainly be around work done in the energy sector, but not limited to.
In the first part, focus will be on existing work revolving around the modeling of a cyber-physical system test-beds and in the second part on various vulnerabilities and attack scenarios that can be used in the thesis.

2.1 Modeling and building cyber-physical system test-beds

In this section, work related to the designing and implementing a hardware based test-bed will be looked upon. In [9], various considerations that should be done when designing a cyber-physical system is highlighted, and act as a simple, down scaled methodology for building a simple test-bed. As there are a multitude of existing test-beds, both [10] and [11] have done thorough reviews of multiple, current, test-beds, highlighting a variety of test-beds for a multitude of protocols and network types.

Based on these articles, the different types of test-beds has been arranged in 4 different platform classifications:

- Simulator
- Hardware
- Hybrid
- Real-Time simulator

Each platform is then defined as either a distributed or a centralized architecture, with a few exceptions that can be designed as both.

Based on these classifications, a set of applicable research areas, capabilities and applications for a test-bed had been derived and are described in these articles.

From [1] a set of research applications has been defined of which a test-bed should adhere to. From this article, [12] has created a set of applications and capabilities (reference Figure 1) that a cyber-physical test-bed should be able to do, and this has in turn been mapped to the test-bed that will be created in this thesis.

Applications

1. Impact analysis - Based on metrics gathered from the test-bed, one should be able to perform an analysis of the impact a given attack had on the test-bed, both in regard to reliability, stability and operation.

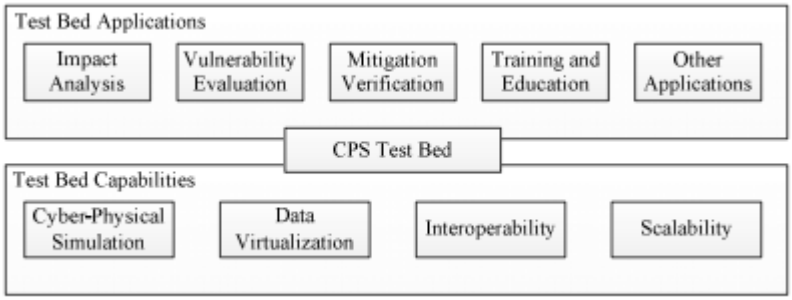


Figure 1: Definition of applications and capabilities in modeling of CPS test-beds [1]

2. Vulnerability evaluation - Given the various attacks, the reports should reveal potential vulnerabilities.
3. Mitigation verification - Implementation of mitigating steps should be possible, verification of these mitigations will then be given in the impact analysis step.
4. Training and education - The purpose of the test-bed is training and education, so ease of use through a user friendly interface and simple rollbacks is important.
5. Other applications

Capabilities

1. Cyber-physical simulation - The test-bed will simulate a specific part of the regional electricity grid with the necessary HMI, PLC's, networks and firewalls.
2. Data virtualization - It should be possible to derive the sets of interested data easily for further studies
3. Interoperability - The system will rely on real time data and it should be possible to connect the system to external hardware.
4. Scalability - The system will be designed to scale both out and up to increase both throughput and nodes/processing power.

In [11] the focus is on IoT smart grids, but is very well applicable in a regional grid as well.

In [13], a specific software simulation test-bed for evaluating cyber security in electric substations was built, based on Java and 100% software based. Here various implementations of protocols are addressed and

[14] focus on the methods that can be used to testing of cyber-physical test-beds and combined with [1] give a high-level overview of architecture, application and evaluation and is a good source for initial tips and tricks.

2.2 Vulnerabilities and attack scenarios in cyber-physical systems

As described in [1], [15] and [10], attack surfaces in a cyber-physical system is not only limited to the actual PLC or HMI, there are multiple entry points to these kind of systems. An engineers workstation or laptop with VPN connected, a guided tour on the distribution centre for a specific region with a unprotected network port or a malicious insider in the regional control centre is other viable paths to access a cyber-physical system.
In the default go-to article in cyber-physical systems, [16], a taxonomy for attacks against SCADA networks is defined, breaking it down to three categories of attacks:

Hardware

The systems designed in cyber-physical systems was originally deemed as safe and secure since it was a closed system with limited access possibility, often you had to be physically on site to access the system. This has since changed, and most cyber-physical systems are now considered a system of systems, connected to central management for easier operations. As an example, this led to a lack of simple access control systems and if there was one in place, it often had a hard coded user with administrative privileges.

Cyber-physical systems also, often, have limited processing power, memory and storage - making attacks against hardware simple if one possess access to a more powerful machine than the targeted device. A simple DoS or targeted Layer 7 attack can then be initiated, exhausting the resources of the targeted system.

Software

A cyber-physical system will consist of a multitude of devices, each one containing a operating system, remote access tools or even a small web server. All of these components have the potential of exposing the unit for a exploitable feature, ranging from escalation of privilege, SQL injections, buffer overflow and remote execution of code to mention some of the heavily exploited attack vectors.

Communication

When it comes to communication, the attack surface is large. Attacks can be performed against a range of layers in the OSI model [17], mainly revolving around layer 3, 4 & 7.
Layer 3 is the Network Layer, opening the stack to attacks such as IP or ARP spoofing, routing attacks, ICMP attacks/flood or Teardrop attack to mention some.
Layer 4 is the Transport layer, providing the possibility of SYN flood attacks, Smurf attacks or in general volume based attacks. This is often used as a Denial of Service (DoS) attack, utilizing zombie armies to overburden the target with traffic so the target reach its connection limits.
Layer 7 is the application layer, utilizing flaws in the application itself. Differentiated from a typical DoS attack through the transport layer by specially crafted requests to exhaust the resources of a target, either overburdening it with work or just making it wait for requests. So in stead of massive amounts of requests overburdening the target, a targeted request designed to maximize the resource usage of the target is made and make it DoS itself.

2.2.1 Attack scenarios

As cyber-physical systems are a set of interconnected devices, often through a LAN or WAN, typical attack scenarios can be utilized. But since there are some legacy systems and systems based on proprietary protocols, this give us a other opening for attacks, as these protocols often have limitations and known vulnerabilities outside the normal scope of a typical attack.
As described in [10], the typical attack types examined in a cyber-physical system test-bed is and most of these will be described in detail in section 5.1:

1. Man-in-the-middle
2. Precision insider
3. Rogue Software
4. Denial of Service
5. ARP Spoofing
6. Eavesdropping
7. Malformed packet
8. Database attack

In [18] a "man in the middle" attack is described in detail as a stealthy attack against both PLCs and HMIs to trick the HMI to display the chosen values, and hijacking the PLCs.

[19] has good examples of both the "Precision insider" and "Rogue Software" attacks.
The "Precision insider" takes account for a malicious insider in the organization with good knowledge of the system, as a random change of parameters will, most likely, not cause significant damage. On needs good knowledge about the system to actually be able to perform damage.
A "Rogue software" attack is a specific attack crafted by a malicious employee (or employees/organization) of a supplier to the organization. Let us say that there is a software developed by a 3rd party organization to manage generators in sub stations of a grid. The malicious agent has injected code in the program supplied and set it to manipulate specific values of the generators at the same time, causing breakdown or worse to happen.

3 Choice of methods/Methodology

This chapter explains the different methods used in this thesis. The methods formed a baseline for necessary information needed to complete the thesis.

The process will consist of several steps and certain steps is dependent on a previous step, following guidelines from [20] and [21]. The phases can be divided into four main parts;

1. Gathering relevant information
2. Modeling and producing an actual test bed
3. Model and perform attacks against the test bed
4. Analyze the outcome of the previous mentioned attacks.

3.1 Background information

To get a understanding of how the specific elements and process in a regional electricity grid works, background interviews with relevant representatives from the industry was conducted. Based on these interviews, a down scaled model of the control system for a regional power grid was be designed, but due to limitations on details that could be provided due to security concerns, all details were given in high level details. Additionally, details around how such a system in general is protected and which procedures and manual fail safes exist was examined by reading relevant literature.

Relevant literature revolving around the basis of regional electricity grid and the components found in them was procured , and in that part specifically remote control of such (or similar) systems. Information around model building and frameworks for the industry was also gathered.

This formed the basis needed to be able to proceed to the next step, which is the actual modeling of a remote control system for a regional power structure.

3.2 Modeling a basic system

To be able to create a model of a remote control system, the characteristics of such an environment must be replicated to the small-scale model. To begin with, the advisories from NIST [22] to build secure critical infrastructures forms a baseline, but this might yield to make space for a realistic, in production scenario deployment in a scenario where that can be procured. This includes simulation of, some of, the physical processes done by cyber-physical systems controlled by the remote system and potential safe guards set in place in such a system was considered to get a realistic as possible scenario. The development model is based on, at least, two intelligent electronic devices (IED), a industrial control system (ICS) with a human machine interface (HMI) and the network between them. Since the remote controlling of these operations is among the targets of this thesis, the network between the IED and the ICS will be simulated to 3G speed (10Mbps) or slower, adding

latency to the mix. In [23] and [24], a good overview of modeling high voltage systems is described, while [10]. [9], [25] and [1] gives a good overview over designing and building security oriented testbeds, while [14] concentrate on testing of the testbeds.

[26] gives a good overview of industrial network security.

A set of software for communicating is developed using [27], simulating the IEDs and HMI, as well as a interaction program between the control station and the HMI. Additionally, scripts utilized in the attacks was created and executed.

3.3 Simulating a CPS

If possible, acquisition of real world data would be ideal to form a baseline for behaviour of the CPS. Again, due to security concerns, this was out of the question. Based on either this, or manual input and feedback from operators, a simulated program was created in C/MatLab/SimuLink. The programs is executed on their assigned, physical, device, forming a complete test bed. The basis for OS on the devices was decided to be Debian 10 [28] for the virtual machines and the latest release (at the time) of RaspianOS [29] for the physical Rapberry Pi3.

3.4 Development of attack scenarios

After the modeling and simulation of the system was complete, the process of designing and conducting attacks against the platform began. In this process, the "standard" attack platforms is utilized, based on the ENISA taxonomy [30], while reviewing existing literature for known attacks in various protocols, like [31], [32], [33] and [34], formed the attacks chosen. DoS attacks, replay attacks and delay of packets, alteration of traffic and general disturbance was performed and the results recorded and analyzed. In these scenarios, the assumption is made that the attacker is already inside the control system network, bypassing the initial steps of getting remote access and elevated rights to the system.

3.5 Analysis

This is the final step of the process, and is where the actual results of this thesis is visible. The form of attacks performed here should yield a good enough data set to perform a statistical analysis. For other, not so numerical data gathered, a qualitative analysis will be done.

Since this process is iterative, analysis was at some stages performed several times, but summarized in chapter 7.

4 Industrial Control Systems for Regional Power Grids Overview

This chapter will give an overview of a regional power grid and a introduction to the common components and protocols in such a infrastructure. All topics will not be a part of the thesis, some are just mentioned in general to present to the reader that the topic is noted, but not necessarily included as a part of the thesis. This section will also cover the basics of attacks and vulnerabilities in a CPS. To sum up this chapter, a reference architecture for a regional power grid will be presented.

4.1 ICS components

An Industrial Control System (ICS) consist of a variety of components. In this chapter, the most common components will be listed and a brief description of the component will be given. In figure 2 a typical presentation of a ICS system can bee seen.

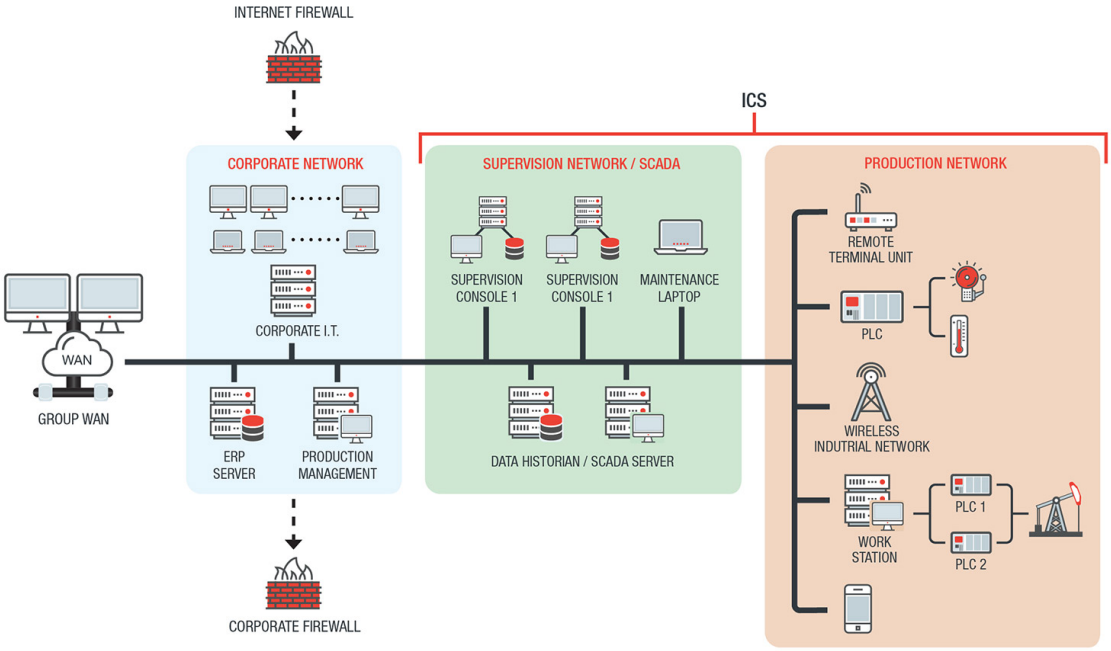


Figure 2: A typical layout of ICS components [2]

4.1.1 Programmable logic controller - PLC

Programmable logic controllers are used for automation in electrical or mechanical processes, and is therefor, hardened devices designed to resist high/low temperatures, physical shock, electrical noise and similar, external disturbances. Is practice, they are, lightweight, computers designed for receiving inputs and providing outputs and execute commands based on a ladder logic program, based on the IEC-61131-3 standard. They are designed for time critical systems, which critical systems often are, and have to be able to respond in a time critical manner. Usually found in local environments with physical communication infrastructure [35].

4.1.2 Remote terminal unit - RTU

Remote terminal units is also called remote telemetry units, and is distinguished from a PLC in the way that they use wireless communication for transmission, making them ideal for use in geographically dispersed configurations. Compared to the PLC, it is usually even more physically hardened so it can be utilized outdoors. An RTU does not support control loops or algorithms, so in this way, it also differs from a PLC, but the software which it runs is, usually, based on the IEC-61131-3 standard [26].

4.1.3 Intelligent Electronic Device - IED

The IED is, like the PLC and RTU, a device that can communicate between sensors and controllers. The main difference between these and an IED is that an IED can communicate with, and control several parts of a physical component. In this way, the component can be monitored and controlled as a whole. The IED allows for a two way communication, allowing both monitoring/acquisition and control. IEDs are, usually, classified by the function, with common types being relay devices, circuit breaker controllers, voltage regulator and re-closer controllers. One IED usually operate more than one function, allowing it to take on multiple roles within one substation. The implementation of IEDs in ICS allows for more automatic controls and decisions to be done locally, as opposed to a MTU/SCADA server setup, where a central server process values and take action based on this. Based on this, a IED based implementation should give a more robust architecture, as it is based on a distributed architecture instead of a centralized node doing all the control functions. An IED usually have a variant of Linux running on it as the base operating system, finely tuned to adhere to timing and availability requirements in a CPS environment.

4.1.4 Human machine interface - HMI

The Human Machine Interface is the visual part of the machine, or the local control panel. The HMI is giving a person the ability to interact with a machine or a device through a interface, often referred to as the Graphical User Interface (GUI) as this puts a visualization to the commands. A operator can configure various set points in the system and adjust thresholds for parameters from the HMI. It can also display historical data gathered from connected devices [26].

4.1.5 Historian

A historian is a time-series database that is often embedded in or used in cooperation with ICS systems that have a need for historical data. Data is archived in this database and later retrieved for analysis, often to fine tune systems or trace the activity of control operations done on a specific device [36].

4.1.6 Master Terminal Unit - MTU or SCADA Server

The MTU is a device in the ICS chain that issue command to RTUs, PLCs and IEDs. The MTU is usually a centralized machine that operate on multiple sub-systems. The MTU gathers data, process them for informational display on HMIs and store them for later retrieval. It also assists in taking control decisions based on the information gathered. Most of these decisions are automatic, but manual input to certain functions can be done [36].

4.2 Standards and protocols used in the Energy sector

The definition of a standard by the Oxford English dictionary is:

> An authoritative or recognized exemplar of correctness, perfection, or some definite degree of any quality.

Within the field of computer science, a standard is usually defined as very specific set of accepted best practices.

In addition to the "common" protocols in a communication network, like Ethernet, IP, UDP and TCP - there are industry specific protocols that expands the different layers in the OSI stack with additional capabilities tailored to the use in a give industry.

This section gives a brief introduction to the different standards and protocols used in the Energy sector will be given, with primary focus on IEC61850 which is the primary standard used in this paper.

4.2.1 ISO/IEC 27019 and 27002 standard

The ISO/IEC 27019 is a standard based on ISO/IEC 27002, a information security standard, but with focus on process control systems used by the energy sector.

4.2.2 ANSI/ISA-95 standard

The ANSI/ISA-95 standard is standard developed by the International Society of Automation to develop a interface between enterprise and control systems. The goal is to provide a foundation for consistent terminology and to provide consistent information and operation models on how information is to be used. In short, it gives best practices for integration between a secure process environment and the business layer.

4.2.3 IEC 60870 standard

The IEC 60870 standard define systems and protocols used in supervisory and data acquisition in electrical engineering and power system automation, enabling interoperability between various vendors of equipment, defining a standard exchange of data. The standard is divided into 6 parts,

where the IEC 60870-5-104 part is the most interesting to look at form a technical perspective. Here the standard define network access, enabling messages as application data over TCP/IP. The standard defines the basis for the IEC 60870 protocol, but is also the standard that the DNP3 protocol derived from.

IEC 60870-5-101 and IEC 60870-5-104

As mentioned in 4.2.3, the main part of the IEC 608750 standard that is interesting for us, is the IEC 60870-5 which defines the transmission protocols between two components over TCP/IP. For the application to separate systems, a set of companion standards were developed, and here IEC 60870-5-101 and IEC 60870-5-104 stands out. The IEC 60870-5-101 defines protocols used in basic telecontrol tasks, while the IEC 60870-5-104 define protocols for accessing the IEC 60870-5-101 over standard transport profiles, basically meaning accessing 60870-5-101 over standard TCP/IP functions. These protocols are widely used in Europe and Asia.

The protocol supports both mast initiated and master/slave initiated communication/data transfer, meaning that a control station can initiate traffic and the node can initiate sending of traffic to a master, giving the possibility of allowing nodes generate alerts to a master without the master polling it. The data sent can be classified into objects or groups, and each object/group can be addressed directly, giving the possibility to read specific parts independently. It is also possible to give various data priority groups, implementing a sort of QoS for the data.

4.2.4 IEC 61850 standard

The IEC 61850 standard define the communication part of electrical, substation automation, but can be, as noted by [4], complex and hard to understand for others than experts within the domain of electrical substation engineering. To be able to build a test-bed and execute security related experiments, background on some of the various elements of the IEC 61850 standard can be useful. In this part, a brief introduction to the documents of the standards will be presented, and core values of the operational part of the standard. Protocol specific information will be covered in the next section.

A goal for the standard is to define a unified information model with a naming hierarchy and data structures to avoid proprietary, vendor specific models. It also defines a communication protocol and unified server functionality to fulfill requirements of automation of substations with keeping in mind timing and availability requirements.

The IEC 61850 standard consist of 10 parts as listed in the table 1. Part 1 to 5 gives an general introduction to the standard, requirements, project management and physical implementation. Part 6 covers IED configuration through a XML based standard called "Substation Configuration Language (SCL)", which was implemented to minimize the component of human error when configuring these complex setups. Part 7, 8 and 9 is the most relevant to our part, as part 7 covers the logical concepts of the standard and part 8 on how the internal objects in the model can be mapped to the application and Ethernet layer. Part 9 define the mapping of sampled measurement value (from now on referred to as SMV) to Ethernet.

The standard, in addition to define how bytes are transmitted over a transportation media, allows

for consistent organization of data across all types of devices. This, in turn, allows for automatic detection and configuration of devices.

In part 7, the abstraction of data items and services are defined. This "new" way to think is one of the reason why IEC 61850 was developed. This abstractions makes the data items and services independent of underlying protocols, and allows mapping of objects and services to any protocol that meet the data and service requirement [3].

The abstraction of data objects is referred to as Logical Nodes (LN) and a concept called "Common data Classes (CDC)" was defined to define building blocks for larger data objects. In part 8, these abstract data objects and services, is mapped to the "Manufacturing Messaging Specification (MMS)" In part 9, the SMV are mapped to the ethernet data frame.

Part	Title
1	Introduction and overview
2	Glossary of terms
3	General requirements
4	System and project management
5	Communication requirements for functions and device models
6	Configuration description language for communication in electrical substations related to IEDs
7	Basic communication structure for substation and feeder equipment
8	Specific communication service mapping (SCMS) - To MMS and Ethernet
9	Specific Communication Service Mapping (SCMS) - From Sampled Values
10	Conformance testing

Table 1: IEC 61850 standard documents

Each IEC 61850 model is defined as a physical device connected to the communication network, usually defined by its network address. Each physical device can contain one or more logical devices (LD), giving the physical device the role as a gateway for multiple devices. A logical device can again contain one or more logical nodes (LN) which is a grouping of data and services that is related to some function. The logical nodes are, usually, following a naming convention with names beginning with a letter, based on its function followed by a Instance ID as a suffix, like the following examples [3]:

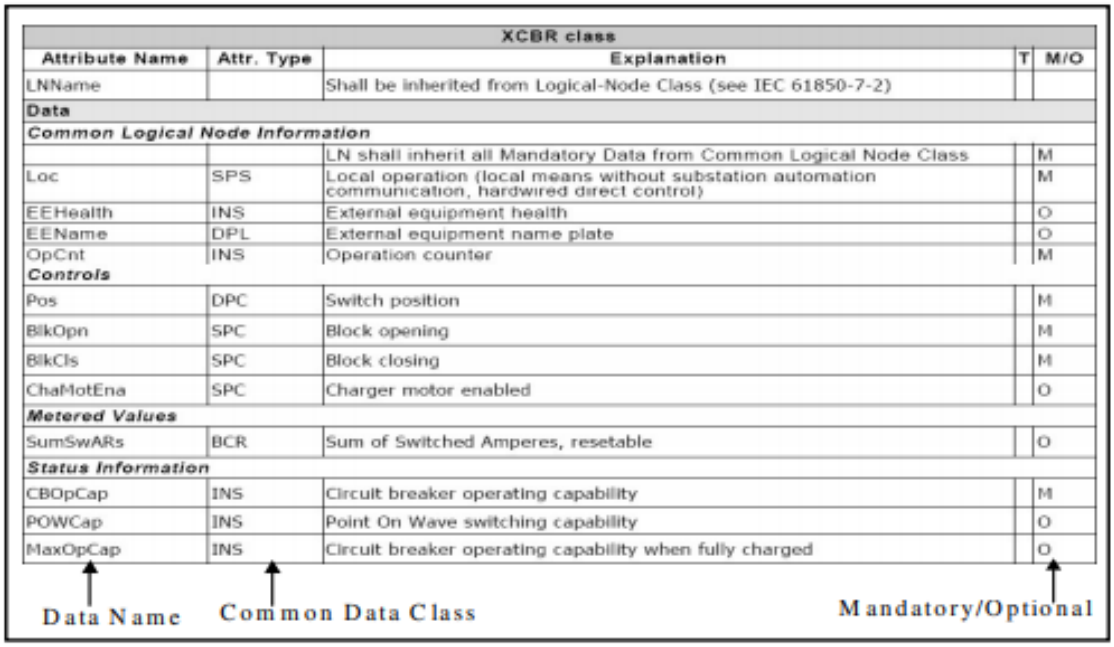
- A - Automatic Control
- M - Measuring and metering
- C - Supervisory Control
- G - Generic functions
- I - Interfacing/Archiving
- L - System logical nodes
- P - Protection
- R - Protection Related

- S - Sensors
- T - Instrument transformers
- X - Switch gear
- Y - Power transformers
- Z - Other equipment

To distinguish between two circuit breakers, with the standard name of XCBR in a LN, the naming of each circuit breaker would then be XCBR1 and XCBR2.

Each LN contain one or more elements of data and each element has has a unique name. The names are determined by the standard and are related to the purpose of the data and is defined in IEC 61850 7-4. See figure 3 for a example of all mandatory (M) and optional (O) data for a XCBR LN. A XCBR contain the following, mandatory, data:

- Loc - Determine if operation mode is remote or local
- OpCnt - Operations count
- Post - For position of breaker
- BlkOpn - Block breaker open commands
- BlkCls Block breaker close commands
- CBOpCap - Circuit breaker operating capability

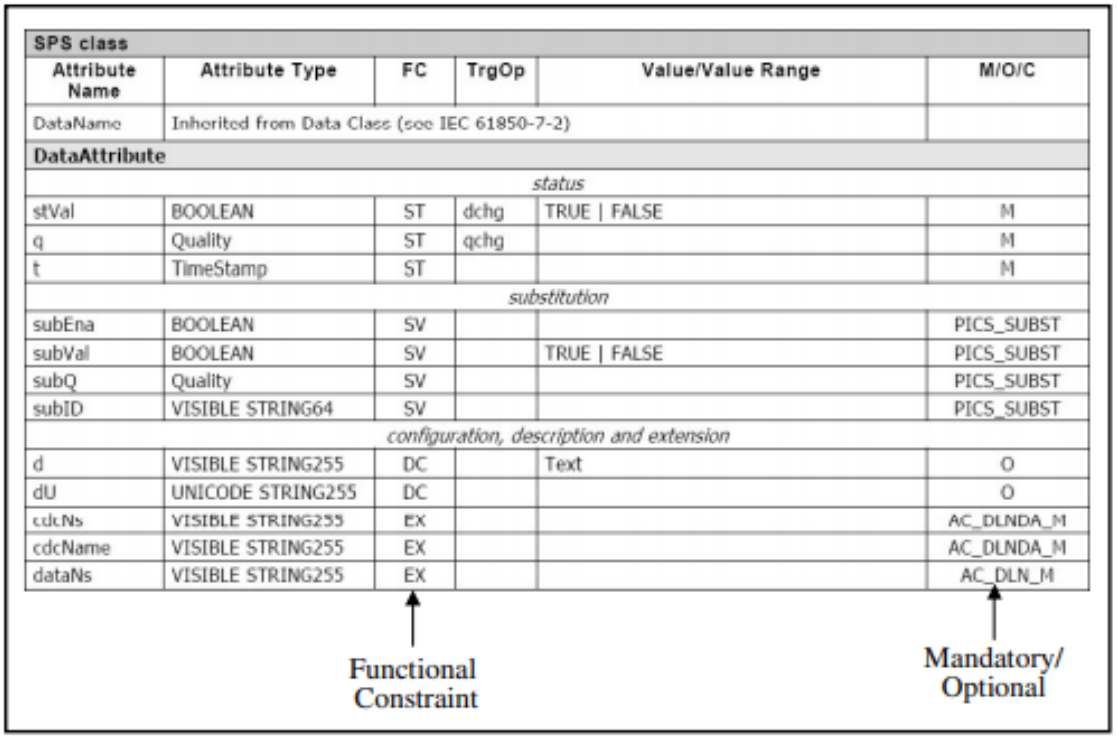


XCBR class				
Attribute Name	**Attr. Type**	**Explanation**	**T**	**M/O**
LNName		Shall be inherited from Logical-Node Class (see IEC 61850-7-2)		
Data				
Common Logical Node Information				
		LN shall inherit all Mandatory Data from Common Logical Node Class		M
Loc	SPS	Local operation (local means without substation automation communication, hardwired direct control)		M
EEHealth	INS	External equipment health		O
EEName	DPL	External equipment name plate		O
OpCnt	INS	Operation counter		M
Controls				
Pos	DPC	Switch position		M
BlkOpn	SPC	Block opening		M
BlkCls	SPC	Block closing		M
ChaMotEna	SPC	Charger motor enabled		O
Metered Values				
SumSwARs	BCR	Sum of Switched Amperes, resetable		O
Status Information				
CBOpCap	INS	Circuit breaker operating capability		M
POWCap	INS	Point On Wave switching capability		O
MaxOpCap	INS	Circuit breaker operating capability when fully charged		O

Data Name | Common Data Class | Mandatory/Optional

Figure 3: IEC 61850 - XCBR overview [3]

Each element in the LN have to conform to the defined CDC in the standard (IEC 61850 7-3). Each CDC describe the data type and structure within the LN. Each CDC has a defined name and a set of attributes (like the LN) with mandatory (M) and optional (O) attributes. In figure 4 the mandatory and optional attributes of the SPS class can be seen. Here we have the three, mandatory, attributes of status value (stVal), quality flag (q) and times tamp (t). The model also contains functional constrints (FC) that groups the attributes into categories. In this example we have the status (ST), substituted value (SV), description (DC) and extended definition (EX) attributes.



SPS class

Attribute Name	Attribute Type	FC	TrgOp	Value/Value Range	M/O/C
DataName	Inherited from Data Class (see IEC 61850-7-2)				
DataAttribute					
			status		
stVal	BOOLEAN	ST	dchg	TRUE | FALSE	M
q	Quality	ST	qchg		M
t	TimeStamp	ST			M
			substitution		
subEna	BOOLEAN	SV			PICS_SUBST
subVal	BOOLEAN	SV		TRUE | FALSE	PICS_SUBST
subQ	Quality	SV			PICS_SUBST
subID	VISIBLE STRING64	SV			PICS_SUBST
			configuration, description and extension		
d	VISIBLE STRING255	DC		Text	O
dU	UNICODE STRING255	DC			O
cdcNs	VISIBLE STRING255	EX			AC_DLNDA_M
cdcName	VISIBLE STRING255	EX			AC_DLNDA_M
dataNs	VISIBLE STRING255	EX			AC_DLN_M

Figure 4: IEC 61850 - SPS CDC overview [3]

This, abstract, model is then mapped onto a specific protocol, defined in the IEC 61850-8-1 part, based on MMS and TCP/IP over Ethernet. In this process, the model is transformed into a named MMS variable object.

I.E a LD named "Relay1" consisting of a circuit breaker LN named XCBR1 and you want to know if it is operating in local or external mode. To know this, one would have to read the object [3]:

`Relay1/XCBR1$ST$Loc$stVal`

- Relay1 is the Logical Device (LD)
- XCBR is the Logical Node (LN)

- $ST is the Functional Constraint (FC)
- $Loc is the Data
- $stVal is the Attribute

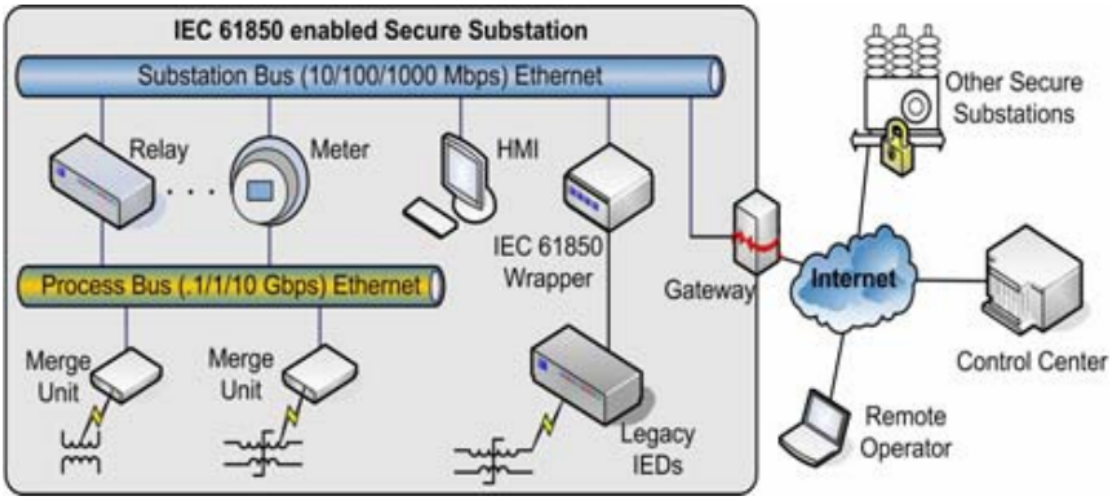


Figure 5: IEC 61850 Substation architecture [4]

4.2.5 Modbus

Released, originally, in 1979 and is considered one of the oldest protocols used in remote controlling in ICS. The original implementation was based on serial communication only (RS485), but due to its widespread usage, a TCP implementation made its way later [37]. It gained popularity due to the fact that it is openly published and royalty-free. A Modbus implementation allows up to 32 devices on one link, and each command sent on the modbus network will be received by each Modbus node, but only the device with the matching address will respond.

From a security perspective, Modbus lacks security in general. It transmits all messages in clear text, no integrity checks are made and no authentication mechanisms exist. It also lacks encryption features.

4.2.6 Distributed Network Protocol 3 (DNP3)

DNP3 was based on the unfinished IEC 60870-5 protocol specification as there was a need for a immediately implementable protocol for North American requirements (citation here) and is still the dominant SCADA protocol used in North America. It defines how devices communicate control command and process data in a SCADA system. It supports three different methods of communication [31]:

- Unicast - Where the master sends a request command to a device and the device responds

with a reply

- Broadcast - The master sends a request command to all devices in the network, the devices does not respond
- Request initiated from device - The remote device sends a response to the master, without the master asking first. This allows for providing updates or alerts.

In addition to the various communication methods, it allows for a multitude of network configurations, but the three most popular as mentions in [31] is:

- One on one - One master connects to one remote device
- Multi drop - One master connects to multiple remote devices
- Hierarchical - One master connects to one or more sub-masters

The latest version of DNP, seen as the "secure" version of DNP. Supports time stamping, authentication, redundancy checks. This protocol is usually favored in America.

4.3 Regional distribution grid

The regional distribution grid is situated between the producers of electricity and the actual end consumer. As can be seen in figure 6, marked with the ledger Green, the regional distribution grid is responsible for stepping down the high-voltage transmission lines and delivering the electricity to sub-stations where it is, in turn, distributed to industrial, commercial and residential consumers.

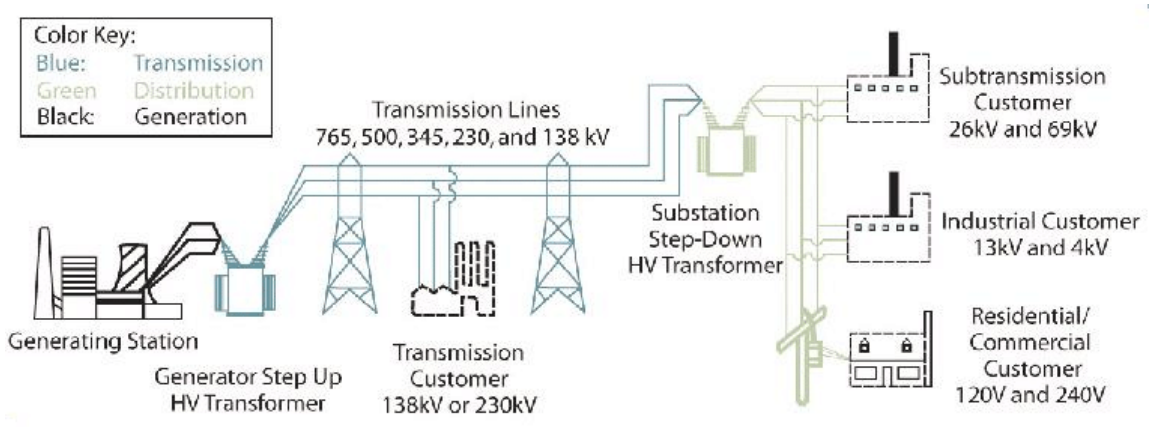


Figure 6: A conventional electrical grid [5]

4.3.1 Phase-shifting transformers

A phase-shifting transformer is a critical component in the regional grid, allowing active control of power flow in the grid independently of the generation of it. By enforcing or blocking load, it improves stability and flexibility in the grid. This has become extremely important in the later

years, when the expansion of renewable energy, causing bottlenecks in some parts of the grid. By distributing load on parallel sections, phase-shifting generators overcome that obstacle, helping the network operator maximize the utilization of existing transmission lines and hardware. The phase shifting transformer is connected to a SCADA system which operate the on-tap controllers that regulate the phase angle of the transformer. This is an automated process, connected to a SCADA system and is usually an automated process, but can be manually processed.

The distribution is done by changing the angle of the phase between the transformers primary and secondary side based on calculations of the system. A tap-changer is regulating the switching steps and balancing the flow of power based on the need of the consuming side.

4.3.2 Substation step down transformer - High voltage converter

A high voltage direct current converter station (HVDC converter) is located at central points in the electricity grid and connects different parts of the grid together. The main purpose of the step-down transformer is to change the voltage to a lower degree than in the transportation part of the grid, usually to distribution substations that is located closer to the consumer, although industrial consumers that require higher voltage can tap directly into the transportation grid. According to [38], an attacker with knowledge in transformer design could cause permanent damage to a transformer.

4.4 Reference architecture

The reference architecture for the attack scenarios is based on a ANSI/ISA-95 separation of network layers and applied to a SCADA architecture [22], as can be seen in figure 7, and define a set of common components utilized in a regional distribution grid, based on a simple control center and a single sub station.

As stated in the standard, it is defined 5 levels of separation between the various components in the architecture>

- Level 4: Business Operation Management
    - As this is not a part of the attack scenarios, this has been omitted in the reference architecture.
- Level 3: Operation management
    - Where the operators are located and monitoring the system. Most automatic corrections is done locally by the "smart" IED, but some manual processes are still done by the Engineering Workstation or SCADA server. In our reference architecture, this is based on one Engineering station and one HMI/SCADA server. Communicate with the sub-HMI through IEC61850 MMS.
- Level 2: Supervision and monitoring
    - A local HMI for process operator/maintenance tasks performed locally. Same setup as in Level 3 Engineering workstation. Communicate with the IEDs through IEC 61850 MMS and potentially GOOSE.

- Level 1: Production and control processes
  - Three IEDs based on libIEC61850 library with customized code. Performing actions based on simple (programmed) logic when interacting with the simulated model - based on Simulink.
- Level 0: Sensors and signals
  - Three Simulink models simulating actuators and sensors. Communicate with the IED through standard TCP/IP.

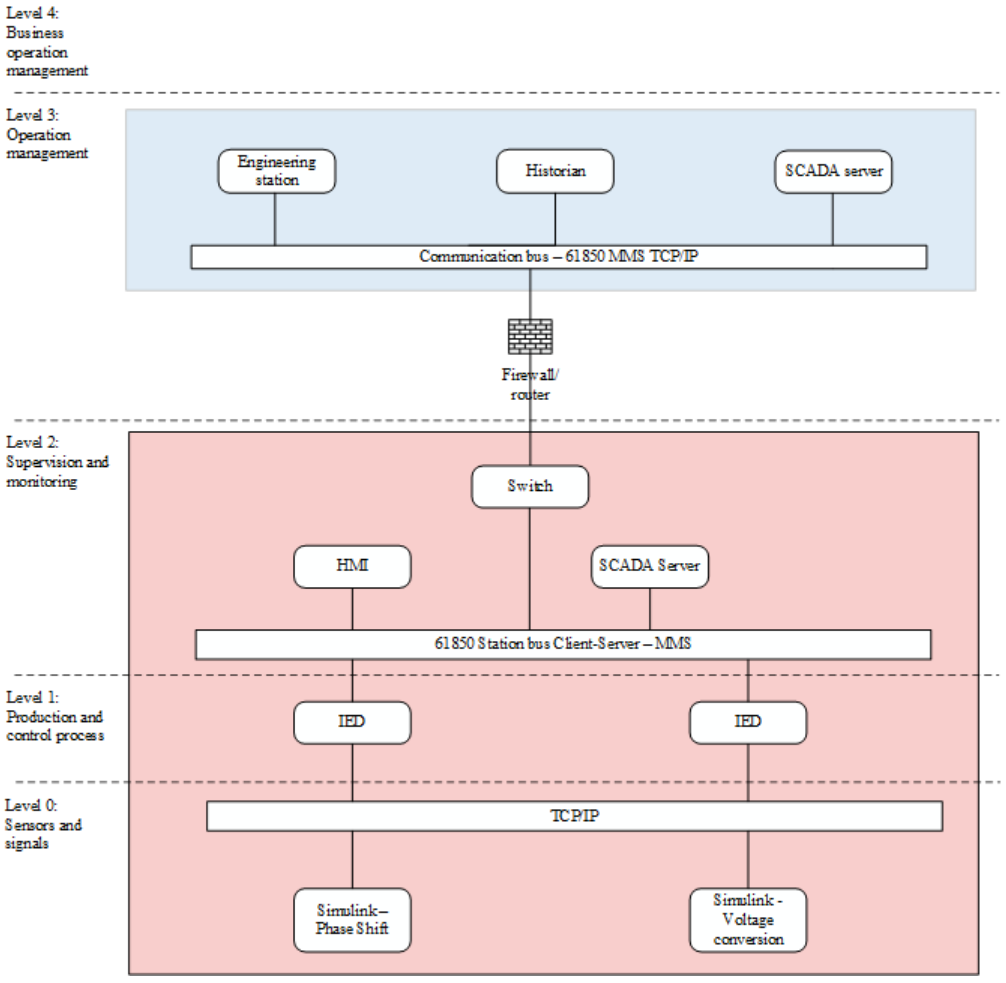


Figure 7: Reference architecture based on the ISA95 standard [6]

Focus in this reference architecture will be on the remote controlling of the regional grid, the traffic between a Engineering workstation/HMI/SCADA server and the components out on the sub-

stations. The measurement will be on how these attacks affect the specific sub system.

A basic regional distribution grid links a transmission grid to a distribution grid, and is therefor a critical component in the conversion of high voltage (transmission grid) to low-voltage (distribution grid), but it can also link directly to production or consumers requiring higher voltages. In Norway, the distribution grid contain voltage spanning from 33 kV to 132kV.

5 Attack scenarios/description of attacks

In this chapter, a description of various attack methods relevant to a ICS is described, based on the different layers in the OSI model. This is then followed by a walk through of three attacks scenarios, where the different attack methods is implemented and all assumptions, prior baselines and prerequisites for the attack to be successful is mapped out. To summarize this chapter, steps on how to mitigate these attacks are presented.

These attack scenarios is a single set of attacks that targets specific parts of the architecture presented in 7, but can be utilized together as joint or cascading attacks to affect larger parts of the infrastructure.

The joint attacks perform attacks at multiple points in the infrastructure simultaneously, while a cascading attack targets essential parts of the grid where an attack will cascade the issue to a point where it will "snowball" and the system will effectively overburden itself as described by [34]

5.1 Description of various attack methods

In this section, a set of known attack methods that can be utilized in the attack scenarios is described, focusing on the effect related to the implementation of the IEC 61850 and weakness in the operating system of the IED's.

In this paper, the following classifications of attacks will be used and a set of methods to perform this is described below.

1. Layer 2/3 Attacks
    - Man-in-the-middle (MitM) attack
    - Replay attack
    - Eavesdropping attack
    - Injection/Modification attack
    - Spoofing attack
    - Delay attack
    - DoS attack
2. Layer 6/7 Attacks
    - Operating system level of IED
    - Application specific attacks
    - Denial of Service attack

5.1.1 Layer 2/3 Attacks

Layer 2/3 attacks is based on attacking the transport level of traffic between the systems, in other words: How can we utilize various applications to exploit general vulnerabilities in the protocols to our advantage.

When it comes to replay, injection and/or delay attacks, the goal of the implementation of this in the attack scenarios is to utilize these techniques to affect commands sent, either manually or automatically, from the control center to the SCADA Server/IED that is performing an action on the actual sub-system. This is done to emulate the effect this action has on the ability of the corresponding system to act and see the results.

For replay/injection to be successful, there is a need to capture traffic from a MTU/HMI to SCADA Server/IED and either have control over the initiating operating system or "blackhole" the original traffic before it reaches the IED by controlling network nodes between the initiator and the receiver. To remain undetected, there is also a need to send expected results back to the origin, so no alarms is triggered. An other possibility here is proxying the traffic through an "invisible proxy" and alter the traffic in transit.

For delay attacks to be successful, we need a way to throttle traffic, without timeouts being sent to the origin and a buffer to keep recorded values to send to the destination after a set amount of time. As delay attacks tend to, primarily, be on time sensitive data, these attacks usually are performed on the sub-station itself, between the IED and the SCADA Server locally. The goal of attack is to disrupt commands from HMI/SCADA Server to IED sufficiently to prevent the mitigation of faults/errors/deviations.

Man-in-the-middle (MitM) Attack

A MitM attack utilize a weakness in communication between two systems, either by exploiting known vulnerabilities in authentication and authorization between two systems, or by gaining access to unencrypted communication between these systems. An attacker can, by using this weakness, observe, control and/or alter communication between two systems without the systems detecting it. This attack method can be utilized by physically gaining access to communication networks between the systems, or by compromising one of the parties and rerouting traffic on one or both sides. This is a common approach to attacking the DNP3, Modbus, IEC60870, and IEC61850 protocols [32].

Replay attack

A replay attack use a known vulnerability in the authentication of a request. An attacker gains access to, i.e. a cookie containing a valid authentication request, and can by reusing this request, send valid messages to the receiving system. The attacker can, by using this method, gain valid access to the system posing as the original sender. [39] states that this type of attack typically targets Modbus, DNP3, IEC60870 and IEC61850 protocols.

Eavesdropping attack

The goal here is to capture valid commands and perform actions based on this, usually by sending the same messages to the IED from a different computer. This type of attack does not need to "infiltrate" the actual host sending/receiving the commands, but passively listens on the wire to capture the messages. As IEC 61850 relies on multi-casting, messages can be eavesdropped on as long as the attacker is situated on the same network, implying that the attacker has physical access to the site or access to a compromised unit on the network.

Injection and/or modification attack

The goal of a injection/modification attack is to capture commands/data sent between two hosts. Usually, this requires one of the hosts to be compromised, but can also be used in spoofing. A valid command is captured, and the original request "black holed", meaning that the recipient never received the original request. The request is then modified by the attacker, and "resent" to the recipient. The request now contain the modified commands and the attacker is able to trick the receiving system.

Spoofing attack

In a successful spoofing attack, an attacker successfully impersonate another user or device by spoofing IP address, MAC address or similar. This is done to attack hosts, bypass access controls or spread malware.

Denial of service (DoS) attack

In a Denial of service attack, the goal is to overburden the victim system with bogus or/and legit traffic to such an extent that it is unable to process it anymore, rendering the system useless/incapable of performing.

Requirements of attack: Preferably real/acceptable traffic to the HMI/MTU or IED. A way to overburden the IED/HMI with sufficient traffic (a VM or similar with enough capacity to generate traffic)

A DoS attack targets the availability of a system by overburdening or interrupting a systems' communication service. This can be done by either overburdening the physical medium (I/O device) by flooding it with relevant or irrelevant data, preventing legitimate traffic from gaining access the system.

5.1.2 Layer 6/7 attacks

Operating system level of IED

IEDs are usually based on a Linux kernel and is therefor also susceptible for vulnerabilities in these. Can utilize 0day attack and the fact that many IEDs are not frequently updated. Gives the possibility of older attack methods working.

Application specific attacks

An application specific attack exploits known vulnerabilities/design flaws in a set of applications, usually exploiting bad coding in the L6/7 stack. These attacks can, usually, only be fixed by releasing

updated versions of running software, but in some instances extra hardening on the firewall level can mitigate some of the attacks.

Denial of Service attack

A other way, is by utilizing a known vulnerability in an exposed application, L7 attack, and effectively making the application overburden itself, rendering it inaccessible for legitimate requests. This is, as stated in [40] and [41], a know vulnerability in IEC61850, IEC60870, Modbus and DNP3 protocols.

5.2 Attack scenarios - Targeted attacks of segments in the infrastructure

As can be seen in figure 7, the attack scenarios are all based on network level 3, 2 and 1 in the ANSI/ISA-95 standard. To get to this stage, there is a set of prerequisites, common to all attack scenarios, that has to be fulfilled for an attack to be successful. Additional, there are some prerequisites that has to be in place to be able to monitor the flow and impact of an attack:

- The attack must be performed from an internal workstation or electronically controllable device with some kind of network connection that can be utilized to piggyback on to the actual control system or network.
- The attack must be legitimate traffic to avoid internal security mechanisms.
- The IED is sending the actions performed to a simulated system based on Simulink. This model is running a TCP server/client architecture, adding to the time used from a command is sent until it is performed. No physical actuators/sensors are implemented in the system.
- The assigned IEDs to each scenario is a simplified version of an IED, performing no corrective measures to prevent the attacks, as the goal is educational purposes of the attacks.
- The lab must be easy to reset for change of attack parameters and add/modify attack scenarios.
- Metrics from the attack must be easily accessible, in this instance we are using Grafana with InfluxDB for graphing and time series metrics storage.

All attacks are performed in a controlled, manual fashion to enhance the steps and give the attacker knowledge of the process and how the system responds, so the timing part of the attacks is not taken into consideration here. In a real-life scenario, the steps taken here, usually, is more or less fully automated to execute all steps within seconds, giving the system as little time as possible to respond to the attacks.

In each of the attack scenarios, the following structure will be used:

- Define prerequisites specific for this attack if outside of standard scope
- Define goals
- Define attack tree
- Define attack methods

5.2.1 Phase shift attack

In this scenario, there is a desire to affect as much havoc and one-time permanent damage as possible, so in this part, there is no need to worry about logging, monitoring or altering of changed values - the effect of the attack will be pretty imminent and visible. The implementation part, does however contain parts where we wish to let the initiating control stations to be unaware of the fact that the specific station is compromised until the attack is complete.

By performing the "Phase Shift attack", an attacker can cause blackouts of a local sub-station, causing thousands of connected subscribers to be out of power for a period of time. There are fail safes in the grid that will re-route electricity through alternative routes, but such an attack will leave the grid vulnerable to smaller outages, or can be a part of a cascading attack as described by [34].

If the attacker is successful in destroying a phase shifter, getting a replacement will also take a long time, it is huge and costly - but this is currently outside the scope of this attack scenario as there does exist mechanical fail safes to prevent this.

Based on the topology in figure 7, an attacker has gained control over an Engineering workstation in Level 3 of the topology. This is a authorized machine that has access to a SCADA server/HMI placed in Level 2 of the topology

This SCADA server/HMI has control over a IED in Level 1 of the topology that controls the actual phase shift generator operation in this substation of the regional grid. By manipulating values in the commands sent, the attacker is able to tell the on-tap changer to alter the phase angle of the generator, and this should immediately be visible on the generator. A set of potential attacks to gain the same result is displayed in figure 8, but the highlighted route is the one this scenario is based on.

The reference model for a Phase shifter has been taken from the example "OLTC Phase Shifting Transformer (Phasor Model)" from Matlab [42] and modified to suit our need and can be seen in Appendix A

Attack goal: Cause disruption in the sub-station by causing a phase shift generator to not sufficiently adjust the phases or overcompensating the angle shifting, potentially causing overload. This is done by sending manipulated commands from a central SCADA server/MTU/HMI to an IED that operate this phase shifter.

Attack methods used:

- "Man in the middle" (MitM) as seen in section 5.1.1
- Rerouting traffic from the Engineering workstation/MTU to the IED as seen in section 5.1.1
- Replay the commands sent with changed variables and submitting these commands directly to the IED, pretending to be the actual control software, as seen in section 5.1.1

Metrics/actions recorded

- Status of the metrics from the IED controlling and monitoring the phase shifter
- Output from the simulation program getting the values from the IED mentioned above

- Output from the MitM program black holing and replaying the traffic sent from the Engineering workstation/MTU

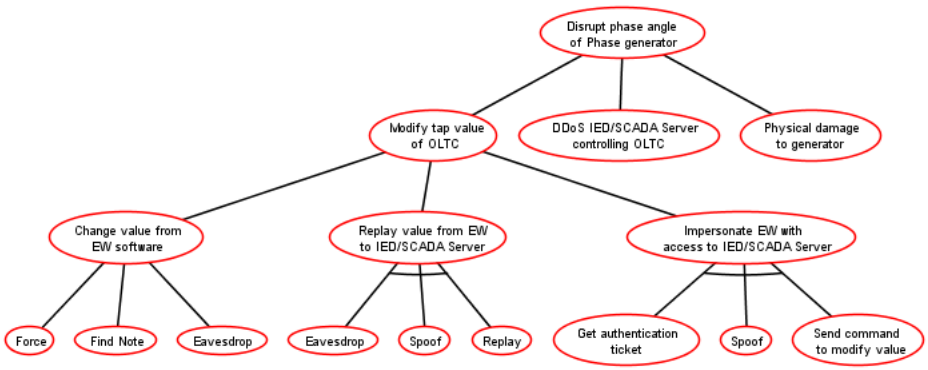


Figure 8: Attack tree - Scenario 1 - Phase Shift attack

5.2.2 Voltage conversion attack

In this attack, the goal is to see how attacks on a IED that handle voltage conversion affect the operations in a sub-station. By performing a voltage conversion attack, an attacker can potentially cause blackouts or destroy physical equipment. Additionally, basic network attacks will be performed between the different components in this scenario, trying to render parts of the sub-station inaccessible to remote operations, performing a set of DoS attack.

To be able to perform this, an attacker must first gain access to a SCADA server/MTU/HMI or direct access to the network where a IED is placed.

The reference model for a Voltage Conversion sub station has been taken from the example "VSC-Based HVDC Transmission System (Detailed Model)" from Matlab and modified to suit our need [42]. The details can be seen in Appendix A The modification of this model was outside the scope of my knowledge, so modifications to parameters were based on examples in the model, where an attacker successfully disable the DC balance control of the operations on the VCS Station 2.

Attack goal: In this attack, the goal is to see how an attack on voltage conversion affect the operations. For this example, the attack is aimed at disabling the DC balance control, effectively delaying balancing of the voltage conversion performed in the sub-station. When this attack is coupled with other attacks in the infrastructure, inefficient balancing of power can affect the stability of the sub-station. Additionally, a set of DoS attacks will be performed to visualize how this affect

response time and jitter in the network.

Attack methods used:

1. MitM/Replay attack, sending "fake" commands to the IED pretending to be the SCADA server or MTU to cause disruption in the grid.
2. Delay attack, utilizing the same MitM approach but not altering the commands sent, just delaying them to degrade service in the grid.
3. DoS, either of the IED or the HMI. This can effectively prevent HMI to send messages to the IED or the IED to be unable to respond to incoming commands from the HMI, leaving the HMI/IED as a standalone device unable to alter to the desired state of an operator. IEDs usually contain some logic to automatically mitigate issues, so this might be combined with a DoS attack on the HMI/MTU/SCADA Server to prevent it from responding.
4. Exploit vulnerabilities in the operating system of the IED, effectively gaining direct access to the IED and attacked processes. This allows an attacker to alter the voltage conversion directly and to block other operators from accessing the device.

Metrics/actions recorded

- Status of the metrics from the IED controlling and monitoring the voltage conversion unit
- Output from the simulation program getting the values from the IED mentioned above
- Output from monitoring of the network between EW and HMI.

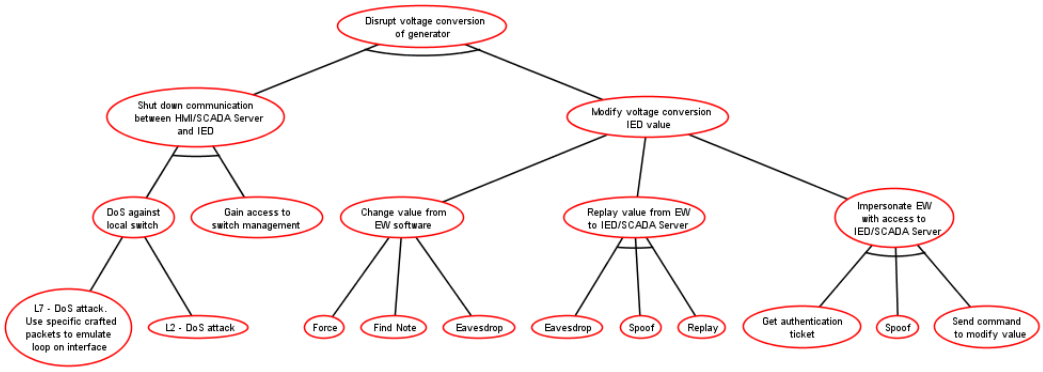


Figure 9: Attack tree - Scenario 2 - Voltage conversion attack

5.3 Mitigating attacks

In this chapter, a set of basic steps to minimize attack surface and hardening of systems will be described in general. Some steps are based on general best practices and some are more tuned to ICS systems. When it comes to mitigation of attacks, the ICS Cyber Kill Chain [2] can be used as a

reference. It maps out the steps that must be performed for an attack to be successful. Effectively eliminating one of the steps is, usually, sufficient to prevent an attack from happening.

In our scenarios, we have assumed that Stage 1 as seen in figure 10 is complete and that the attacker is inside our system, and are focusing on the steps of Stage 2 as seen in figure 11. Here the attacker performs actions on our system. It is assumed that the attacker before this stage, has developed and tested the attacks and knows to a certain level that the methods utilized will be successful. This is,usually, done in smaller increments and as stealthy as possible to avoid detection.

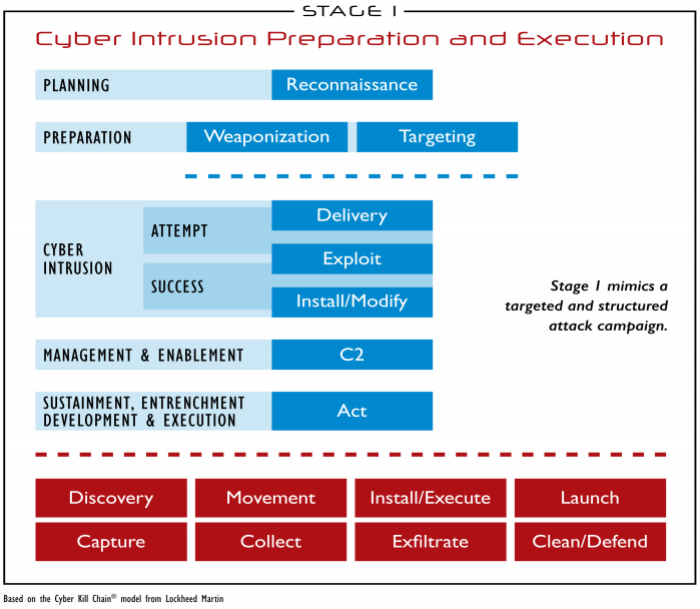


Figure 10: ICS Cyber Kill Chain - Step 1 [7]

5.3.1 Steps that can be taken mitigate attack surface in a ICS

As suggested in [6] and [43], there are a number of steps that can be taken to minimize the risk of attacks happening and containing them when they occur. Mitigating attacks at whole is considered an impossible task, so there is an understanding in the cyber security realm that we mitigate and harden as much as one can, and have contingency plans for when it occur.

Minimize or eliminate access to ICS system from external networks

To prevent unauthorized access to critical components, minimize the available paths to the system by hardening policies, firewalls, user access and entry points in general to ICS systems. When possible, keep the systems as offline as possible and never allow direct access to ICS systems from insecure networks.

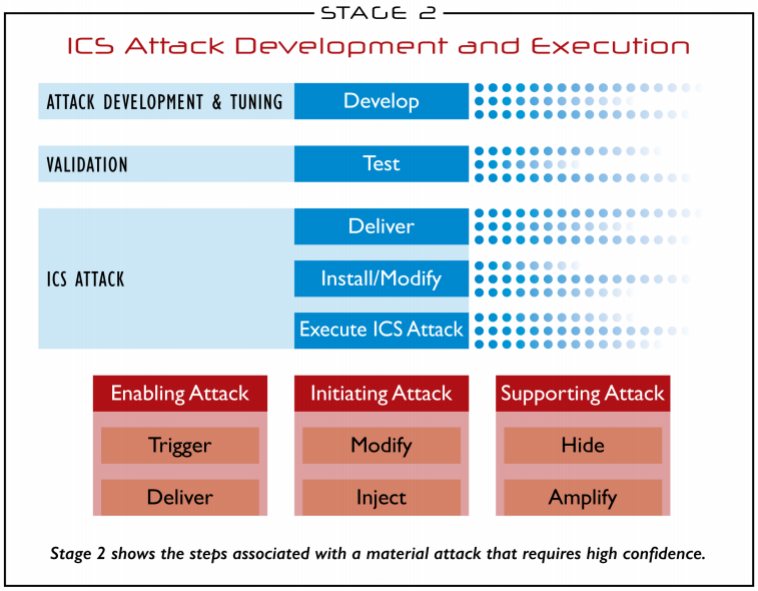


Figure 11: ICS Cyber Kill Chain - Step 2 [7]

Segment networks and implement firewalls between them

Segment networks into applicable groups of systems. Systems that naturally communicate with each other within the same security level can be placed in the same segment, while external support systems can be placed in another. Plan and design networks according to policies and classifications of data. Keep user activities separate from servers as much as possible and harden the type of traffic that is allowed to traverse segments. Define the possible paths from one segment to another when it is deemed necessary that traffic must flow. This gives the defender of the systems the possibility to contain an attack and limit the potential ramifications of such an attack.

Activate and harden firewall on devices, where applicable

Implemented with the section above, this gives additional security within a network segment. By activating a local firewall on a system, it keeps the other systems within a zone unable to access services other than those that is specifically exposed. There are, however, systems in a ICS that does not have the possibility of enabling a local firewall due to limitations on operating system, hardware resources or time critical systems where even a basic firewall verdict on a packet can cause harm.

Allow only controlled and secure access to networks, communication devices and ICS systems

Having control over actions performed in a ICS is important, not only for forensic purposes, but for tuning systems like a IDS. This can help detect unusual traffic patterns in the network, potentially blocking an attacker before getting to deep inside the systems.

By enforcing strong password policies, scanning for the use of default passwords on units and

employing Role Based Access Control (RBAC), one can harden systems and prevent an attacker from escalating privileges.

Keep systems up current on updates and patches

To prevent attacks on known weaknesses in systems, it is important to keep them current on available updates and patches from vendors, be it software or hardware. Lagging on keeping the systems up to date, expose them to potential threats and exploits that can be utilized through normal access channels, leaving the vulnerable for attacks.

Implement policies and staff training on cyber security

The implementation and, even more important, enforcing of policies related to cyber security or security in general is an important step in the process of hardening systems. A password can only be strong if it is not revealed, and note under the keyboard of an operator with the password to a central unit in the network can be sufficient to compromise the system as whole.

Implement, execute and maintain a disaster recovery plan for all parts of the ICS

We all plan to be secure, but it is just as important to plan for disaster. When an attacker gains access to the systems, and note that I write when, it is just as important to know how to recover from such an attack. A response plan will reduce the time from an attack is performed until the system is fully operational again. Additionally, a well formed plan will limit the damages an attacker can perform, and just as important, it shows confidence in the fact that the business has assessed the risks and knows how to recover from a potential attack.

The plans should contain procedures for manual intervention of critical parts of the system, the possibility of segmenting parts of the network offline and focusing on getting critical parts of the systems online again as soon as possible.

Just as important as the plans themselves, there is a need to execute them on a regular basis, exercising the organisation so all teams know what to do and when.

6 Implementation and building the lab

In this chapter, the implementation and building of the actual system is documented in a high level detail. It starts off with a description of prerequisites, followed by a overview of components used and how they are configured and is completed with a description of the attacks performed in each scenario. This chapter covers research question 1 and 2, and partially question 5 as stated in 1.5. All source code and reference data/guides is available at a public Github repository for scrutiny [44]

6.1 Building the basis of the lab

Here, the initial sections give a general overview of the components used to build the lab. A little more detailed installation details will be provided in the "Implementations and executions" sections in the end of this chapter. Ultimately, the system should replicate a small, simplified, set of remote control components for a sub station in a regional electricity grid.

There are some prerequisites to building this, that it is assumed that the reader has knowledge about.

1. Basic C knowledge i.e. how to create, edit and compile with gcc.
2. Basic network knowledge, i.e. how to deploy a network with multiple VLANS and route traffic between them
3. Know how to install, configure and create virtual machines using VMWare Workstation and/or know how to initialize and install a Raspberry Pi v3
4. Know how to install and configure a basic linux distro (in our case Debian GNU/Linux 10 (buster))
5. Know how to create and launch Simulink models
6. Basic linux knowledge, sufficient to launch predefined scripts and edit parameter inputs

6.1.1 Network and Communication

The network is segmented into four tiers, tier 0 (T0) to tier 3 (T3) following the ISA95 standard [6], each level containing a higher security level in descending order. In this paper, a engineering workstation will be placed in tier 3 to simulate a control center, and a HMI/SCADA Server will be placed in tier 2. A tier 4 could be added for corporate business, but this is not relevant in this scenario, so it has been omitted, but is still visible in the reference architecture 7.

- T3 consist of the control center and operators, where monitoring and management of the substations is done.
- T2 is the substation tier, mainly consisting of a local HMI, a historian and in most cases service stations for maintenance.
- T1 is the actual IED/PLC tier, where communication to and between the IEDs is performed.

- T0 is the physical tier with actual sensors, breakers, merging units etc. In this thesis, this is simulated in Matlab/Simulink

In this case, the actual segmentation is done by utilizing VLANs on a single switch, performing logical, but not physical segmentation. In the initial setup, local routing between the VLANs is configured and a simple, stateful, firewall with a set of defined rules added, allowing communication to flow between the networks. The networks are configured using 24 bits sub-nets, allocating 254 addresses for each segment, which in this case is more than sufficient.

Network details:

Tier	**Network**	**VLAN**	**Comment**
Tier 3	10.40.40.0/24	VLAN 100	Control center and operators
Tier 2	10.40.50.0/24	VLAN 200	Substation network
Tier 1	10.40.60.0/24	VLAN 300	IED network
Tier 0	10.40.70.0/24	VLAN 400	Physical network, actuators and simulations

Table 2: Network overview

Node details:

Name	**Function**	**IP Address**	**Network layer**
NTNU-EW01	Engineering workstation	10.40.40.10	Tier 3
NTNU-HIST01	Historian	10.40.40.11	Tier 3
NTNU-HMI01	HMI node	10.40.50.10	Tier 2
NTNU-IED01	IED node #1	10.40.60.10	Tier 1
NTNU-IED02	IED node #2	10.40.60.11	Tier 1
NTNU-SIM01	Simulink model - Phase Shift	10.40.70.10	Tier 0
NTNU-SIM01	Simulink model - Voltage Conversion	10.70.40.11	Tier 0

Table 3: Node overview

Additionally, an subset of machines used in the different attack scenarios will be added in the relevant places during the attack scenarios, and a subset of ports on the network switch will be used to perform port mirroring for analyzing packet flow. Wireshark, running on Debian GNU/Linux 10 (buster) will be utilized to copy traffic for replay/modification.

The basic flow of traffic in the setup can be seen in figure 12.

Firewall

As a firewall in this setup, a Ubiquti UniFi Security Gateway Pro 4 was used, as this already existed in my home network. The firewall was connected to the switch mentioned below and 4 new networks were configured. Between these networks, the firewall openings as described in table 4 were enabled.

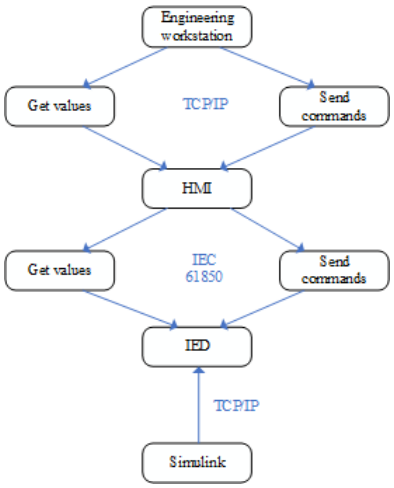


Figure 12: Communication flow from EW to Simulink

Source	Target	Port	Comment
10.40.40.10	10.40.50.10	TCP/22 TCP/6185	NTNU-EW01 to NTNU-HMI01
10.40.50.10	10.40.60.10	TCP/22 TCP/102 TCP/8165	NTNU-HMI01 to NTNU-IED01
10.40.50.10	10.40.60.11	TCP/22 TCP/102 TCP/8165	NTNU-HMI01 to NTNU-IED02
10.40.60.10	10.40.70.10	TCP/36880	NTNU-IED01 to Simulink - Phase shift
10.40.60.11	10.40.70.11	TCP/36881	NTNU-IED02 to Simulink - Voltage Conversion
10.40.60.10	10.40.40.11	TCP/8086	NTNU-IED01 to Historian
10.40.60.11	10.40.40.11	TCP/8086	NTNU-IED02 to Historian

Table 4: Firewall rules between VLANs

Switch

In this lab, a basic switch, HPE Officeconnect 1820 24 ports, was used. This switch support standard L3 switching and the possibility to enable mirroring of traffic out on a dedicated port assigned as a SPAN port.

The 4 VLANS described in table 2 was configured and a set of ports assigned to each VLAN, see table 5 for a overview. Port 1 was assigned as the uplink port to the firewall, all VLANS were then tagged on this interface, transporting the 4 VLANS to the firewall, enabling communication with and through the firewall.

6.1.2 Components

There are two, alternative, setups in this lab, as it has been configured in a way that it can be used in both a physical and a virtualized environment. The installation and configuration is fairly similar in both scenarios, it is just the interfaces used to configuration that varies.

The setup used to develop and test the environment, is purely based on a virtualized method,

VLAN	Untagged ports	Tagged ports	Comment
VLAN 100	Port 2 to 5	Port 1	Tier 3
VLAN 200	Port 6 to 9	Port 1	Tier 2
VLAN 300	Port 10 to 14	Port 1	Tier 1
VLAN 400	Port 15 to 19	Port 1	Tier 0

Table 5: VLAN assigned to physical ports

mainly due to portability and the possibility to work on the system "on the go". The additional bonus of snapshots for quick revert to a set system state makes this approach more desirable for a development environment.

The final implementation is based on a combination of Raspberry Pi's and virtual machines, where the Raspberry Pi's is used to simulate all components except the Simulink model and the engineering workstation - for use in a closed lab environment.

For simple replications and setup, it is a goal that the system should be as independent as possible, so dedicated hardware communication with external hardware will be limited to TCP/IP connections with the Simulink model. All simulations and reactions to attacks will be performed/simulated in software, which gives the additional advantage of being portable to actual HMI/IEDs as well - given that the input from the simulation model can be accepted by the system. In either way, it should be simple to modify the setup to facilitate new or additional components.

The components is as follows:

- 2 IEDs based on libIEC61850
- 1 HMI based om libIEC61850 and a custom developed front-end
- 1 SCADA server based on libIEC61850 - in this version the SCADA server and HMI is combined to one server/role
- 1 Historian/reporting server - This is based on InfluxDB (time-series database) and Grafana (visual display of metrics) for logging and displaying metrics from the IEDs and attacks.
- 2 Simulink models, displayed in the reference architecture 7 as 2 Simulink boxes running on 1 Simulink machine
- 1 Engineering station as the compromised system
- 1 switch/router with 4 VLANs
- 1 basic firewall with routing capabilities

IED

For the simulation of a IEC61850 IED, the library libIEC61850 [27] will be used to simulate a physical IED. This library provide a high-level IEC61850 API, and was chosen for its dynamic approach and its portability. According to the author, it supports independent MMS mapping such tat it will be capable to handle other mappings in the future (SCSM is specifically mentioned) . Given its Hardware Abstraction Layer (referred to as HAL), it can be compiled to run on both linux, windows and macos. See figure 13 for details. It automatically generates the MMS device model out of the

Name	Function	Version	Comment
VMWare Workstation Professional	Virtualization software	15.5.1	
Debian GNU/Linux 10	Operating system	Kernel 4.19.0.6	Minimal install
libiec61850	IED Server/Client	1.4.2	TCP/IP server added
OpenSSH Server	SSH access	7.9p1	

Table 6: Components used virtualized system

IEC 61850 data model. Additionally it provides support for control model, log service, data sets and reporting. It is written in C, but a C#.NET branch has been created, but with limited support compared to the C library. This IED Server is based on the C implementation and can be referred to as the "User provided server application" in the layers of figure 13.

The compilation and development of the IED server was done on a development machine (not depicted in the reference architecture) and later copied to the respective virtual machine and Raspberry Pi v3 in the physical implementation.For this process, the libIEC81650 library was compiled on a Debian GNU/Linux 10 (buster) x64 machine following the guides here [45]. Support for GOOSE and SSL was not implemented in this scenario, but should be considered in "future work" for higher security in inter-communication between IEDs where that is applicable. The choice between virtual and Raspberry Pi was coincidental, as there was a wish for a physical implementation, but it can just as easy be done in a fully virtualized environment. The various components and software used in the implementation is described in table 6 and table 7.

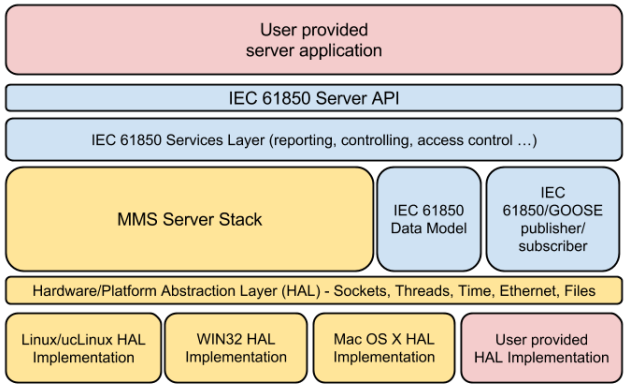


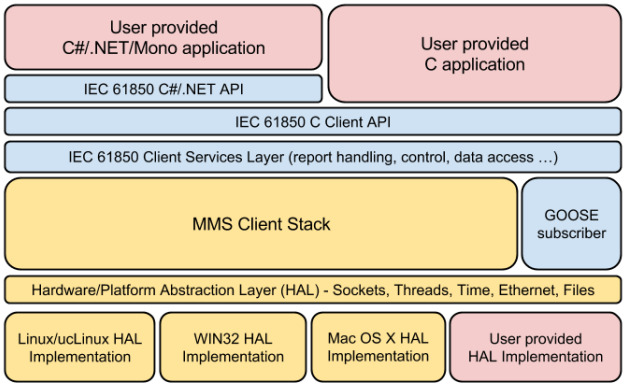
Figure 13: Software layers - libIEC61850 Server

IED Client - HMI/SCADA Server

For the IEC 61850 clients, the libIEC61850 library [27] was, again, chosen. The client implementation support model discovery and can read/write variables to the server. Additionally, it supports reporting and control services. See figure 14 for details. Like its sibling, the server client, it is written in C, but a C#.NET branch has been created. This IED Client is based on the C implementation

Name	Function	Version	Comment
Raspberry Pi	Physical unit	v3	4GB RAM, Power supply
Raspbian Buster Lite	Operating system	February 2020	Minimal install
libiec61850	IED Server/Client	1.4.2	TCP/IP server added

Table 7: Components used physical system



!h]

Figure 14: Software layers - libIEC61850 Client

and can be referred to as the "User provided C application" in the layers of figure 14.

The IED Client actually act as both a IED server and an IED Client, accepting commands from EW and relaying the command to a given IED. Additionally, the HMI displays current values of the underlying IEDs, giving a local operator the possibility to interact. For this implementation, two basic C programs was developed, one client and one server part. The server program is listening for commands from an EW and forwarding the commands to the IED, while displaying the operations in a text based console. The client program is a text based program where an operator can trigger predefined operations from the console.

The compilation and development of the IED client was done on a development machine (not used in the reference architecture) and later copied to the respective virtual machine and Raspberry Pi v3 in the physical implementation.

Engineering workstation

For the engineering workstation, a Debian GNU/Linux 10 (buster) x64 were configured with a graphical desktop. A default installation of Debian was chosen, and the IED client software were copied to the machine and executed. The EW is interacting with a predefined HMI, connecting to it through standard TCP/IP, sending control commands to the HMI, which in turn triggers the underlying IED.

Historian

The Historian is running Debian GNU/Linux 10 (buster) x64 server. On this server, a Influx time series database was installed and a Grafana graphical front end for displaying metrics.

The base operating system was installed with defaults and a static IP assigned in the relevant VLAN. Later a standard installation of a Influx database server was installed, following the install instructions from the manufacturer. On top of this, a graphical representation layer, called Grafana, was installed to present metrics and graphs for analysis. The installation procedure can be seen in listing 1. The client/server programs described in 6.2 was configured to forward all metrics and commands executed to the Historian for traceability.

Listing 1 Command to install InfluxDB and Grafana

```
# Download Influxdb repository
wget -qO- https://repos.influxdata.com/influxdb.key | sudo apt-key add -
source /etc/os-release
# Install InfluxDB from repository
sudo apt-get update && sudo apt-get install influxdb
# Start InfluxDB service
sudo service influxdb start
# Enable the http endpoint by editing influxdb.conf
# Set enabled = true under the [http] section
sudo nano /etc/influxdb/influxdb.conf
# Restart the service
sudo service influxdb restart
# Download Grafana
wget https://dl.grafana.com/oss/release/grafana_6.2.2_armhf.deb
sudo dpkg -i grafana_6.2.2_armhf.deb
# Install Grafana
sudo apt-get update && sudo apt-get install grafana
# Start the grafana service
sudo service grafana-server start
# Set the service to start at boot
sudo update-rc.d grafana-server defaults
# open GUI with your browser and start configuring (not covered)
http://<ip-address-of-historian>:3000
```

Wireshark

The Wireshark client is running on a Debian GNU/Linux 10 (buster) x64 GUI client with Wireshark installed. Installed components can be seen in listing 2

6.1.3 Simulink

For the physical power system, Matlab/Simulink was chosen as the simulation software, as purchasing and building complete replicas of a physical substation and power grids was outside the scope

Listing 2 Command to install Wireshark and allow non-sudo suers to perform captures

```
# Install Wireshark
sudo apt-get install wireshark
# During installation, allow non-sudo users to perform captures
# Add the user that will perform captures to the group "wireshark"
usermod -a -G wireshark your-user-name
```

of this thesis. In this thesis, there are three attack scenarios, where each scenario targets different parts of the infrastructure, rendering the need for three separate models. To simplify switching between the different attack scenarios, the models are kept quite simple, with just enough components to facilitate the actual process of triggering changes and a baseline of data sent to the IED and HMI. The Simulink models are available in Appendix A or as executable models stored on the Github repository here [44]

Phase shift

The Simulink model of the Phase Shift model is based on the OLTC Phase Shifting Transformer (Phasor Model) by [42]. No major changes is done to the simulation, beside adding TCP/IP input/output to the model and let this direct the stepping of the OLTC.

The model itself consist of two 120 kV 1000 MVA networks that are connected through a a phase shifting transformer (PST). The model allows for the phase shift to be changed through on load tap changers (OLTC). As the author describes it, the impact of triggering the OLTC can be observed by viewing different traces in the model. These traces is, in turn, also sent to the IED for logging purposes, and further from the IED to the Historian. The model allows a phase angle from 0 to 32.2 degrees, but can be modified to allow even steeper phase angles. Initially, the model starts with the OLTC at 0 degrees, with now power flowing as both sides are symmetrical. As the phase shift is increased, bus B2 will be lagging bus B1, causing power to flow from right to left. This can be seen as loading the power from one busy line to another. The 5, original, traces shows tap position (Trace 1), superposition of positive-sequence voltages measured at bus B1 and bus B3 (trace 2), phase shifts of positive-sequence voltages measured at output terminals (trace 3), comparison of the active power measured at bus B1 and bus B3 (trace 4) and comparison of the phase currents at bus B1and bus B3 (trace 5)

Voltage conversion

The Simulink model for the Voltage Conversion model is based on the VSC-Based HVDC Transmission System (Detailed Model) by [46]. AS with the Phase Shift model, no major changes was done to the model, except for the addition of a TCP/IP stream input/output module for the interaction with the IED to facilitate the changing of DC balancing trigger. This model contains vast opportunities for interaction, but a simple switch was chosen.

The actual model is a Voltage-Sourced Converter (VSC) which contain rectifiers and inverters, as well ac step down transformer, ac filters, converter reactor, capacitors and DC filters. It allows

control of converter active and reactive power output and simulate a 75 km cable between two sources.

6.2 Development of IED programs

6.2.1 IED server

For development of the software running on the IED, the server_basic_io example [27] from the libIEC61850 library was chosen as a baseline and later modified. See the Gitlab repository [44] for detailed code. As this program need to communicate with a Simulink model that is simulating the physical part of this CPS, a simple TCP server and client were added to facilitate communication with the model. For this, the standard library for socket communication in C, <sys/socket.h>, was implemented and the TCP server/client written while using this tutorial as a starting point [47]. Both running IEDs utilize the same SCL model for simulating the IED; with the same senor values and components, and is to be considered a baseline for customizing an IED. A pseudo code example of the TCP server can be seen in Algorithm 1.

Algorithm 1 IED - Input/Output TCP Server

```
1:  while input = true ... do
2:      if source =matlab..., then
3:          Analyze values against policy θ_P1 in environment E
4:          Store values in dataset θ_DataSet1 in environment E for T time sets
5:          Forward values in dataset θ_DataSet1 to Historian
6:      else if source =HMI..., then
7:          Change value of variable V1 in environment E
8:          Store value V1 in dataset θ_DataSet1 in environment E
9:          Send value V1 for environment E to Matlab/Simulink
10:         Forward triggered change with value V1 in environment E to Historian
11:     end if
12: end while
```

The IED server_basic_io example, by default, generate 4 SIN waves that is use for output in the data sets. This was disabled, and actual values from the Simulink model was implemented over TCP. Additionally, various input modules were implemented in the different Simulink models to receive input from the IEDs over TCP. In this program, the control model of it allows for controlling the IED, this is a standard feature of the libIEC61850 library and works for all data objects that is a controllable CDC, like SPC, DPC and APC as described in 4.2.4. This is used in operations towards the Simulink model, i.e. when a on-load tap changer (OLTC) is to be triggered. Data sets are used in this program to store logs/reports that can be sent to an Historian or HMI. From a HMI/MTU point of view, direct polling or custom data sets can be created for data transfer. This provides a dynamic way for clients of the IED to connect to it and create its own data sets, making the model flexible and customized. The library also support basic client authentication, currently only with a predefined password and is implemented in this program. The program it self is configured to

start when the IED is powered on, and runs in a loop until terminated, listening for values from the Simulink model and connections from IED clients - here defined as HMI, Engineering Workstation and Historian.

From Simulink, a TCP connection is made over a predefined port and a set of values is sent as a string, representing various data from the Simulink model. Currently, it consists of 2 integers and 8 float numbers separated with space. All values is not implemented in each scenario, so the data sent from Simulink has been expanded to facilitate all scenarios, making the program as generic as possible. The values are then mapped, based on position in the transfer to the representative data objects in the SCL model, and stored in a data set.

Raw Output from Simulink - Scenario 1: From client: 1 1 0.998153 0.998123 -6.366281 -6.367136 -54.559440 -54.574482

Name	Value	Comment
Tap position 1	0	Position of Tap 1
Tap position 2	0	Position of Tap 2
Sensor 1	0.857157	Value of Sensor 1
Sensor 2	0.029719	Value of Sensor 2
Sensor 3	-0.825042	Value of Sensor 3
Sensor 4	-0.921263	Value of Sensor 4
Sensor 5	0.65842	Value of Sensor 5
Sensor 6	0.86214	Value of Sensor 6
Sensor 7	-0.25325	Value of Sensor 7
Sensor 8	-0.09325	Value of Sensor 8

Table 8: Examples of values sent from Simulink to IED in Scenario 1

From the IED, a set of 2 binary values is sent to the Simulink model, mainly representing set points for triggering components in the model, representing the physical part of the CPS simulated here.

Name	Value	Comment
Tap 1	0	Increase value of Tap with 1
Tap 2	0	Decrease value of Tap with 1

Table 9: Examples of values sent from IED to Simulink in Scenario 1

6.2.2 IED client - HMI

For development of the IED client, used in the HMI the predefined example called libiec61850_client_example_4 was used and modified to use the same input/output as in the IED server model. The client connected to a predefined IED, and different variations of the clients were developed, customized to each function and to facilitate each Simulink model/attack scenario.

The client for the HMI was developed to facilitate controlling of the IED, while the Historian only gathers predefined reports based on data sets from the IED for logging purposes. All clients

have implemented basic authentication against the IED. The program is configured to start when a user initiates the program, and runs in a loop until terminated, displaying values from the IED server based on the client configuration. An example of values displayed on the HMI can be seen in figure 15.

Listing 3 Code snippet - Display values from IED

```
if (error == IED_ERROR_OK ) {
            /* read new data set */
            ClientDataSet clientDataSet;
            clientDataSet = IedConnection_readDataSetValues(con, &error, "simpleIOGenericIO/LLN0.AnalogueValues", NULL);

            if (error == IED_ERROR_OK) {
                                while (newDataSetEntries != NULL)
                                {
                                        printDataSetValues(ClientDataSet_getValues(clientDataSet));
                                }
                                ClientDataSet_destroy(clientDataSet);
            }
            else {
                printf("Failed to read data set (error code: %d)\n", error);
            }
            IedConnection_deleteDataSet(con, &error, "simpleIOGenericIO/LLN0.AnalogueValues");
        }
        else {
            printf("Failed to create data set (error code: %d)\n", error);
        }
```

Additionally, a control part for the IED created for triggering components in the Simulink models. This is based on the predefined example called libiec61850_client_example_control where a basic TCP/IP server was created to listen for input from the EW on a specific port. The client then receive a standard TCP/IP packet with a object to connect to and a value to change, based on this, the correct IED is selected and a control message sent to the IED as a MMS packet from the HMI. So in this instance, the HMI also act as a SCADA server, combining the roles. In listing 4 a code snippet from the code used to trigger and send MMS commands to the IED from the HMI.

See the Gitlab repository [44] for detailed code.

```
From client:1
1
0.998153
0.998123
-6.366281
-6.367136
-54.559440
-54.574482
```

Figure 15: Values printed from IED with OLTC set to 1

Listing 4 Code snippet - Control IED

```
// Function to process data coming from the EW
void hmi_processor(char buff[MAX_BUF], char iedIP, int iedPort)
{
                // Process the incoming TCP/IP data and perform actions on a given IED
                printf("Inside EW Processor\n");
                char * pch;
                // Split the buffer, as multiple values can be passed
                pch = strtok(buff," ");
                while (pch != NULL)
                {
                                printf("Value received: %s\n", pch);
                                // Send command to the HMI program to perform action on IED
                                sendCommandToIED(iedIP, iedPort, pch);

                        pch = strtok(NULL, " ");
                }

                // Zero out the buffer
                bzero(buff, MAX_BUF);
}
```

6.2.3 Engineering Workstation client

For the development of the program facilitating interaction between the EW and the HMI, a basic C program was developed, consisting of a basic TCP/IP client which connects to the HMI. There are two programs developed, one client that reads incoming values from the HMI and one that interacts with the HMI, sending commands for specific IEDs controlling the simulink models. The program is tailored to the scenarios and provides a simple, menu choice based interaction with predefined actions that can be performed against the HMI over a predefined port. Examples of the menu choices and interactions can be seen in figure 16 and 17

```
Choose IED to operate on
-----------------------------
 1. NTNU-IED01 - Phase Shift
 2. NTNU - IED02 - Voltage Conversion
 3. NTNU - IED03 - Smart Switching
 4. Quit

Enter IED number
```

Figure 16: Engineering Workstation - Interaction with HMI

6.3 Implementation and execution of attack scenarios

To implement and execute the scenarios, the following sub-tasks were defined and completed in sequence to form the basis of the lab, building on the reference architecture:

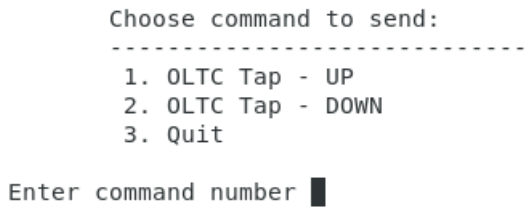


Figure 17: Engineering Workstation - Interaction with HMI - OLTC stepping

- Build and modify the Simulink models [42] [46]
- Implement TCP/IP communication in the Simulink model for external communication (send/receive) [48]
- Compile the libIEC61850 base library [27]
- Build and customize the libIEC61850 client/server to utilize as a IED server/client [27]
- Implement TCP communication in the libIEC61850 client/server to communicate with the Simulink model [47]
- Build and customize HMI to display metrics and configure the Historian for logging
- Build and customize a basic control program for the HMI, used from the EW
- Copy the programs to the respective IED, HMI or Engineering Workstation and initiate them
- Start the Simulink models to generate data

6.3.1 Attack scenario - Phase shift transformer attack

The physical power system consist of a substation where a phase shift occur is modeled and simulated in Matlab/Simulink. It consists of two incoming sources with varying phases, and a phase shift is to be performed. This is a time critical operation, and need to be monitored and performed in a automatic way, usually handled by the IED itself. The model sends continuous data to the IED with measurements over TCP/IP, and the IED is to perform automatic remediation based on the measurements. A breaker can be triggered manually from the control center/HMI which can cause damage to the physical environment, this is the component we are to trigger.

Attack the flow between EW and HMI: In this scenario, the attacker has managed to install a packet capture software, Wireshark on the EW. This is then used to capture the current state of the HMI and replaying these values back to the EW, currently not modified.

A message is then constructed by the attacker, specifically crafted to change the TAP value on the systems phase shifter from 0 to 5, which will lead to a surge in the grid. The attacker sends this crafted message from the EW, pretending to be a valid operator, and in the same process spoofs the values sent from the HMI to the EW, pretending that everything is OK. This allows the attack to proceed until the SCADA server picks up on the alerts triggered by the local IED monitoring/controlling the bay.

Programs used:

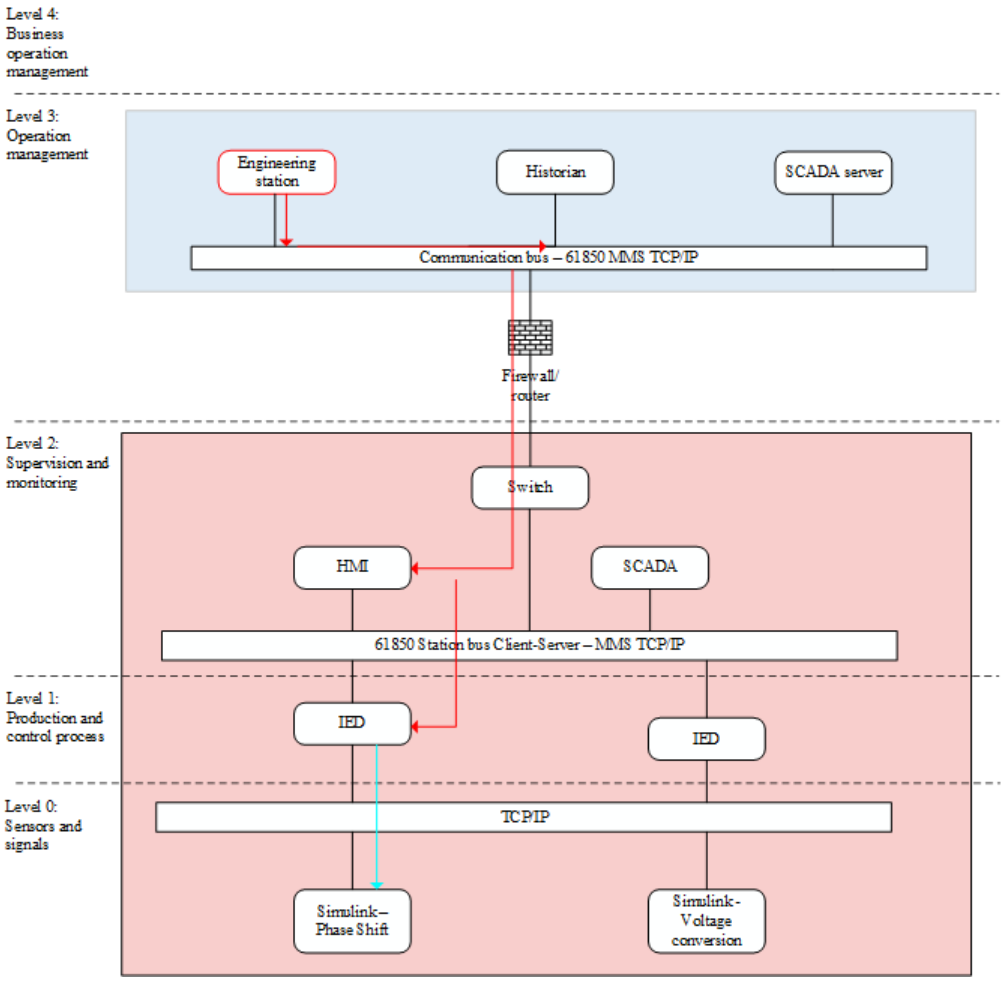


Figure 18: Attack flow - Scenario 1

1. Wireshark
2. putty or equivalent SSH utility
3. nano or equivalent text editor
4. PlayCap
5. mitm_script
6. Burp Suite Invisible Proxy

Step 1 Capture traffic between the EW and the local HMI with Wireshark. In our instance, we are using the graphical interface of Wireshark, capturing all traffic between the EW and the HMI.

Step 2 Reroute or black hole traffic from EW to HMI by routing traffic from EW to HMI to a locally

installed, spoofed HMI instance. In this case, the use of iptables allow the DNATing of traffic outbound on the machines interface.

Step 3 Send captured traffic from HMI to EW in loop by utilizing values from the Wireshark capture and let a, locally installed, server emulating the HMI send the values in loop. This is, of course, dependent on being able to capture the current state of the HMI and replicate the setup. In this case, the embedded SIN waves will be send in loop to the EW.

Step 4 Alter the previous captured TCP/IP capture between the EW and the HMI to facilitate the desired output. This is done by altering the Wireshark capture and replay it using a tool, such as PlayCap. Alternative method for altering the values in transit is also displayed.

Step 5 Observe changed either in raw output from the IED or through logging tools.

Initial values: In the Matlab model containing phase shift, the OLTC values is set to 0 and as the two sources have their internal angles set at 0 degree. This implicates that there is no current flowing in the model. The compromised machine, EW, have the capability of controlling this OLTC. To prevent this operator from interfering with our attack, we "black hole" the traffic sent from/to the workstation, tricking the operator to assume everything is ok. We here assume that the operator is controlling an IED in the sub-station through commands sent to/from a local HMI as can be seen in the reference architecture 7.

Step 1 - Capture traffic between the EW and the HMI with Wireshark

To simplify the process, Wireshark was initiated from a separate computer, listening on a SPAN port. All traffic was then captured between the machines NTNU-EW01 and NTNU-HMI01, storing the captures in .pcap in files for later retrieval and replay.

Step 2 - Reroute/black hole traffic from EW to HMI

As we have full access to the machine NTNU-EW01, we can easily route the traffic destined for a specific sub-net or specific machine to a different place using a combination of commands. By creating DNAT rules, traffic can be redirected, tricking the machine to force all traffic going from NTNU-EW01 to NTNU-HMI01 on TCP/8165 back to NTNU-EW01 on TCP/8166 where the spoofed HMI server is running.

```
iptables -t nat -A PREROUTING -s 10.40.40.10/32 -p tcp --dport 8165 -d 10.40.50.10
-j DNAT --to-destination 10.40.40.10:8166
```

Additionally, we want to enable NAT and IP forwarding, allowing the remaining traffic destined for that host or other hosts to be allowed:

```
$ echo 1 > /proc/sys/net/ipv4/ip_forward
$ iptables -t nat -A POSTROUTING -o $EXTERNAL -j MASQUERADE
$ iptables -A FORWARD -i $EXTERNAL -o $INTERNAL -m state --state RELATED,ESTABLISHED -j ACCEPT
$ iptables -A FORWARD -i $INTERNAL -o $EXTERNAL -j ACCEPT
```

Step 3 - Send captured traffic from HMI to EW in loop

As mentioned in Step 1, to simplify the attack, we have here used Wireshark on a separate computer, listening on a SPAN port. All captured traffic has been stored in a log file for replaying of the traffic,

and it is these files we will use for the basis of the manipulated traffic. This traffic is then, at this level, manually replicated in the HMI and the HMI is started up acting as the original HMI, displaying values on the EW. Alternatively, in this scenario, the existing SIN waves that is generated by the program can be sent in loop to generate traffic to the HMI.

Step 4 - Alter captured traffic messages and send to HMI

In this step, we use the same, recorded, traffic from Step 1 and manipulate it. In the traffic, we see the communication between our EW and the HMI. Even if the EW haven't changed the tap value of the IED yet, we can use basic knowledge of TCP/IP traffic and how the IEC 61850 standard to do some assumptions. The example in this instance is a simple trigger value that is to be changed, so the EW basically sends a 1 or a 0 to the HMI over TCP/8165. This allows us to send the altered values to the HMI based on previous traffic, by simply letting the EW send a 1 to the HMI 5 times.

Alternative method

We could replay communication between the EW and the HMI with altered values, but by expanding the process used in step 2, we can implement the use of a mitm_script and Burp Suite Invisible Proxy to make the process a little more graphical and educational.

By installing Burp Suite Invisible proxy [49] on the EW and using the script described in [50], we use the same port specified in step 2, but instead of sending all traffic to the spoofed HMI, we by default allow all traffic to pass through the mitm_scipt and Burp destined for the HMI. This give us a possibility to record all requests, and even rewrite packets on the fly, intercepting them before they are sent to the HMI. This process is made possible by the use of wrapping client/request responses in the body of a HTTP POST request and sending it to a basic web server initiated by the script. This request is then sent to the Burp suite Invisible Proxy and readable, here packets can be dropped or forwarded, either modified or unmodified. The whole process with examples can be observed in [50].

Step 5 - Observe changes in system

Observe the output and logs of the EW, IED and HMI for all steps. See chapter 7 for an overview.

6.3.2 Attack scenario - Voltage conversion attack

In this scenario, the attacker has, like in the previous scenario, gained control over an engineering workstation (EW) in level 3 of the architecture as can be seen in figure 9.

The voltage conversion system consist of a substation where a voltage conversion occur and is modeled and simulated in Matlab/Simulink.

The model sends continuous data to the IED with measurements over TCP/IP, and the IED is to perform automatic remediation based on the measurements, although this is not implemented in the model. A metric can be triggered manually from the control center/HMI which can cause the voltage conversion to change in the physical environment, this is the component we are to trigger.

Programs used:

1. putty or equivalent SSH utility
2. nano or equivalent text editor

3. hping3
4. PingPlotter 5

Using this EW as a jump station, the attacker has managed to get the credentials needed to access the HMI located in level 2 of the architecture, managing the IED controlling the voltage conversion operations of the Simulink model. This is done by performing a SSH connection from the EW to the HMI and executing a predefined script, altering the values of the DC conversion, causing the voltage conversion/balancing to be delayed which again can lead to blackouts on the sub-station. This is basically a libIEC61850 client script derived from the iec61850_client_control predefined script, and can be triggered to change values. The modified script is available in the public Github repository [44] where all code related to this thesis is located.

To further complicate the issue, the attacker, after gaining full control of the HMI, sends a specifically crafted packet through the switch in the level 2 of the architecture, aiming to effectively cause a DoS attack against the network of which the HMI is connected. Most switches contain a integrated DoS/loop-back protection, and we will try to utilize this to block or severely hinder traffic on the port where the HMI is connected. This should cause the EW to loose connection with the HMI. The local SCADA server is additionally unable to contact the HMI which controls the IED directly, needing physical attendance to the system to halt the attack. The drawback of this attack, is that the attacker is unable to do further operations on the system, as he is also unable to access it, but the goal is accomplished. To enable this attack, the attacker disables the checksum check of NTNU-HMI01 Ethernet card using the command in listing 5:

Listing 5 Command to disable verification of checksum on local interface

```
# Disable checksumming on eth0 of device NTNU-HMI01
sudo ethtool -K eth0 tx off20
```

We then execute the python described in listing 6, sending faulty packets to the switch.

As an alternative path to try and disable the port where the HMI is connected, we execute a standard hping3 smurf attack from the HMI to the IED, trying to render the system unresponsive or unstable to respond and to see the effect that had on the latency and the jitter between the HMI and the IED. The command in listing 7 was executed from the HMI against a IED node, installing hping3 on the on the NTNU-HMI01 machine in the process.

Additionally, for a final scenario, the port mode for NTNU-HMI01 is changed from 1Gbps duplex to 10Mbps half-duplex to mimic a low bandwidth link (3G) between the EW and the HMI. In this process, a hping3 syn flood attack is performed from the sub-station network against the control station network, in this instance effectively between NTNU-HMI01 and NTNU-EW01. This should result in overburdening the EW, causing it to drop connections after a while. With this command, we are occupying available ports on the EW where the three-way handshake of the TCP protocol.he command in listing 8 was executed from the HMI against a EW node, expecting hping3 to be installed from the previous attack. In this instance, -c defines the number of packets to send (10

Listing 6 Script used to send faulty packets to the switch

```
#!/usr/bin/env python

from socket import *

#
# Ethernet Frame:
# [
#   [ Destination address, 6 bytes ]
#   [ Source address, 6 bytes      ]
#   [ Ethertype, 2 bytes           ]
#   [ Payload, 40 to 1500 bytes    ]
#   [ 32 bit CRC chcksum, 4 bytes  ]
# ]
#

s = socket(AF_PACKET, SOCK_RAW)
s.bind(("eth0", 0))
src_addr = "\x01\x02\x03\x04\x05\x06"
dst_addr = "\x01\x02\x03\x04\x05\x06"
payload = ("["*30)+"PAYLOAD"+("]"*30)
checksum = "\x00\x00\x00\x00"
ethertype = "\x08\x01"
s.send(dst_addr+src_addr+ethertype+payload+checksum)
```

Listing 7 hping3 Smurf Attack [51]

```
# Install hping3 on the NTNU-HMI01
sudo apt-get install hping3
# Execute attack against NTNU-IED02
sudo hping3 -1 --flood -a 10.40.60.11 255.255.40.20
```

000), -d is the packet size (120 bytes), -S is enabling the SYN flag and -w specifies a TCP windrow size of 64. -p specifies the port where we should connect, in our instance there is a open SSH server we are able to reach (deviance from the firewall openings specified earlier to facilitate the attack) and –flood specifies the program to send the packets as fast as possible. The –rand-source parameter generate the spoofed IP addresses where the ACK should go, but will never reach.

In a complete environment, this attack would be done against the edge facing router/firewall where a published service would be. This attack was done to see if a compromised component in the sub-station could render the control center unable to perform actions on the sub-station by overburdening the link with traffic.

Although it is easy to implement safeguard against these attacks, they can be highly successful

in local networks, especially the Smurf attack in a sub-station environment, and where focus on securing sub-stations has not been given sufficient attention.

Listing 8 SYN flood attack

```
# Execute SYN flood attack against NTNU-EW01 from NTNU-HMI01
sudo hping3 -c 10000 -d 120 -S -w 64 -p 22 --flood --rand-source 10.40.40.10
```

The attack path is described in figure 9 where the red line is the initial attack from the EW to the HMI, and the purple line is the attack from the HMI against the switch/EW and the script running to change parameters on the IED.

Step 1 Gain control of the engineering workstation
Step 2 Alter values of voltage conversion using a script to complete the process
Step 3 Send crafted packets to the switch, causing the port to block or be severely degraded
Step 4 Observe changes, either in raw output from the IED or through logging tools such as PingPlotter

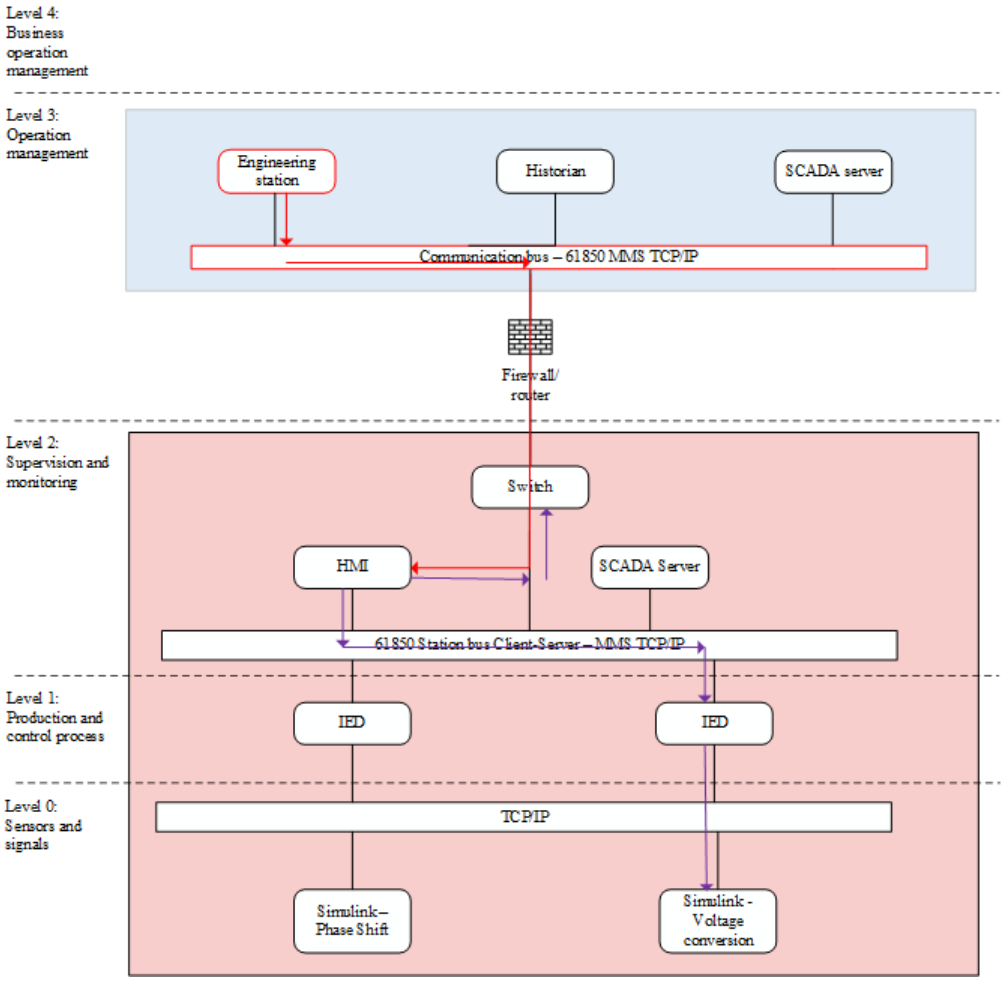


Figure 19: Attack flow - Scenario 2

7 Results

In this chapter, the results from the actual attacks will be presented and analyzed. At this point, the system is designed with a "least resistance" mindset when it comes to system security, and a natural evolvement of the scenarios would be to implement hardening of the systems to make the attacks harder to perform and modifications of the attacks to observe the results. This chapter partially covers research question 3 and 4 as stated in 1.5.

7.1 Results scenario - Phase shift attack

The assigned IED does no controlling of the actual signals/values coming from the model, as the model is confined to its own operating environment.

In figure 20 we see the normal operations of the system then the OLTC tap level is set to 0, indicating a angle of 0 degrees in other words, no current flowing.

By manipulating the value sent from the NTNU-EW01 to NTNU-HMI01, the value of the IED is, in turn, changed to "tap" the OLTC, which we in this case did three times. As we can see from the figure 21, the value of the OLTC is here accepted and changed from 0 to 3 after three tap operations, each change in the model take approximately 3 seconds as can be observed in the figure. This, in turn affect the flow in the substation from the two sources, which if done at the right time and with the right angle modification can potentially cause instability in the grid.

This should flag a response to the HMI with an alert. As we do not have alerting enabled in this scenario, we can observe on the HMI and in the running Simulink model that the change was done. In a scenario where altering is enabled, this alert would be captured and displayed on the local HMI, but since we are rerouting traffic sent to the control center (the EW in this instance), the operator will not see this alert.

There is, of course, a SCADA server involved in this process which will take action, so the attack is most likely a short termed success, and there might even be internal mechanisms in the IED/local HMI/SCADA server to prevent this from happening.

This is a natural evolvement of this paper, and will be discussed on conclusion/further work.

7.2 Results scenario - Voltage conversion attack

In figure 22 we see the balance operation of the voltage conversion in normal operating mode, effectively balancing the flow.

When the attacker modified the trigger for DC balancing and turns the function off, the model reacts slower to the balancing and conversion actions, making the system lagging.

This is, of course, just one of many operations that can be performed in an attack against this model, but has been chosen as the most feasible way to trigger and view the actions of an attacker.

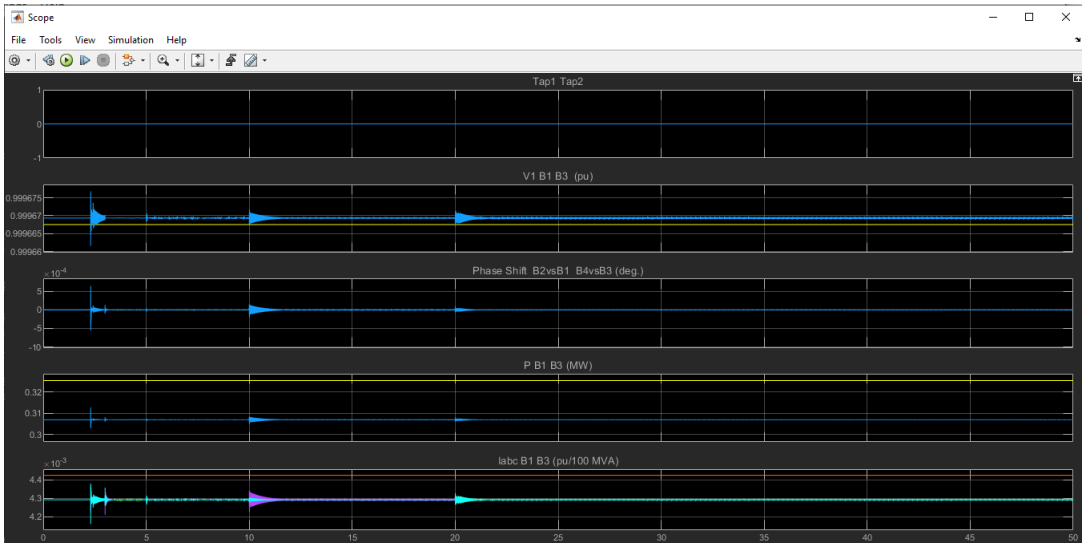


Figure 20: Values with OLTC set to 0

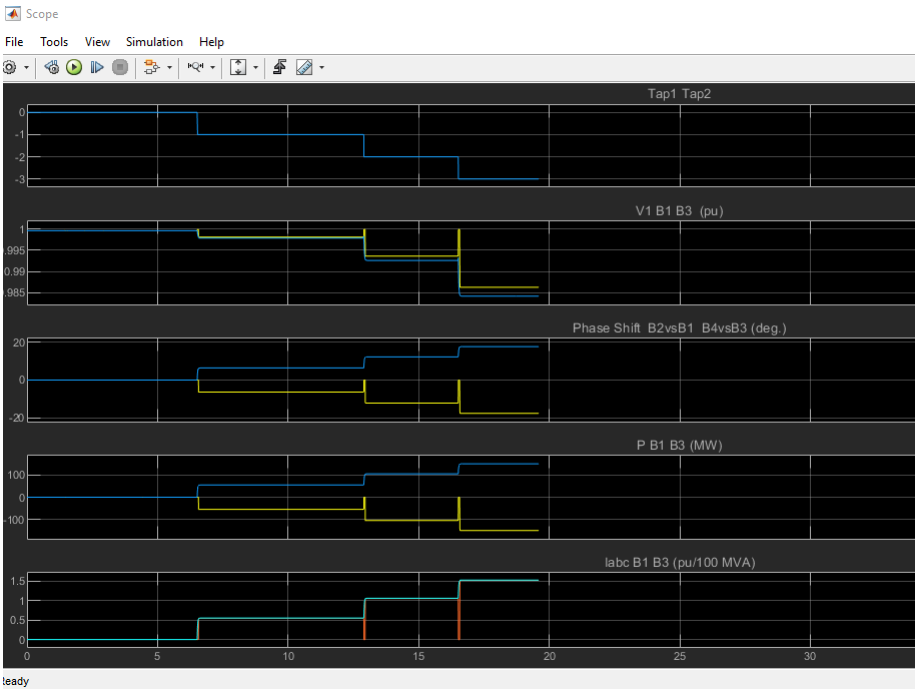


Figure 21: Values with OLTC "tapped" 3 times

As can be seen in figure 22 and 23 there is a noticeable variation compared to normal operations.

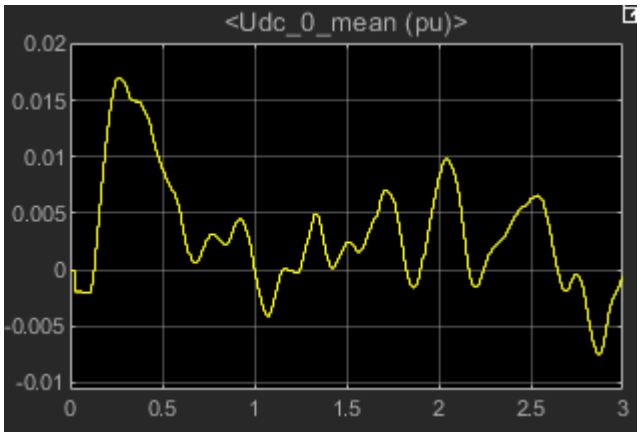


Figure 22: Voltage Conversion - DC Balance normal operations

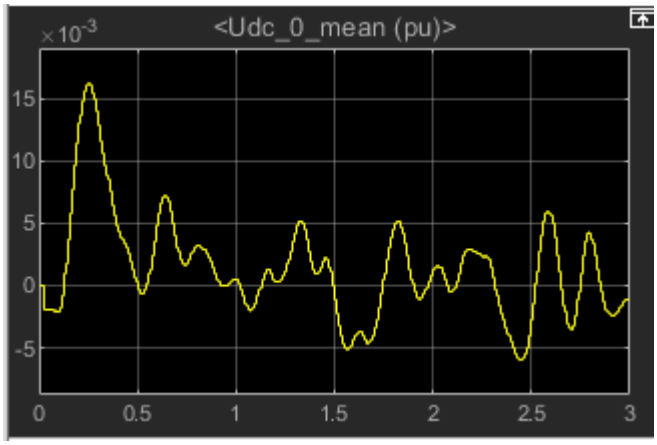


Figure 23: Voltage Conversion - DC Balance turned off

When it comes to the DoS attack, the attacker is able to access the operating system of the HMI, disable the checksum check of the units Ethernet card. After this, a Python script as seen in listing 6 is triggered.

This script creates a faulty packet and sends it to NTNU-IED02 through the switch, triggering Tx errors on the switch port. The Tx errors is observable on the switch management interface, but no changes in response time or jitter between NTNU-HMI01 and NTNU-IED02 can be observed from the monitoring machine. This can indicate that the processing power of the switch is sufficient to handle the attack and simply adjust for the Tx errors.

All data is recorded using Pingplotter from an external machine, monitoring the values of the

IED interface, plotting over a time period of 5 minutes. Additionally, metrics from the switch is gathered to see that the Tx errors actually occur.

After this, a basic hping3 attack was performed from NTNU-HMI01 against NTNU-IED02. By using the preinstalled hping3 program on the HMI, a standard flood attack was launched as described in section 6.3.2. The results before the attack can be observed in figure 24:

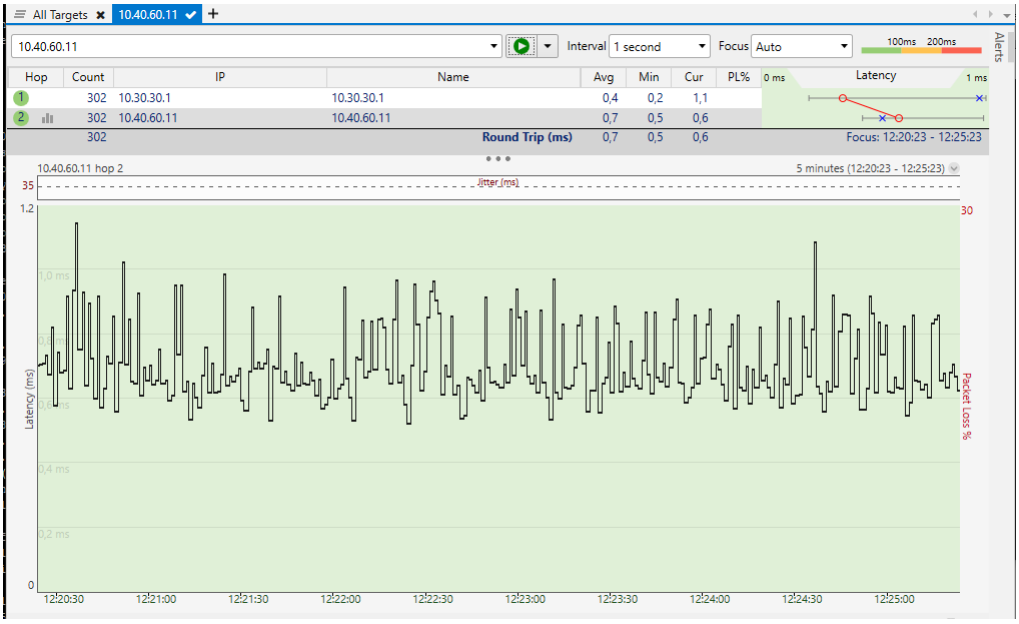


Figure 24: Response and jitter - Normal operations

As observed in figure 25, one can see that immediately after initiating the attack, the response time increased and packet drops occurred. This also affected the response time of the HMI network, Tier 2, in general.

To fulfill the simulation of a remote control scenario, a DoS attack utilizing the SYN flood attack was made against NTNU-EW01 from NTNU-HMI01, simulating the link between the sub-station and the control station - executed from the sub-station LAN. This caused the response from the simulated 3G (10Mbps half-duplex) link to be somewhat slower and can be observed when comparing the graph before in figure in figure 26 and during in figure 27. As the attack was performed from a single attack source, the effect is limited, but observable with a couple of larger spikes and, although marginal, higher average than before the attack, In a larger setting or utilizing a more powerful hardware than an Raspberry PI and with multiple compromised hosts, and internal attack in a substation could yield a more significant results. Also, if the operator has done insufficient hardening of their remote controlling network, either by exposing the MPLS routes to other customers or using publicly available networks over 3G, a network of zombie hosts could be used to perform a DDoS attack against the edge router, causing the whole sub-station to be unavailable for

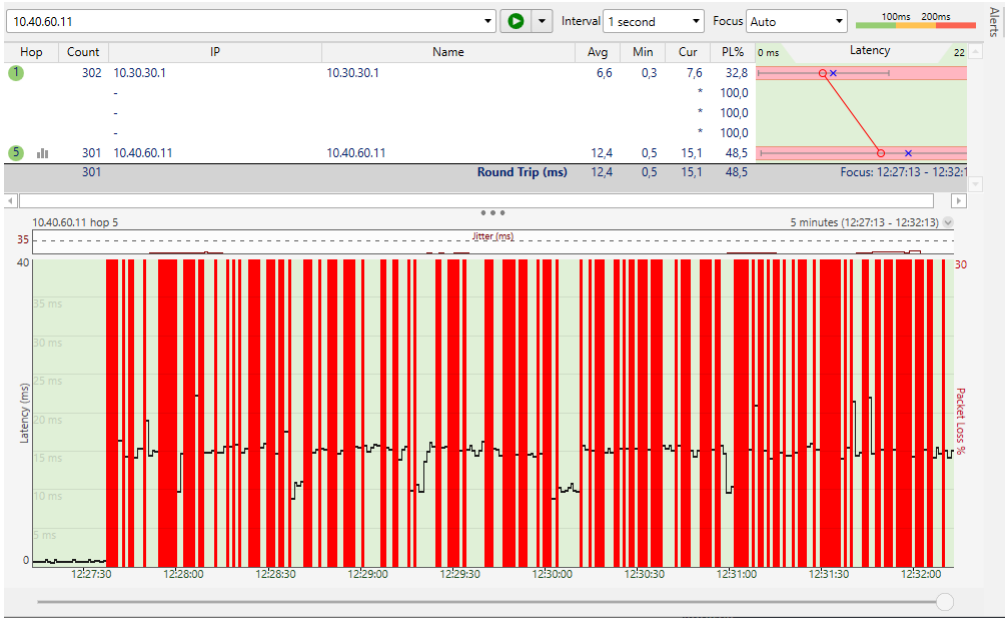


Figure 25: Response time - Under attack. Red bars indicate packet loss

remote operations.

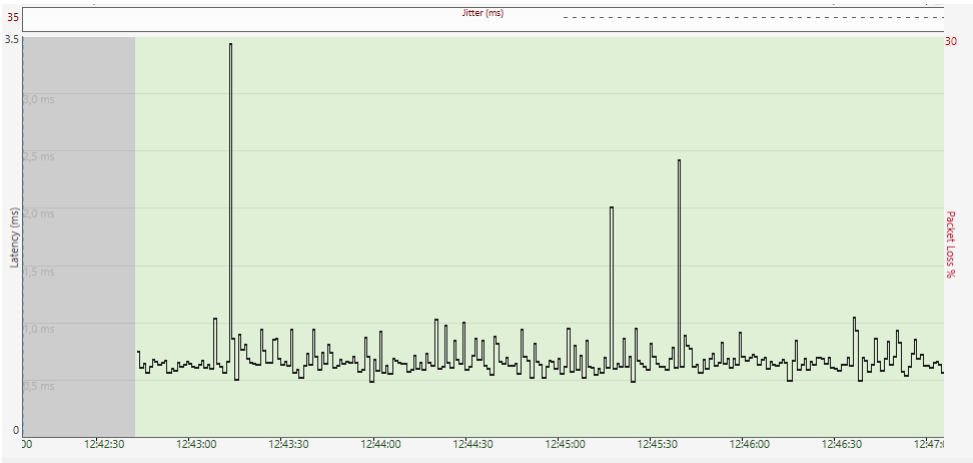


Figure 26: Response time - Before SYN Attack - Average 0,7ms, Minimum 0,4ms

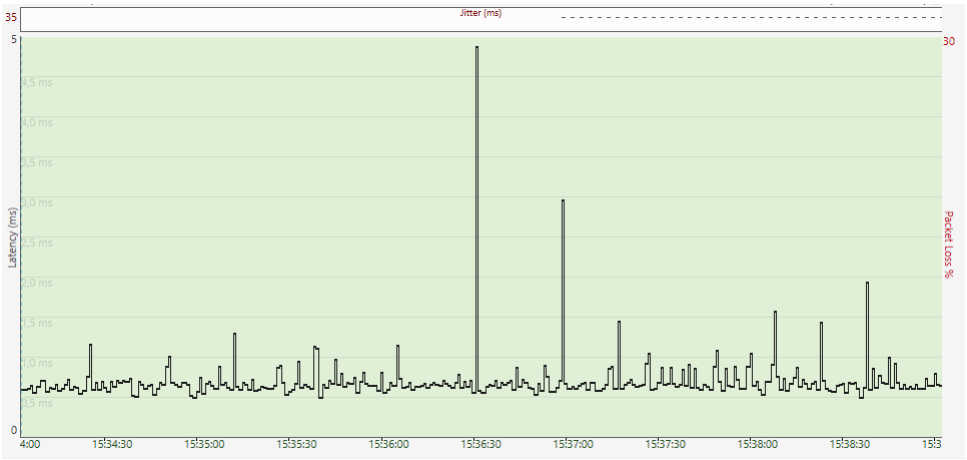


Figure 27: Response time - During SYN Attacks- Average 0,8ms, Minimum 0,6ms

8 Conclusion

8.1 Discussion

Building a test environment for a remote control scenario for the electrical power grid has given a good overview of the current state of the researched part of this vast field of knowledge. However, there are some challenges that has been blocking the path at times, mainly the lack of willingness to discuss and comment on production systems. These systems are, still, considered black boxes and there has been a lack of input from the stakeholders of these systems to give out details around existing topology and fundamental building blocks of the remote controlling of the grid. This thesis is, therefor, mainly build on existing research knowledge and components, with the limitations that applies here. The lab has some basic security functions that is missing, mainly noted is no IDS/IPS implemented and only basic firewalling enabled. While the firewall issue is circumvented by utilizing a compromised engineering workstation, there are other safeguards that can be implemented to further secure the environment. There are, for example, host based IDS/IPS that can be implemented to detect rouge agents on a host - and one should expect this on a host located in a control station environment to provide extra security to prevent this kind of attacks.

In addition to the physical implementation with low cost hardware, the lab can be created fully virtualized, giving the possibility to easily revert to the initial configuration with snapshots.

The different attack scenarios are build to be easily replicated and the results easy to read. These scenarios can be expanded and modified to suit a given set of components and parameters and is only to be considered as a starting point for a lab.

The program used for building the IEDs are based on open source software, with the possibilities and limitations that applies. Basic support is given through forums or email and the whole project is, currently, fully based on a single developer. But considering that it is open source, a fork could easily be created and expanded.

At the start of the thesis, the focus was put on the C#.NET implementation, as that is my preferred programming language. Early in the process however, a bug was discovered while customizing the program. This was reported to the creator of the libIEC61850 program. Although the bug was easily circumvented and fixed rather quick by the developer of libIEC61850, a shift in the development process of the IEDs was done to the C implementation. The shift was, mainly, done due to feedback during the bug report/fix process that the C#.NET implementation has limitations and was considered a side-project to the C implementation. All programs are tailored to suit each scenario, but can be tailored to a more dynamic approach, as the library support dynamic configuration of the IEDs based on APIs - making it highly suitable for this kind of lab.

8.2 Conclusion

This thesis forms a baseline architecture and lab environment for a system where student and researchers can perform various attacks on a IEC 61850 system.

Given that attack scenario 1 is successful and that an attacker gains sufficient access to the control center environment, this leads to an alternative attack scenario, where we can use a simulated HMI/SCADA server in our control center environment which will respond to all incoming requests from engineering workstations or other control functions, spoofing a whole sub-station - given that the attacker can control larger parts of the control infrastructure. This could, in turn, lead to even more interesting results, but require cooperation within the community of researchers and operators of a regional distribution grid.

In scenario 2, we observed the possibility of gaining control of a HMI in a local sub-station. This gives an attacker the possibility to render a sub-station offline from the outside world, requiring manual intervention to get the systems online again. This can be used as a part of a cascading attack, where multiple components of the regional grid can be attacked simultaneous to maximize the damage. The scenarios also show that basic network attacks can be successful, especially when confined in a sub-station. This, of course, require full access to a device in the sub-station network. Given the results from the sub-station, one can assume that identical results could be visible when performed internally in a control station as well.

The simulated attacks show, even if it is in a facilitated environment, that methods that can be seen as mundane, simple and even old, can be used to attack critical infrastructure if combined correctly. There are, however, continuous work from vendors that focus on the IDS/IPS part of CPS to scan and verify control messages in such a system in real time and could, potentially, block some of these steps, rendering them obsolete. This thesis forms the baseline of components in a CPS for regional distribution networks, and a set of highlighted improvements that was not feasible to implement in the timeline of the thesis will be presented in the next section.

8.3 Future Work

For a more reliable and realistic system where testing of attacks and scenarios in critical infrastructure can be performed, there are further actions that could, and should, be implemented. Among things I wish to highlight is the implementation of a holistic setup, where a whole substation with a variety of components all communicate together, and a set of basic prevention mechanisms is implemented to give an attacker a challenge. An additional feature that is desirable, is realistic real-time data that can be played in a loop to simulate real events from a sub-station environment.

The customized programs is based on existing examples with added functionality, and the primary interest in this thesis was to get the communication flowing and change values, and then attack the flow to alter the values in transit. A basic HMI with continuous flow was initiated, but the display of values should be considered to be a static field with the updated values or a graphical visualization, replacing the existing, continuous flow of text that is displayed on the HMI. Additionally, the communication between the EW and the HMI should be based in IEC 61850 traffic, not

"standard" TCP/IP as it is today.

The system should also implement the GOOSE protocol for inter-communication between IED within the same system, coupling the components together and giving a overview of impact on the system as a whole. One should also consider implementing other protocols, like DNP3 and IEC 60870, to see the impact on these protocols as well in the various scenarios. Additionally, the system should implement TLS encryption between the control center and the sub-station to protect the communication even better - where that is applicable, and a API driven approach to the configuration of the IEDs should be considered to make it more dynamic.

Acronyms

ANSI American National Standards Institute

ARP Address Resolution Protocol

CDC Common Data Classes

CPS Cyber-Physcial Systems

DNP istributed Network Protocol

DoS Denial of Service

DDoS Distributed Denial of Service

ENISA European Network and Information Security Agency

HMI Human Machine Interface

ICMP Internet Control Message Protocol

ICS Industrial Control Systems

IEC International Electrotechnical Commission

IED Intelligent Electronic Device

IoT Internet-of-Things

IP Internet Protocol

ISA International Society of Automation

ISO International Organization for Standardization

LAN Local Area Network

LD Logical Device

LN Logical Node

MMS Manufacturing Messaging Specification

MPLS Multiprotocol Label Switching

MTU Master Terminal Unit

NIST National Institute of Standards and Technology

OSI Open System Interconnection

PLC Programmable Logic Controller

QoS Quality of Service

RBAC Role Based Access Control

RTU Remote Terminal Unit

SCL Substation Configuration Language

SCADA Supervisory Control And Data Acquisition

SMV Sampled Measurement Value

SQL Structured Query Language

TCP Transmission Control Protocol

TX Transmit in a network

VPN Virtual Private Network

WAN Wide Area Network

XML Extensible Markup Language

Bibliography

[1] Hahn, A., Ashok, A., Sridhar, S., & Govindarasu, M. June 2013. Cyber-Physical Security Testbeds: Architecture, Application, and Evaluation for Smart Grid. *IEEE Transactions on Smart Grid*, 4(2), 847–855. URL: http://ieeexplore.ieee.org/document/6473865/, doi:10.1109/TSG.2012.2226919. (document), 2.1, 1, 2.1, 2.2, 3.2

[2] The Industrial Control System Cyber Kill Chain. 24. (document), 2, 5.3

[3] Mackiewicz, R. E. November 2006. Overview of IEC 61850 and Benefits. *IEEE*, 8. doi:10.1109/PSCE.2006.296392. (document), 4.2.4, 4.2.4, 3, 4, 4.2.4

[4] Liang, Y. & Campbell, R. H. May 2008. Understanding and Simulating the IEC 61850 Standard. 12. (document), 4.2.4, 5

[5] Kaplan, S. Electric power transmission: Background and policy issues. CRS Report for Congress R40511, Congressional Research Service, January 2011. (document), 6

[6] ENISA Threat Landscape Report 2018. URL: https://www.enisa.europa.eu/publications/enisa-threat-landscape-report-2018. (document), 7, 5.3.1, 6.1.1

[7] Assante, M. J. & Lee, R. M. October 2015. The Industrial Control System Cyber Kill Chain. 24. (document), 10, 11

[8] Case, D. U. 2016. Analysis of the cyber attack on the Ukrainian power grid. *Electricity Information Sharing and Analysis Center (E-ISAC)*. (document)

[9] Frank, M., Leitner, M., & Pahi, T. November 2017. Design Considerations for Cyber Security Testbeds: A Case Study on a Cyber Security Testbed for Education. In *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, 38–46, Orlando, FL. IEEE. URL: http://ieeexplore.ieee.org/document/8328365/, doi:10.1109/DASC-PICom-DataCom-CyberSciTec.2017.23. 2.1, 3.2

[10] Cintuglu, M. H., Mohammed, O. A., Akkaya, K., & Uluagac, A. S. 2017. A Survey on Smart Grid Cyber-Physical System Testbeds. *IEEE Communications Surveys & Tutorials*, 19(1), 446–464. URL: http://ieeexplore.ieee.org/document/7740849/, doi:10.1109/COMST.2016.2627399. 2.1, 2.2, 2.2.1, 3.2

[11] Gunduz, M. Z. & Das, R. March 2018. A comparison of cyber-security oriented testbeds for IoT-based smart grids. In *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, 1–6, Antalya. IEEE. URL: https://ieeexplore.ieee.org/document/8355329/, doi:10.1109/ISDFS.2018.8355329. 2.1, 2.1

[12] Chen, B., Butler-Purry, K. L., Goulart, A., & Kundur, D. September 2014. Implementing a real-time cyber-physical system test bed in RTDS and OPNET. In *2014 North American Power Symposium (NAPS)*, 1–6, Pullman, WA, USA. IEEE. URL: http://ieeexplore.ieee.org/document/6965381/, doi:10.1109/NAPS.2014.6965381. 2.1

[13] Gunathilaka, P., Mashima, D., & Chen, B. 2016. SoftGrid: A Software-based Smart Grid Testbed for Evaluating Substation Cybersecurity Solutions. In *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy - CPS-SPC '16*, 113–124, Vienna, Austria. ACM Press. URL: http://dl.acm.org/citation.cfm?doid=2994487.2994494, doi:10.1145/2994487.2994494. 2.1

[14] Zhou, X., Gou, X., Huang, T., & Yang, S. 2018. Review on Testing of Cyber Physical Systems: Methods and Testbeds. *IEEE Access*, 6, 52179–52194. URL: https://ieeexplore.ieee.org/document/8464069/, doi:10.1109/ACCESS.2018.2869834. 2.1, 3.2

[15] Huang, X., Qin, Z., & Liu, H. November 2018. A Survey on Power Grid Cyber Security: From Component-Wise Vulnerability Assessment to System-Wide Impact Analysis. *IEEE Access*, PP, 1–1. doi:10.1109/ACCESS.2018.2879996. 2.2

[16] Zhu, B., Joseph, A., & Sastry, S. October 2011. A Taxonomy of Cyber Attacks on SCADA Systems. In *2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing*, 380–388, Dalian, China. IEEE. URL: http://ieeexplore.ieee.org/document/6142258/, doi:10.1109/iThings/CPSCom.2011.34. 2.2

[17] Zimmermann, H. April 1980. OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection. *IEEE Transactions on Communications*, 28(4), 425–432. doi:10.1109/TCOM.1980.1094702. 2.2

[18] Kleinmann, A., Amichay, O., Wool, A., Tenenbaum, D., Bar, O., & Lev, L. 2018. Stealthy Deception Attacks Against SCADA Systems. In *Computer Security*, Katsikas, S. K., Cuppens, F., Cuppens, N., Lambrinoudakis, C., Kalloniatis, C., Mylopoulos, J., Antón, A., & Gritzalis, S., eds, volume 10683, 93–109. Springer International Publishing, Cham. URL: http://link.springer.com/10.1007/978-3-319-72817-9_7, doi:10.1007/978-3-319-72817-9_7. 2.2.1

[19] McDonald, M. J., Conrad, G. N., Service, T. C., & Cassidy, R. H. September 2008. Cyber Effects Analysis Using VCSE. 56. 2.2.1

[20] Johannesson, P. & Perjons, E. 2014. A Method Framework for Design Science Research. In *An Introduction to Design Science*, Johannesson, P. & Perjons, E., eds, 75–89. Springer International Publishing, Cham. URL: https://doi.org/10.1007/978-3-319-10632-8_4, doi:10.1007/978-3-319-10632-8_4. 3

[21] Hevner, A. R. 2007. A Three Cycle View of Design Science Research. 19, 7. 3

[22] European Network and Information Security Agency. 2017. *Communication network dependencies for ICS/SCADA systems*. ENISA, Heraklion, OCLC: 1044412009. 3.2, 4.4

[23] Çakil, T., Carlak, H. F., & Özen, September 2015. MODELING of POWER NETWORK SYSTEM of the HIGH VOLTAGE SUBSTATION: a simulation study. *International Journal Of Engineering & Applied Sciences*, 7(3), 39–39. URL: http://akdeniz.dergipark.gov.tr/doi/10.24107/ijeas.251253, doi:10.24107/ijeas.251253. 3.2

[24] Kalbandhe, S. H. & Bhasme, D. N. R. Modeling &Simulation of a 100/22 kV Transmission Substation for Energy Audit. 10. 3.2

[25] Gunathilaka, P., Mashima, D., & Chen, B. 2016. SoftGrid: A Software-based Smart Grid Testbed for Evaluating Substation Cybersecurity Solutions. In *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy - CPS-SPC '16*, 113–124, Vienna, Austria. ACM Press. URL: http://dl.acm.org/citation.cfm?doid=2994487.2994494, doi:10.1145/2994487.2994494. 3.2

[26] Knapp, E. & Langill, J. T. 2015. *Industrial Network Security*. Elsevier. URL: https://linkinghub.elsevier.com/retrieve/pii/C20130068363, doi:10.1016/C2013-0-06836-3. 3.2, 4.1.2, 4.1.4

[27] Zillgith, M. libIEC61850 / lib60870-5 | open source libraries for IEC 61850 and IEC 60870-5-104. URL: http://libiec61850.com/libiec61850/. 3.2, 6.1.2, 6.1.2, 6.2.1, 6.3

[28] Debian – The Universal Operating System. URL: https://www.debian.org/index.en.html. 3.3

[29] Download Raspberry Pi OS for Raspberry Pi. Library Catalog: www.raspberrypi.org. URL: https://www.raspberrypi.org/downloads/raspbian/. 3.3

[30] Threat Taxonomy. URL: https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends/enisa-threat-landscape/threat-taxonomy. 3.4

[31] East, S., Butts, J., Papa, M., & Shenoi, S. 2009. A Taxonomy of Attacks on the DNP3 Protocol. In *Critical Infrastructure Protection III*, Palmer, C. & Shenoi, S., eds, volume 311, 67–81. Springer Berlin Heidelberg, Berlin, Heidelberg. Series Title: IFIP Advances in Information and Communication Technology. URL: http://link.springer.com/10.1007/978-3-642-04798-5_5, doi:10.1007/978-3-642-04798-5_5. 3.4, 4.2.6

[32] Kang, B., Maynard, P., McLaughlin, K., Sezer, S., Andrén, F., Seitl, C., Kupzog, F., & Strasser, T. September 2015. Investigating cyber-physical attacks against IEC 61850 photovoltaic inverter installations. In *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*, 1–8. ISSN: 1946-0759. doi:10.1109/ETFA.2015.7301457. 3.4, 5.1.1

[33] Maynard, P., Mclaughlin, K., & Haberler, B. September 2014. Towards Understanding Man-In-The-Middle Attacks on IEC 60870-5-104 SCADA Networks. doi:10.14236/ewic/ics-csr2014.5. 3.4

[34] Yan, J., Tang, Y., Zhu, Y., He, H., & Sun, Y. December 2015. Smart Grid Vulnerability under Cascade-Based Sequential Line-Switching Attacks. In *2015 IEEE Global Communications Conference (GLOBECOM)*, 1–7. doi:10.1109/GLOCOM.2015.7417255. 3.4, 5, 5.2.1

[35] Bolton, W. 2009. *Programmable logic controllers*. Newnes, Amsterdam ; Boston, 5th ed edition. 4.1.1

[36] Colbert, E. J. M. & Kott, A., eds. 2016. *Cyber-security of SCADA and Other Industrial Control Systems*, volume 66 of *Advances in Information Security*. Springer International Publishing, Cham. URL: http://link.springer.com/10.1007/978-3-319-32125-7, doi:10.1007/978-3-319-32125-7. 4.1.5, 4.1.6

[37] Thomas, G. 2008. Introduction to the Modbus Protocol. 4. 4.2.5

[38] Physical Security of the U.S. Power Grid: High-Voltage Transformer Substations. Library Catalog: www.everycrsreport.com. URL: https://www.everycrsreport.com/reports/R43604.html. 4.3.2

[39] Taylor, J. M. & Sharif, H. R. December 2017. Enhancing integrity of modbus TCP through covert channels. In *2017 11th International Conference on Signal Processing and Communication Systems (ICSPCS)*, 1–6. doi:10.1109/ICSPCS.2017.8270454. 5.1.1

[40] Radoglou-Grammatikis, P., Sarigiannidis, P., Giannoulakis, I., Kafetzakis, E., & Panaousis, E. July 2019. Attacking IEC-60870-5-104 SCADA Systems. In *2019 IEEE World Congress on Services (SERVICES)*, 41–46, Milan, Italy. IEEE. URL: https://ieeexplore.ieee.org/document/8817093/, doi:10.1109/SERVICES.2019.00022. 5.1.2

[41] Rashid, M., Yussof, S., Yusoff, Y., & Ismail, R. November 2014. A review of security attacks on IEC61850 substation automation system network. 5–10. doi:10.1109/ICIMU.2014.7066594. 5.1.2

[42] Sybille, G. OLTC Phase Shifting Transformer (Phasor Model) - MATLAB & Simulink - MathWorks Nordic. URL: https://se.mathworks.com/help/physmod/sps/examples/oltc-phase-shifting-transformer-phasor-model.html. 5.2.1, 5.2.2, 6.1.3, 6.3

[43] WaterISAC. October 2016. 10 Basic Cybersecurity Measures. URL: https://www.waterisac.org/sites/default/files/public/10_Basic_Cybersecurity_Measures-WaterISAC_Oct20165B25D.pdf. 5.3.1

[44] Johansen, T. NTNU-IED GitLab Repository. Library Catalog: gitlab.com. URL: https://gitlab.com/thomas162/ntnu-ied. 6, 6.1.3, 6.2.1, 6.2.2, 6.3.2

[45] Zillgith, M. Building the library | libIEC61850 / lib60870-5. URL: https://libiec61850.com/libiec61850/documentation/building-the-library/. 6.1.2

[46] Casoria, S. VSC-Based HVDC Transmission System (Detailed Model) - MATLAB & Simulink - MathWorks Nordic. URL: https://se.mathworks.com/help/releases/R2017b/physmod/sps/examples/vsc-based-hvdc-transmission-system-detailed-model.html. 6.1.3, 6.3

[47] September 2018. TCP Server-Client implementation in C. Library Catalog: www.geeksforgeeks.org Section: C. URL: https://www.geeksforgeeks.org/tcp-server-client-implementation-in-c/. 6.2.1, 6.3

[48] Matlab. Stream Input/Output - MATLAB & Simulink - MathWorks Nordic. URL: https://se.mathworks.com/help/sldrt/ug/stream-input-output.html. 6.3

[49] Burp Proxy: invisible proxying. Library Catalog: portswigger.net. URL: https://portswigger.net/burp/documentation/desktop/tools/proxy/options/invisible. 6.3.1

[50] jrmdev. May 2020. jrmdev/mitm_relay. original-date: 2017-01-04T04:15:49Z. URL: https://github.com/jrmdev/mitm_relay. 6.3.1

[51] 0daysecurity.com. Hping3 Examples - Firewall testing | 0DAYsecurity.com. Library Catalog: 0daysecurity.com. URL: http://0daysecurity.com/articles/hping3_examples.html. 7

A Appendix - Simulink models

Overview of Simulink configuration, components and IO parameters. Detailed instructions of components used and changes to default parameters in Simulink. All changes related to IO between Simulink and IED details in here.

A.1 Simulink - Phase shift

With Matlab and Simulink installed, the initial model can be loaded by executing the following command in Matlab command line:

`power_PSTdeltahex`

In this model, the following blocks were added:

- Stream Output
- Stream Input (in the sub-model under "Tap Control")
- 3x demux with 2 outgoing ports
- 1 mux with 8 incoming ports

The data coming from the "Signal Processing" box was tapped into and the signals sent to a demuxer for then to be connected to a mux with 8 ports. This mux is used to combine all the incoming signals and connected to a "Stream Output" block which sends the signals in a binary stream to the IED controlling the model. The Stream Output block is configured to act as a client, sending data to the IP address of the IED on a predefined TCP/IP port and can be observed in figure 28.

The "Stream Input" box is configured to act as a server, listening on port tcp/36880 and receive two int8 parameters which is connected to a demuxer to split the incoming signal to the "step up" and "step down" blocks of the OLTC function as can be seen in figure 29.

A.2 Simulink - Voltage conversion

With Matlab and Simulink installed, the initial model can be loaded by executing the following command in Matlab command line:

`power_hvdc_vsc`

In this model, the following blocks were added:

- Stream Output, in the sub-model under "Data Acquisition Station 2"
- Stream Input, in the sub-model under "VSC Controller Station 2"
- 1 mux with 9 incoming ports

The signals coming from parts of the "Data acquisition station 2" box was tapped into and the signals connected to a mux with 9 ports. This mux is used to combine all the incoming signals

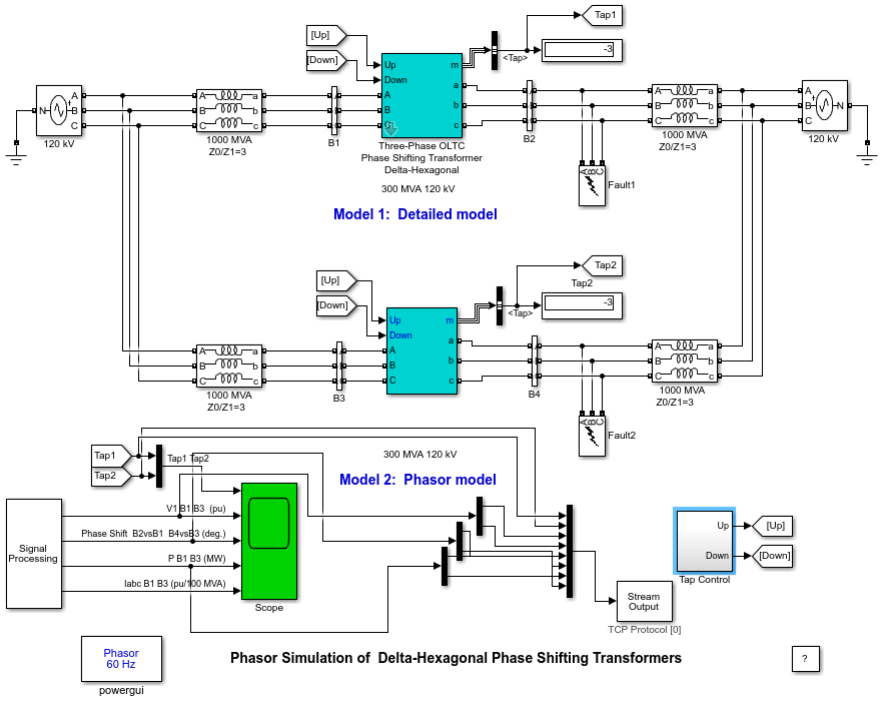


Figure 28: OLTC Phase Shifting Transformer (Phasor Model)

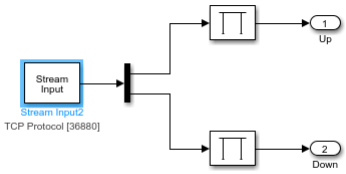


Figure 29: OLTC Phase Shifting Transformer - Stream input model

and connected to a "Stream Output" block which sends the signals in a binary stream to the IED controlling the model. The Stream Output block is configured to act as a client, sending data to the IP address of the IED on a predefined TCP/IP port and can be observed in figure 31.

The "Stream Input" box is configured to act as a server, listening on port tcp/36881 and receive one int8 parameters which is connected to a output box where a constant is stored. This constant can then be used to, programmatic, trigger the DC balance part of the "Discrete VSC Controller" and trigger a "DC Voltage Balance on" and "DC Voltage Balance off" feature in the model, as can be seen in figure 30.

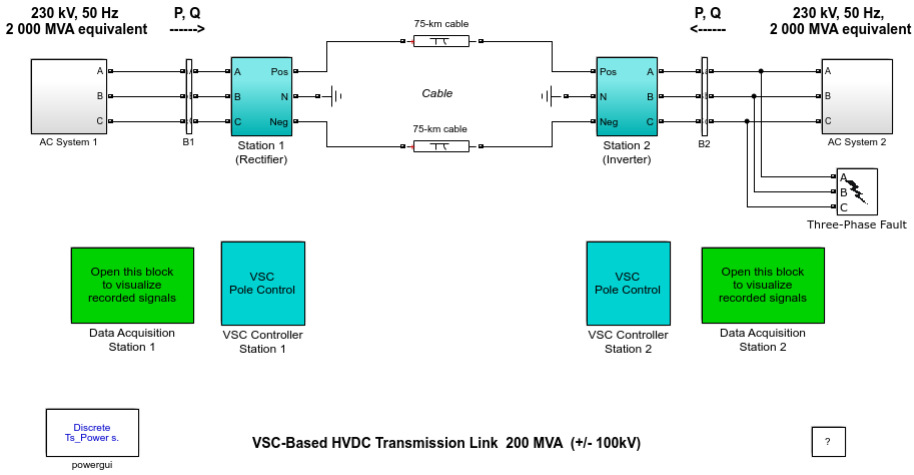


Figure 30: VSC-Based HVDC Transmission System (Detailed Model)

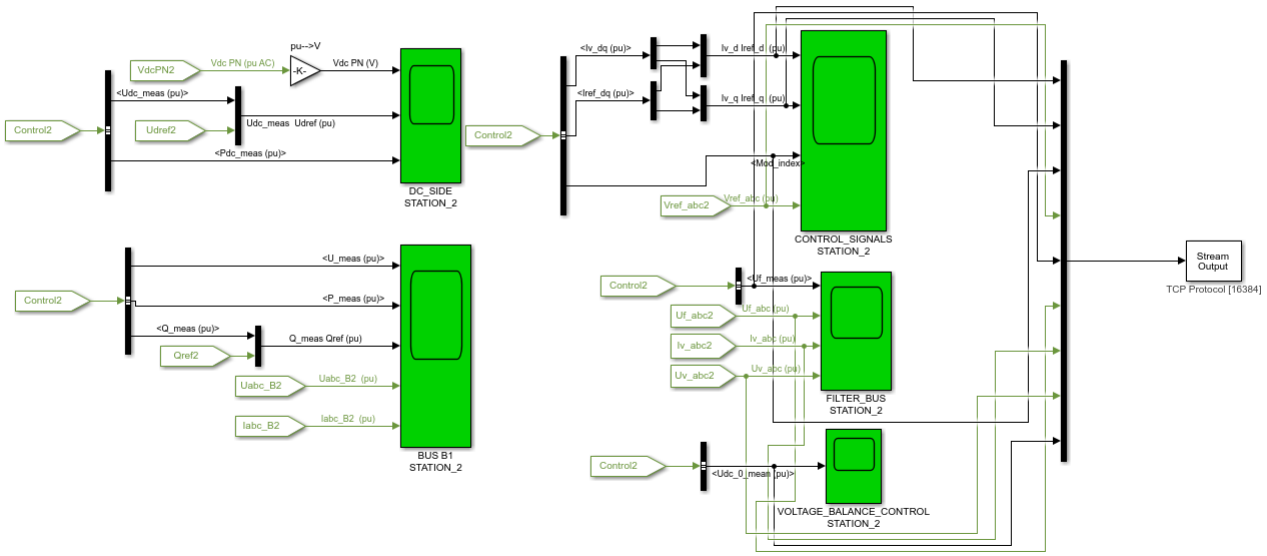
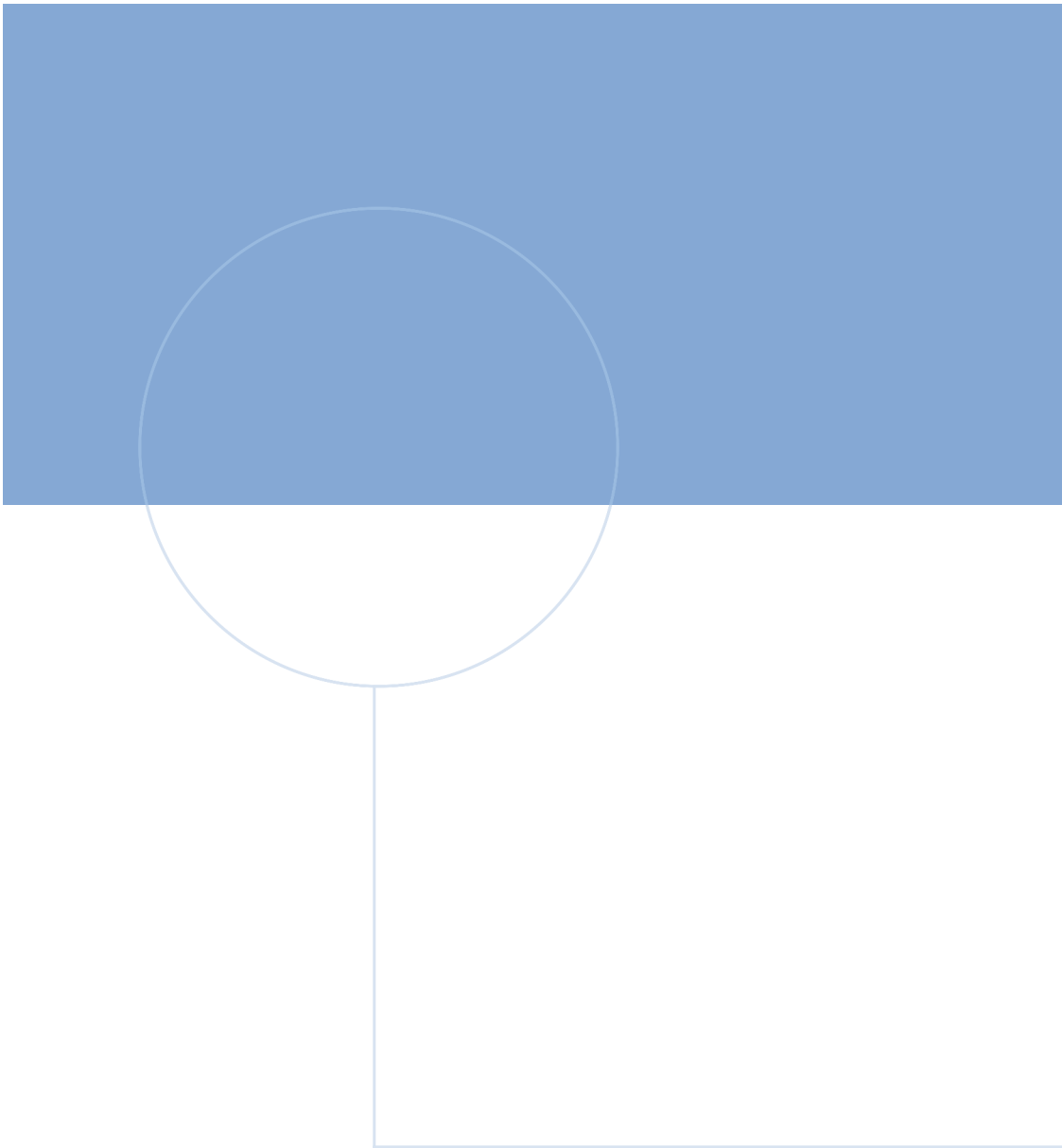


Figure 31: VSC-Based HVDC Transmission System (Detailed Model)



NTNU

Kunnskap for en bedre verden