Master's thesis

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

Maja Simons Markusson

Consequence-based Detection of Cyberattacks in Operational Technology

Master's thesis in Cybernetics and Robotics
Supervisor: Mary Ann Lundteigen
Co-supervisor: John Petter Indrøy
May 2023

NTNU
Norwegian University of
Science and Technology

Maja Simons Markusson

Consequence-based Detection of Cyberattacks in Operational Technology

Master's thesis in Cybernetics and Robotics
Supervisor: Mary Ann Lundteigen
Co-supervisor: John Petter Indrøy
May 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

NTNU
Kunnskap for en bedre verden

Department of Engineering Cybernetics

TTK4900 - Masteroppgave, teknisk kybernetikk

Consequence-based Detection of Cyberattacks in Operational Technology

Author:
Maja Simons Markusson

June, 2023

Abstract

With the increasing interconnectedness of industrial systems, organizations face a growing risk of cyber-related incidents. This poses a significant challenge to ensuring the safety and security of critical infrastructure. A well-known mitigating strategy is monitoring to detect ongoing cyber-attacks. Monitoring is performed using intrusion detection system.

This study aims to enhance intrusion detection in industrial control systems by incorporating techniques from specification-based detection. Identifying worst-case scenarios and implementing detection based on specific feature specifications can increase the ability to detect attacks and enhance situational awareness. This approach allows for cost-effective detection by focusing on critical system components and process variables rather than implementing a complete specification-based solution.

The research methodology involves utilizing the CCE method to identify critical subsystems and components within an organization. These components are further analyzed to determine essential parameters that indicate malicious activity based on an adversary' expected approach. The chosen parameters are modelled using system knowledge and specifications, and the specific aspects of the systems behavioural model are integrated into the detection routine.

A limitation of this method is that it only provides partial coverage, focusing on specific components and process variables. However, by prioritizing critical elements and high-consequence incident evasion, this approach offers a cost-effective solution to enhance safety and security with limited resources.

The research contributes to the existing body of knowledge by proposing a methodology to improve detection in critical components, supplementing exsisting intrusion detection system. The implementation and testing of the proposed approach using a representative dataset demonstrate its effectiveness in detecting potential attacks. The study also illustrates how monitoring can be used to ensure system integrity confidence when utilizing TCP/IP-based communication in the lower levels of industrial systems.

Keywords: Cybersecurity, Industrial Control Systems, Intrusion Detection Systems, Specification-based Detection, Operational Technology, Signature-based Detection, Consequence-based Cyber-informed Engineering.

Sammendrag

Med økende sammenkobling av industrielle systemer står organisasjoner overfor større risiko knyttet til cyberangrep. Når det gjelder kritisk infrastruktur utgjør dette en betydelig utfordring. Et vanlig verktøy for å oppdage cyberangrep er innbruddsdeteksjonsystemer.

Denne studien har som mål å forbedre signaturbaserte deteksjonssystemer ved å inkorporere teknikker fra spesifikasjonsbasert deteksjon for kritiske system. Ved å identifisere hendelser som kan føre til stor-ulykker og implementere deteksjon basert på observerbare parametre knyttet til disse hendelsene, kan en øke evnen til å oppdage angrep og forbedre situasjonsforståelse. Dette er en kostnadseffektiv måte å forbedre deteksjonsevne uten å måtte modellere og implementere et helt system til bruk i spesifikasjonsbasert deteksjon.

Studien bruker CCE-metoden for å identifisere kritiske subsystemer og komponenter innenfor en organisasjon. Disse komponentene analyseres videre for å bestemme sentrale parametere som indikerer fiendtlig aktivitet. De valgte parameterne modelleres deretter ved hjelp av systemkunnskap og spesifikasjoner. Denne modellen brukes så i deteksjon for å oppdage både fiendtlig aktivitet og systemfeil. En begrensning med fremgangsmåten er at den kun gir delvis dekning. Dette er fordi den kun ser på ett subsystem og ett scenario. Imidlertid, ved å prioritere kritiske elementer og hendelser med potensielt store konsekvenser, kan den fremdeles være en kostnadseffektiv måte å forbedre sikkerheten med et lavere ressursforbruk.

Denne studien bidrar til feltet innenfor cybersikkerhet i industriell teknologi ved å foreslå en metode for å forbedre deteksjonen inspirert av signaturbasert deteksjon. Testing av den foreslåtte løsningen gjøres ved bruk av et representativt datasett. Resultatene viser at løsningen effektivt kan oppdage potensielle angrep. Studien illustrerer også hvordan overvåking kan brukes for å opprettholde systemtillit når TCP/IP-basert kommunikasjon benyttes på de laveste nivåene av industrielle systemer.

Nøkkelord: Cybersikkerhet, Innbruddsdeteksjon, Signaturbasert Innbruddsdeteksjon, Spesifikasjonsbasert Innbruddsdeteksjon, Konsekvensbasert Evaluering, Industrielle Kontrollsystemer, Operasjonell Teknologi.

Preface

This research report was written during the spring of 2023 as a part of my Master's thesis at the Department of Engineering Cybernetics at NTNU. The topic is intrusion detection for operational technology, as a continuation of my project thesis the preceding fall.

During my project thesis, I was invited by my supervisor to attend several meetings and seminars. Here, I became aware of the massive changes happening within the field of cybersecurity within operational technology and what is yet to be researched. I am intrigued by being a part of a field undergoing rapid development and massive changes. The research topic, in specific, is inspired by one of the main conclusions from the first literature study I performed. Security solutions currently applied in operational technology need to become more specific to utilize the full potential of the methods. With this study, I hope to expand the horizons of intrusion detection in critical infrastructure.

This report is written for readers with a technological background. It assumes some knowledge of automation, computer science, and embedded systems. Introductory cybersecurity knowledge is advantageous, but all terms relevant to the discussions and conclusions are explained.

I want to express my gratitude to those who have supervised me during this project. Prof. Mary Ann Lundteigen, for helping and encouraging me through both the Master's and project thesis. You have always been able to help me see new possibilities when I have been feeling stuck. John Petter Inderøy, my primary supervisor at Equinor, for your creativity and guidance through all aspects of completing this thesis. Øyvind Varne and Einar Færaas, for your expertise and availability, whenever I've had a question. I've been incredibly lucky to have that many talented people supporting me.

Trondheim, 31-5-2023

Maja Simons Markusson

Table of Contents

List of Figures

List of Tables

Abbreviations

A&E Alarms & Events.

APS Abandon Platform Shutdown.

ATT&CK Adversarial Tactics, Techniques, and Common Knowledge.

BD Blow-Down.

BDV Blow-down Valve.

CAP Critical Action Panel.

CCE Consequence-driven Cyber-informed Engineering.

CCTV Closed-circuit Television.

CIA Confidentiality, Integrity and Availability.

CPU Central Processing Unit.

DA Data Access.

DCS Distributed Control Systems.

DMZ Demilitarized Zone.

DoS Denial of Service.

ESD Emergency Shutdown.

ESV Emergency Shutdown Valve.

EWS Engineering Work Station.

GL Guideline.

HCE High-consequence Event.

HDA Historical Data Access.

HMI Human Machine Interface.

HTTP Hypertext Transfer Protocol.

HTTPS Hypertext Transfer Protocol Secure.

I/O Input/Output.

IACS Industrial Automated Control System.

ICS Industrial Control System.

IDS Intrusion Detection System.

IEC International Electrotechnical Commission.

IIoT Industrial Internet of Things.

INL Idaho National Laboratory.

IT Information Technology.

KO Knock-Out.

LAHH Level Alarm High High.

MitM Man in the Middle.

NaN Not a Number.

NIST National Institute of Standards and Technology.

OPC-UA Open Platform Communications-Unified Architecture.

OT Operational Technology.

pcap Packet Capture.

pcapng Packet Capture Next Generation.

PLC Programmable Logic Controllers.

PRV Pressure Relief Valve.

PSA Petroleum Safety Authority.

PSD Process Shutdown.

PSV Pressure Safety Valve.

RTU Remote Terminal Units.

SCADA Supervisory Control and Data Acquisition.

SIEM Distributed Intrusion Detection System.

SIL Safety Integrity Level.

SIS Safety Instrumented System.

SNMP Simple Network Management Protocol.

SSH Secure Socket Shell.

TCP/IP Transmission Control Protocol/Internet Protocol.

UPS Uninterruptible Power Supply.

1 Introduction

1.1 Background and Motivation

Moving into a more connected industrial era, organizations face increasing cyber-related risks. Now, major incidents can be caused intentionally by malicious actors. Legacy equipment, complex systems and topology and lack of system patches increases these risks additionally. A characteristic of operational technology systems is that they can affect the health, environment and safety of their surroundings. Safety is defined as *"freedom from unacceptable risk"*[1]. So preventing incidents compromising the system safety is a necessity.

Detecting malicious activity and system failure is essential to avoid safety incidents. Malicious activity detection is often done using an intrusion detection system. Signature-based detection tools are most commonly used because they provide the most straightforward set-up and produce the least false alarms. Signature-based detection for industrial control systems has been criticized[2] because of their inability to detect new attacks, indicating they currently do not provide sufficient coverage for operational technology. They also comment that the signatures rarely consider the system specifications or state.

Altering system states according to the specifications was one of the main features in the Stuxnet attack[3]. By targeting the programmable logic controllers that controlled the centrifuges, Stuxnet aimed to manipulate their speed and functionality, leading to mechanical failures and damaging the centrifuges. Therefore one of the research questions aims to answer if signature-based detection tools can be supplemented so that the total security solution provides sufficient coverage without implementing the entire solution.

Safety-related subsystems, also called safety instrumented systems, are considered a crucial part of the overall system. Incidents in these subsystems can aggravate the effect of other attacks or system malfunctions. The Triton malware is an example of malware attacking such safety-instrumented systems in Schneider's Triconex device[4]. One of the key techniques in Triton is the masking of sensor readings between the sensors and the human-machine interface, causing the system to appear in a normal state while under attack.

Modelling an entire organization based on specifications would be a specifications-based approach. Specification-based detection is highly resource demanding with respect to modelling and implementation. By only implementing detection for some parts of the system, the cost and complexity will be lower than for a complete setup. Partial modelling raises the question of which part of the system should be prioritized. Using cyber assessments frameworks, critical subsystems and components can be identified in an organization. One type of assessment method is consequence-based assessments, which focuses on the consequences of cyberattacks. The literature is sparse compared to the one regarding more conventional risk assessment methods. A common denominator for these methods is that they do not use likelihood as an evaluation criterion.

The project has been performed in cooperation with Equinor, a Norwegian energy company. At the start of the project, they introduced a case study about a flare system, which could be assessed to improve detection. The flare system is a safety instrumented system, so it is important to maintain system integrity confidence. They have have provided the background for the flare system case study and given continuous technical guidance about automation, industrial network topology and cybersecurity. Cooperating with a

company have given insight into real-life experiences and ensures that the research topic and solutions has been relevant.

The study takes the opportunity to perform a consequence-based case study assessment. The assessment helps to point out critical system points for potential high consequence events. Limiting the number of process variables to monitor can improve situational awareness during incidents by effectively identifying which systems are affected. The increased situational awareness can limit precautionary shut-downs of parts or the entire facility.

1.2 Related work

Intrusion Detection Systems for Operational Technology[5] is a literature study from 2022 researching intrusion detection systems (IDS) for operational technology and industrial control systems. The study identifies open source systems and software performing intrusion detection, and identifies strength and weaknesses when compared to the characteristics of operational technology. The study concludes that many systems and ideas from the IT domain is relevant, but will have to be customized according to operational technology characteristics to provide sufficient coverage. The study does, however, not perform any tests on actual devices or systems.

Idaho National Laboratory (INL) has developed the Consequence-Driven Cyber-informed Engineering (CCE) method[6] used in this study and published two case studies performing such analyses[7][8]. Both case studies analyze larger, more well-defined cases than what is handled in this study. They also perform the entire mitigations stage rather than focusing on the detection. In comparison, by focusing on only selecting features for detection, this study extracts features and designs a specific solution for intrusion detection based on system specifications.

Shin and Hyung[9] discuss the usage of consequence-based assessment methods for industrial systems based on considerations from IAEA's INFCIRC 225_Rev5 and NRC NUREG/CR6847. They suggest a new approach based on the frameworks mentioned, focusing on the different system levels rather than only the level of compromise. The research is not based on a specific case study and does not suggest mitigations or protections.

Mitchell and Chen[2] conducted an extensive survey of intrusion detection techniques for cyber-physical systems in 2014. They evaluated 30 different solutions, of which nine included systemspecifications, and only two relied exclusively on signature-based detection. They justify the low number of signature-based techniques by arguing that the method is insufficient for cyber-physical systems because of its inability to detect zero-day attacks. Zero-day attacks exploit a known vulnerability that has not yet been patched. It is also stated that attackers will become more sophisticated when targeting critical systems and not rely on known vulnerabilities.

Including process parameters in the traditional signature-based detection is a form of multi-trust. Multi-trust is a concept where external variables or opinions are used to decide whether the system is attacked. Gyamfi and Jurcut apply one example of this technique in their detection system for IIoT[10]. This detection system uses parameters such as network traffic, CPU usage, and energy consumption on the device.

1.3 Project Objective and Tasks

The main objective is to research the possibility of signature-based detection of cyber-attacks in critical components using signature-based techniques. The hypothesis is that consequence-based assessments can aid in improving detection tools for critical systems in industrial control systems, by identifying useful process parameters and data points. The main objective is realised through two main goals;

- Identify critical data points and process parameters in a flare system using a CCE-inspired assessment method.
- Improve detection in operational technology using techniques from specification-based detection and process parameters identified in the conducted assessment.

Data points are points in the network topology that can be used to extract information. Process parameters are parameters describing the state of a specific process, e.g. temperature. The main goals can be further divided into sub-tasks, providing a more detailed approach to realising the main objective. To illustrate the approach, a flare system use case has been included. The subtasks has been performed on the case study.

1. Assess potential high-consequence events in a flare system using the CCE method.
2. Extract possible data points or parameters from the consequence assessment.
3. Choose relevant detection parameters. Detection parameters are variables or states used to decide if the system is behaving abnormally.
4. Model these parameters using the system's expected behaviour.
5. Implement detection routines.
6. Test implementation against a representative dataset.

1.4 Research Approach

The research approach has been diverse and includes; literature research, assessment processes, software design, implementation and testing. The project started off with gathering and summarizing the necessary literature before assessing the case study using the first three steps of the CCE method. During the CCE assessment the study; identified potential cyber events in a flare system, chose a high-consequence event to focus on, collected all available information on the system and developed an attack scenario. The results were used to implement functional code. The information collected was used to create the model of the expected system behaviour, while the attack scenario was used to create detection routines.A test environment was used to produce a test dataset. The implemented solution was tested on this dataset.

The literature study referred to in Section 1.2 forms the theoretical baseline of intrusion detection systems in operational technology[5]. To perform the sub-tasks of this study, some additional literature needed to be included. The topics added were specification-based detection and flare systems. In order to find relevant literature, searches were performed on Science Direct, Engineering Village and Google Scholar. The search terms have mainly been;

- "safety systems intrusion detection"
- "signature-based ids"
- "safety security cyber-physical systems"
- "specification signature ids cps"
- "specification ids cps"
- "flare systems"
- "flare system vulnerabilities"

As some of the motivation behind this study is to implement a solution that can be used in existing industrial applications, guidance and discussion with Equinor have been crucial. Some time has been spent attending meetings with external actors to gain additional opinions and further information on specific topics.

- Meeting with IFE 27.1.2023
- Meeting with Livinius Nweke from NTNU 31.1.2023
- Meeting with Aker BP 21.3.2023
- Presentation for CBM project 4.5.2023
- Attendance to CDS forum 25.5.2023

The CCE method requires a large amount of information and assessment during the different stages. Much of the project has been used to document and research relevant topics before proceeding to the development. Early in the process, it was decided that tests and data should be synthetically produced to avoid disclosing sensitive information and ensure the tests were performed. This work was done in parallel with the latest stages of the CCE method. The test scenarios and code design were finished before the suggested solution was designed and implemented, but the datasets used for testing were collected post-implementation.

1.5 Assumptions and Limitations

In order to investigate and answer the research topics within the scope of the study, some limitations have been made. The list below describes the limitations and the reasoning behind them:

- Risk assessing and modelling parameters of an entire facility in utility or oil & gas industry would consume much more time than the one in disposition. Therefore a case system was provided by Equinor. The system reflects upon reality but is not a reproduction of any real facility.
- The assessment of the system used to perform feature selection is conducted less thoroughly than the CCE method. However, it is strongly influenced by it and inherits its main components. It is limited because a CCE assessment of an entire organization is estimated to take around one year and requires knowledge from several disciplines.

- As the case study is not a real installation, it cannot provide real data to test the implementation. Which means that any data used for testing needed to be synthesised.
- The suggested monitoring solution is a proof of concept, the quality and robustness of the implemented solution are not considered benchmarks for quality when answering the research topics.

1.6 Structure of Report

As the project includes a wide range of topics, the section regarding theory is quite extensive. Therefore, this part has been divided into three different sections - operational technology (Section 2.1), cybersecurity (Section 3), and the consequence-driven cyber-informed engineering (CCE) method (Section 4). The reader is invited to read all parts or skip the ones where they have sufficient background knowledge.

Following the theory sections, a case study is performed in Section 5 to illustrate the usage of the CCE method. The use case is introduced, and the three first steps of the method are applied. The resulting attack scenarios are used to perform a detection parameter selection.

In Section 6, the chosen detection parameters are analysed to design and implement suitable monitoring solutions for the flare system. The section also introduces a test framework to synthetically produce test data is described. Numerical results are presented in Section 7. The method and quality of the study are discussed in Section 8. Summing up the report, conclusions and future work is presented in Section 9.

2 Operational Technology

This section briefly introduces operational technology (also referred to as industrial control systems) and topics within the field relevant to the study. Breaking it down, Section 2.1 introduces some general concepts on operational technology. Section 2.2 presents the communication technologies used in the case study; OPC-UA and Ethernet APL. Operational technology or cyber-physical systems are often subject to regulations, so some of the most relevant industrial standards are described in Section 2.3. To form a baseline for the case study, Section 2.4 describes flare systems and their potential hazards.

2.1 Operational Technology Principles

Operational Technology is a collective term for programmable systems that can affect the physical environment around them[11]. Operational technology is a sub-group of cyber-physical systems (CPS) but is more specifically used for industrial purposes like energy production or automation. Industrial control systems (ICS) allow humans to control the operational technology to obtain a desired effect or state. OT consists of a variety of components and sub-systems. These components are usually, but not limited to:

- Supervisory Control and Data Acquisition systems (SCADA)
- Distributed Control Systems (DCS)
- Engineering Work Stations (EWS)
- Human Machine Interfaces (HMI)
- Remote Terminal Units (RTU)
- Programmable Logic Controllers (PLC)
- Sensors
- Actuators

An OT environment is composed of a variety of devices. This is called heterogeneity. It increases the possible attack surface and the number of devices and systems that can potentially be exploited[5]. According to the Purdue Enterprise Reference Architecture[12] (Purdue model for short), an OT environment comprises several levels. The purpose of the Purdue model is to provide segmentation and separation between the levels in a hierarchical manner. In such way, air-gapping is obtained. Air-gapping is one of the most common security measures in OT.

Levels 4 and 5 contain normal the enterprise network with normal IT traffic. These are separated from the OT network by a demilitarized zone (DMZ) at level 3.5. The demilitarized zone is used to communicate safely between the IT and OT networks. Level 3 is the administration of the ICS, containing systems and devices like EWS (can also be placed at level 2) and historians. Devices at level 2, like HMI and SCADA, are used to interact with the ICS through PLCs at level 1. Finally, the devices interacting with the physical environment, sensors and actuators are placed at level 0. The devices at the lower levels are generally considered to have limited computing capacity, and a low tolerance for delays[13].

The devices in an OT environment use different communication protocols to interact. The protocols used in industrial environments differ from the ones used in an IT network. One of the most considerable differences is that industrial communication protocols traditionally have not been using encryption or other security measures. Devices like PLC or SCADA can also come with proprietary protocols, one example being Siemens S7 protocol [14][15].

The lack of encryption and security measures originates from characteristics of embedded and industrial systems and the fact that these systems have been air-gapped from the internet[5]. Controllability and observability are essential for ICS systems to maintain stability in autonomous systems and control loops. This is a key requirement for ensuring the reliability and effectiveness. Heavy-computation encryption techniques may endanger the real-time properties. Because of these characteristics, availability is the most important priority in an OT system, followed by integrity and confidentiality. Conversely, in IT networks, the main priorities are typically confidentiality, integrity, and availability (CIA), in that order. However, industrial protocols with a higher focus on security are starting to appear, like OPC-UA and Ethernet-based protocols.

Other characteristics of operational technology that need to be taken into consideration when working on the subject are the high requirements of determinism and real-timeliness. Approving a system or device for use is a tedious and resource-demanding process. As a result, many systems contain legacy equipment. Legacy equipment is devices or systems that are outdated, unsupported or no longer in production. These can become vulnerabilities as they might not be patchable, even if system protocols allow them. An unpatched system results in a high number of possible zero-day vulnerabilities. A zero-day vulnerability is a vulnerability which is known but not have been patched in the system.

2.2 Communication in OT

This section introduces the baseline theory about OPC-UA and Ethernet-APL, which are used in the flare system case study in Section 5. OPC-UA is a framework widely used for communications and control in industrial applications. Ethernet-APL is a new industrial Ethernet standard specifically designed for process automation applications. Its goal is to provide TCP/IP-based communication on the lower levels of the Purdue model.

2.2.1 OPC-UA

Open Platform Communications Unified Architecture (OPC-UA) is a machine-to-machine communication platform for industrial automation[16]. By platform, it is meant that it provides both a communication protocol and a framework for information architecture. It provides a standardized and secure way for different devices and systems to exchange data, regardless of the manufacturer, operating system, or programming language used. Communication is done using binary TCP/IP. OPC-UA combines the existing OPC primary components; OPC DA, OPC A&E, and OPC HDA. The goal is to provide a more comprehensive and integrated solution than what could be done with the earlier versions of the platform. Listed below is a short description of the three main components:

- OPC DA (Data Access): This component of the OPC specification provides a standard way for data to be accessed and exchanged between OPC servers (which host the data) and OPC clients (which request and receive the data). OPC DA is used

primarily for real-time data, such as sensor readings, process variables, and machine status.

- OPC A&E (Alarms & Events): This component is used to manage alarms and events generated by industrial processes or equipment. OPC A&E allows OPC clients to monitor and respond to these events, such as machine faults, process deviations, and other abnormal conditions.
- OPC HDA (Historical Data Access): This component enables access to historical data from industrial processes, such as production rates, quality metrics, and equipment performance. OPC HDA allows OPC clients to retrieve and analyze this data for trend analysis, process optimization, and other purposes.

One of the main drivers for implementing the platform is the improved security. OPC-UA provides communication with;

- Session encryption
- Message signing
- Sequenced packets
- Authentication
- User control
- Auditing

The communication flow can be configured using different communication patterns. Communication configuration will affect the type and amount of messages sent between a client and a server and how vulnerable the architecture will be (see Section 3.3.2). An OPC-UA client/server relation can be configured to communicate using the following patterns;

- Client/Server: This is the most common communication pattern in OPC UA. The client sends a request to the server, and the server responds with a response message. This pattern performs operations like reading, writing, and browsing the server's address space.
- Publish/Subscribe: The server publishes data to clients that have subscribed to it. The data is sent in notification messages, and the clients receive it without explicitly requesting it. This pattern is used for real-time data streaming and monitoring.
- Method Call: The client requests the server to execute a specific method. The server responds with a response message indicating the success or failure of the method execution.
- Event Notification: The server sends notifications to subscribed clients when certain events occur, such as a change in a variable value or an alarm condition.

2.2.2 Ethernet-APL

Ethernet Advanced Physical Layer (Ethernet-APL) is a relatively new Ethernet-based technology specifically designed for industrial applications. The communication protocol facilitates IP-based communication down to level 0 devices. Devices at level 0 are currently likely to communicate using techniques like 4-20mA signals. The protocol is designed to have the ability to be installed both in existing and new facilities. *"Ethernet-APL uses a two-wire Ethernet technology which provides both communication and power to devices. Defined strictly as a physical layer, Ethernet-APL supports any Ethernet-based protocol, including those with real-time capabilities. Therefore, FieldComm Group, ODVA Inc, OPC Foundation and PROFIBUS and PROFINET International (PI) cooperate in the development of the APL technology"*[17].

Implementing Ethernet-APL in the lower levels of industrial applications would have pros and cons. The most apparent advantages are increased speed, bandwidth and integration. Using Ethernet, one might achieve data rates up to 10 Mbps full duplex over a 200 m distance. The signal can be routed for longer distances using switches. Using IP-based standards also provides increased data portability, requiring less alteration of data to obtain readable communication. It is also argued that using Ethernet-APL will become advantageous when the Purdue Model is switched in favour of more interconnected topologies in Industry 4.0.

The downside is that implementing a new communication protocol of this sort requires a change of physical infrastructure. Altering the infrastructure means cost related to investment of hardware (like power and field switches) and cabling. Ethernet is also a more advanced protocol than 4-20 mA, demanding the operators installing and maintaining it to have sufficient competence. Introducing an Ethernet-based protocol to the down-most levels also introduces security concerns. By using a protocol with larger capabilities, the attack surface is increased. Compared to the previously restricted 4-20 mA, the Ethernet protocol enables more sophisticated and composite attacks.

2.3 Industrial standards

All companies and organizations need to comply with various regulations to be allowed practice within geographical areas or industries. Regulations and guidelines are described by standards and laws, which can discuss topics like safety, security, required equipment etc. There will be several applicable standards for all organizations dealing with operational technology. The following section shortly introduces standards discussing relevant topics in this study.

IEC 62443[18] is a set of standards by the standardization organization International Electrotechnical Commission. The standards includes requirements and processes for securing Industrial Automated Control System(IACS). In total there are nine standards, a technical report, and technical specifications. The nine standards are classified into General, Policies and Procedures, System, and Component. The part about IACS in general (IEC 62443-1) describes terminology, general concepts, and models. Policies and procedures (IEC 62443-2) are the foundation for requirements on the two last parts. The system (IEC 62443-3) part defines requirements for cybersecurity management during the design and system life cycle. The component (IEC 62443-4) part focuses on securing the system-level design and risk assessments. Component regards cybersecurity concerning the design and life cycle of specific components.

In 2018 The Norwegian Oil and Gas Association released a guideline popularly called GL070 describing "Application of IEC 61508 and IEC 61511 in the Norwegian Petroleum Industry (Recommended SIL requirements)". It includes many of the relevant regulations for the petroleum industry[19]. The guideline is for this study especially relevant in its discussion of flare systems. It refers to API-521 section 5.7.9.8 and NORSOK P-002 section 21.2 for sizing the flare knock-out drum and requirements for reliable shutdown to minimize liquid flow.

The guideline states that a level alarm high high (LAHH) in the flare knock-out drum can cause facility-wide process shutdown (PSD). A LAHH is an alarm indicating that the levels are reaching an unwanted high level and that action must be taken instantly. It generally causes global shutdown because it is difficult to detect where the overfilling originates. In case of overfilling, the KO drum should close the feed to the vessel. Inlets to the installation and separator to be closed. In this case, both emergency shutdown (ESD) and PSD logic must be able to shut down the system. The guideline recommends using two separate level transmitters for initiating the ESD and PSD.

The NORSOK standards are a collection of standards used in the oil and gas industry. It is a collaboration between federate and industrial resources. The S-001 standard[20] describes technical safety, which includes description and requirements for several of the components discussed in this study.

S-001 describes the emergency shutdown (ESD) principle hierarchy, which is events initiating an EDS and which response functions are executed. Abandon platform shutdown (APS) is the most extreme state, which automatically initiates ESD levels 1 and 2 and process shutdown (PSD). As it is a hierarchy, one protocol initiation will initiate the lower protocols. The standard describes the events causing a shutdown protocol to be initiated, and which actions to perform on each level.

The standard also describes blow-down (BD) and flare/vent systems requirements. The BD system must be independent of hydraulic power, instrumented air and uninterruptible power supply (UPS). Knock-out drums must be dimensioned to contain liquid over a predefined period without restricting gas outlet flow. There are also remarks about the allowed thermal radiation for flares and maximum gas concentrations from cold vents. BDV shall be treated as ESV with remote and local position indications. It is also possible to have a remote reset of BDV. The flare/vent system normally interfaces these safety functions:

- Process safety
- ESD
- Gas detection
- Fire detection
- Passive fire detection

S-001 requires all facilities to have a Critical Action Panel (CAP). The CAP is a secondary operator interface that is not dependent on any programmable devices. The CAP's main function is to provide ESD functionality in case of primary operator screen black-out. It has both status and control functions. None of the status and alarm capabilities are specific towards the flare system, but it includes a gas alarm. It can control several things,

but the activation of APS, ESD 1 and 2, and BD are relevant to the flare system. The control functions are independent of any networking and are implemented using hardwired push-button circuits.

2.4 Flare Systems

The Norwegian Petroleum Safety Authority (PSA) defines a flare system in the petroleum industry as a "*Secure system for collecting hydrocarbon gas and liquids from blow-downs and pressure bleed-off/ventilation of equipment. Gas and liquid are led through flare piping to a knock-out drum for separation. The gas is flared, while liquid is returned to the process*"[21]. A flare system is a safety instrumented system (SIS) required in industrial facilities that handle more than 1000 kg of pressurized hydrocarbons at a time[20]. The purpose is to burn off excess flammable gases that cannot be processed or stored safely. Burning is done to prevent the release of hydrocarbon gases into the atmosphere, where they could cause fires or explosions. Burning off such gasses is costly for the companies and damaging to the environment. It is used only for safety purposes.

There are different types of flare systems. The main types are elevated flares and ground flares[22]. Elevated flares are flares where the flame tip is raised significantly over the ground. This category can be further split into sub-categories based on; the number of points, pressure, and assist gas. For ground flares, the flare tips are at or near ground level. Ground flares are divided into sub-categories based on if the design is open or closed.

A pressure relief valve (PRV) is a type of safety valve used in various engineering applications to protect pressurized systems from overpressure. A PRV may also be called a pressure safety valve (PSV). The terms PRV and PSV are sometimes used interchangeably, but PRV is generally the term used in litterature. The working principle of a pressure relief valve is based on the balance between the system pressure and the force exerted by a spring or other mechanical device. When the system pressure exceeds the set pressure of the relief valve, the valve opens to release the excess pressure. They are often calibrated to open gradually.

A blow-down valve (BDV) rapidly depressurizes a system by discharging fluid directly into the atmosphere or a disposal system. Blow-down valves are typically used during emergencies, such as when a vessel needs to be quickly emptied or depressurized. Unlike a pressure relief valve, which operates automatically, a blow-down valve is manually operated by an operator or a control system. Meaning that it can be controlled trough programming.

On a typical flare system, several relief- and blow-down lines are gathered in headers and routed to a knock-out drum. Hydrocarbon liquid and water are separated from the vapour stream and may be recovered. From the knock-out drum, the gas is directed into the flare stack. The flare stack might include a second flashback prevention section (such as a molecular seal or a velocity seal) near the flare tip. Seals are normally not used for facilities with burning flares, as the positive stream of gas will stop air from entering the system. If seals are used the vapour stream passes to a flare seal drum after the knock-out drum, which is partially filled. The vapours bubble through a dip pipe to maintain a positive pressure in the blow-down headers and prevents any flashback from the flare tip into the upstream system.

Typically two flare purges can be included; if so, one set of purges is located at the extremities of the relief /blow-down headers, and a second individual purge is near the base of the flare stack. These purges ensure that positive pressure is maintained in the relief headers and a sufficient forward flow of material to prevent air diffusion into the flare stack. One or more pilot flame is provided at the flare tip to ensure the combustion of any vented materials, with a suitable ignition system present. Steam is often added to provide an external momentum force which contributes to smokeless burning. There may also be an alternative gas recovery system in place.

In emergency situations, the flare system can be manually activated by facility personnel. This can be done in a control room or at a remote location. Manual activation is typically reserved for emergency situations where automatic activation is not possible or ineffective, such as in the case of a power failure or equipment malfunction.

2.4.1 Flare Systems and Safety Incidents

This section takes a closer look into incidents where flare systems have been involved or have had a negative impact on the course of events. An incident is defined as a past hazardous event[23]. The Norwegian Petroleum Safety Authority (PSA) oversees the Norwegian petroleum industry's health, safety, and environment. PSA conducts supervision for functional facilities and investigations and forensic work after incidents. Their publicly available reports provide insight into which faults or vulnerabilities might reside in flare systems.

In the period between 19.-21. of March 2014, PSA conducted a major accident audit at the Kårstø Equinor facility[24]. The audit resulted in one non-conformity regarding pressure relief being identified. PSA mentions several points of improvement, among others:

- There was no documentation proving that the system is capable of providing the required pressure relief.
- The flare system was not sized for concurrent pressure relief from several points.
- The system implemented to avoid over-pressure of the flare system was not sufficiently documented.
- Manual opening of pressure relief valves could only be done using a touch screen.

At another Equinor facility, an investigation was conducted after a hydrocarbon leak on the 26th of May 2012[25]. During a shutdown to perform leak testing in two emergency shutdown valves (ESV), a third valve was exposed to too high pressure, resulting in it exploding, and the pipe started leaking gas. The pipe containing the first valve was designed to withstand a 130 bar pressure, and this valve was opened manually. The valve downstream was only designed to handle 16 bar but was exposed to 129 bar. The gas leak is estimated to have released around 3500 kg of gas over a period of 252 seconds. There were flammable concentrations of gas current for around seven minutes. If the gas had been ignited during these minutes, it was estimated that it could have caused a four-minute jet fire. This would have led to considerable damage, and a potential explosion could have been fatal. The underlying causes were technical, operational and organizational. The technical was pointed out as insufficient design, which made it possible to expose parts of the pipe leading to the flare to over-pressure and not being robust enough to withstand human errors.

A few years after this, a Conoco Philips facility experienced an oil spill through the sea drain system after a shutdown the previous day[26]. On the 6th of August 2014, all power and systems were shut down at facilities Eldfisk A, FTP and Embla, while Eldfisk E lost main power. The shutdown happened because of a faulty electrical circuit. On start-up the following day, high levels were detected in the knock-out drum in the flare system. The high level was caused by a closed outlet valve which was then opened. After the valve was opened, an alarm was raised on high levels in the oily water drain caused by

the flow from the KO drum. The operator detected a leak, and through the next 30 minutes, several alarms were raised, indicating liquid accumulation in the flare system. These were assumed to come from the pressure relief during the emergency shutdown. Reports of extreme heat from the knock-out drum lines were detected, and investigation showed that a pressure relief valve (PRV) was left open. An oil spill was detected early the next morning, and operators identified that the storm drain was full of oil and was causing the leak.

The investigation of the Eldfisk incident reveals that the cause of the leakage was the open PRV which resulted in oil being directed towards the flare system and drained out into the sea. Known problems in the human-machine interaction, the number of alarms and the fact that some alarms were conflicting made identifying the issue difficult. In addition, there was no complete overview of the statuses of all the PRVs. If the closed liquid drain in the knock-out drum had not been detected, the flare system could have been filled with oil. If a need for pressure relief had occurred at that time, it could have resulted in a hydrocarbon leak, fire or explosion.

An example of an incident with a fatal outcome occurred in Poza Rica, Mexico, in 1951[22]. A malfunction in the waste flare system caused toxic gases to spread in the local community, claiming the lives of 22 citizens and hospitalizing 320. Even if the specific malfunction might not be as relevant today, it illustrates the potential impact a major accident involving flare systems could have.

The Texas BP refinery incident[27], also known as the Texas City refinery explosion occurred on March 23, 2005. This incident was not directly tied to the flare system, but it included the overfilling of liquid hydrocarbons. The incident was triggered by the ignition of flammable hydrocarbons released from an isomerization unit, causing a series of explosions and fires. The explosion and subsequent fires resulted in 15 fatalities and injured more than 170 people. Investigations revealed significant safety lapses, including inadequate training, equipment failures, and organizational deficiencies.

2.4.2 Flare system hazards

Denham and Donnelly reviewed hazards encountered in flare systems for both on- and offshore facilities[22]. A hazard is a potential source of harm. If the flare system malfunctions, hazardous situations can occur. Table 1. render a short summary of the hazards connected to technical equipment described by the article, along with causes, consequences and mitigating measures. In addition, they describe hazards associated with; heights, personnel working on flare systems, and environmental and particular hazards in offshore systems.

Table 1: Flare system hazards

Hazard	Cause	Consequence	Mitigation
Hazards of liquid overfill and liquid slugging	Liquid entering flare system or liquid pocket accumulation	liquid rain-out from the flare tip, loss of containment due to liquid hammer, overpressure of upstream vessels, embrittlement and hydrocarbon release into liquid waste are	Effective level measurement in the knock-out drum and automatic pump out of knock out drums on high liquid levels
Hazard of flame out	Lack of fuel, ignition or air	Flash fires or vapour cloud explosions	Monitoring of ignition trough thermal, CCTV or thermocouples.
Hazard of flaring toxic streams	Flame-out of toxic materials flare systems	Toxic streams released into the environment and surrounding area	Prevention of flame-out, monitoring of ignition system and high-reliability re-ignition system
Hazard of air ingress	Air entering the flare system risking uncontrolled combustion in other areas than the flare tip	Deflagration or detonation inside the flare system	end-of-header purges, partially liquid filled flare seal drums and secondary gas purges
Hazards of Blocking the Relief Path	Blockage or partial restriction of flare systems due to closed valves, ice or solids build-up	Loss of containment with potential for fires and explosions	High-pressure detection and field emergency response exercises
Hazards of heat and cold	Heat emitted by flares, or failure of metal components due to extremely low temperatures	Extreme heat radiation or equipment breakage	separation between flare and personnel, and components that withstand worst-case temperatures.

As offshore systems are relevant to Equinor, these are described more closely. Offshore installations, like oil rigs, are receding off land shore. They are usually reached by boat or helicopter, which raises an extreme safety focus due to an incident's potential impact on human life and the environment. Particular considerations for offshore facilities are:

- Risk of collision with close-moving ships or helicopters.
- Increased impact from wind and waves.
- Corrosion of equipment.
- Limited space for physical segregation.
- Knock out drums and similar equipment in proximity to flammable inventories.

3 Cyber-security

This section presents theory related to cyber-security in OT. Many concepts are applicable from the IT domain[5]. Therefore the section' focus is predominantly on OT specifics. Risk and risk assessments are also introduced in Section 3.1 as they often are included in safety- and security considerations. Section 3.2 summarizes which threat actors might be interested in attacking an OT system. Vulnerabilities exploited and known tactics for operations conducted towards OT and cyber-physical system (CPS) are presented in 3.3. Section 3.4 addressed intrusion detection systems.

3.1 Risk and Risk Assessments

Both risk and risk assessment are central concepts in cyber-security. Risk[28] refers to the likelihood and potential impact of a security breach or cyber incident on an organization's systems, data, and operations. Risks can arise from various sources, including external threats and internal vulnerabilities. Risk can be measured in qualitative and quantitative manners.

A risk assessment[29] is a systematic process of identifying, analyzing, and evaluating potential risks to an organization. This process typically involves identifying assets and their values, identifying potential threats and vulnerabilities, assessing the likelihood and impact of a security incident. After the initial assessment is conducted, the results should be used to develop strategies to mitigate or manage risk. Risk assessments are tools or frameworks used to help an organization prioritize potential security risks and allocate mitigating measures where it is deemed necessary. Well-known frameworks for risk assessment are the NIST Cybersecurity[30] framework and the ISO/IEC 27001 standard[31].

When assessing risk within technical safety, one often considers the likelihood of an event occurring along with the consequence. When assessing risk within cyber-security the question of likelihood is more complex. As all incidents will occur as a result of an intentional action, factors like behavioural psychology and geopolitical climate will become more relevant. One approach is to disregard the likelihood variable all together. This approach is similar to the *"it is not a question of **if** an incident will occur, but **when**"* mindset. By eliminating the likelihood aspect of risk assessments, one is left with assessing the consequences of potential incidents. This is called consequence-based assessment methods.

In a consequence-based assessment method, the consequence of an incident is the central evaluation criteria. The risk is evaluated by the severity of the consequence[32]. The CCE framework discussed in Section 4 is an example of such a method. The result of a consequence-based method will be the cyber-events inflicting the most damage given a set of evaluation criteria, regardless of how likely the event is to occur.

The aspect of determining and quantifying risk is considered to be challenging. As mentioned above the definition of risk will vary depending on the framework and approach used. Additionally, the assumption of expert system knowledge and access to all relevant information is nor always true, as stated by Cherdantseva et. al.[32]. As a result, there is no formalized way to quantify risk or separate it into categories.

Some common classifications when discussing likelihood are; "very unlikely", "unlikely", "likely", and "very likely". When discussing consequences the categories may be; "tolerable", "acceptable" and "unacceptable". The definition and quantification of risk is, in

other words, central for the outcome of a risk assessment. As risk- and consequence assessments usually are performed by a selected group, the determining of risk is also biased by subjective opinions.

The bow-tie technique is a visual analysis tool used to assess and manage risks. The technique is widely used in the safety- and reliability domain[33]. Its name derives from the shape of the diagram used, with the main risk event located at the center, forming the knot of the bow tie. The left side of the bow tie represents the pre-event scenario, highlighting the causes and factors that contributing to the event. On the right side, the post-event scenario depicts the potential consequences, including both immediate and long-term impacts. The technique also incorporates mitigating measures on the left side and recovery measures on the right side[34].

3.2 Threat Actors towards Operational Technology

Threat actors are individuals or organizations that could damage a system. Threat actors can act intentionally or unintentionally (unintentional threat actors could be sloppy employees or natural disasters). This study focuses on intentional threats, often referred to as attackers or adversaries. There are different categories of adversaries, which are listed below:

- Hackers
- Consumers/customers
- Competitors
- Hacktivists/extremists
- State nations
- Criminal organizations
- Insiders

Some general attributes can characterize these threat actors. Zografopoulos et al. presents an adversary model formulation[35]. This adversary model can be used to assess different adversaries and their capabilities of damaging the system. Their model is composed of four dimensions:

1. Adversary knowledge: Strong-, limited- or oblivious-knowledge
2. Adversary access: Possession of physical devices or no possession of physical devices.
3. Adversial specificity: Targeted- or non-targeted attacks
4. Adversial resources: Insufficient resources or sufficient resources

The characteristics can, for simplicity, be classified on a scale from low to high. Motivation is usually included to describe the adversary's desire to obtain their goal. Highly motivated adversaries are likely to use more resources and be more persistent than the ones with low motivation.

3.3 Vulnerabilities in Operational Technology

As mentioned in the previous section, OT security has traditionally relied heavily on air-gapping and network segmentation, using the Purdue model, firewalls, and proxies. As the connectivity of industrial environments increases, so does the attack surface. Characteristics of OT systems have led to built-in security measures like authentication and encryption being used much less than in the IT domain. As the attacker's goal is to impact the system's physical components, attacks towards OT systems are conducted differently than pure IT systems.

This section introduces some of the most well-known malware used towards OT. Then, common attacks tactics on ICS and known attack techniques are briefly introduced to understand how an adversary designs and launches an attack towards an ICS. Resources by SANS, Mitre, and Dragos are well suited to form an understanding of adversary behaviour.

3.3.1 Previous Attacks Towards Operational Technology

Stuxnet is a sophisticated computer worm first discovered in 2010[36]. The operation was using a malware aiming to alter the physical behaviour of a uranium enrichment facility. The mechanics of the Stuxnet attack involved several stages; initial infection, propagation, targeted payload and subverison of control. Stuxnet primarily spread through infected USB drives, exploiting vulnerabilities in Microsoft Windows systems. It employed multiple propagation methods to spread within the target network. Among these were stolen digital certificates to make itself appear legitimate. The worm exploited a zero-day vulnerability in Siemens' Step7 software and PLCs. Once inside the target system, Stuxnet modified the code running on the PLCs, causing them to operate in an abnormal manner while simultaneously hiding its presence. This manipulation led to the physical destruction of the facilites centrifuges.

In 2017 hackers used the malware Triton to try disrupt safety instrumented systems in a Saudi Arabian petrochemical plant[4]. The main goal of the attack is not certain but was likely to cause production shutdown or drive the system into an unsafe physical state[37]. The attackers gained initial access to the targeted network through a combination of social engineering and spear-phishing techniques. The malware itself targets Schneider Electric's Triconex safety instrumented system and exploited a vulnerability in the programming modes to inject malicious code. By injecting code into the firmware, the attackers sought to disable or manipulate the safety controls. The attack was unsuccessful due to a bug in the malware but had the potential to inflict great damage if the safety systems had been taken down.

Stuxnet was the first attack against ICS to gain society's attention in 2010. The Ukrainian power grid was attacked twice in 2015 and 2016 by hackers using BlackEnergy3 and Industroyer, respectively[36]. The adversaries successfully took down their target, causing a temporary power outage. Then, as mentioned, the Triton malware was discovered in 2017. All of these examples are attacks using malware specifically designed for ICS to disturb or destroy the physical processes. Alladi et al. performed a case study on several cyber-attacks towards ICS[36]. They comment that a normal approach when attacking such systems is to use phishing emails or insecure internet connections to access the OT network. The attackers then inject malware or exploit devices with inherited- or unpatched vulnerabilities.

As proven by the attacks on the Ukrainian power grid, cybercriminals can affect civilians through their operations. The potential impact on humans and the environment can be catastrophic as OT plays a central role in everything from utility to water and food production. While not all IT systems will affect the public if taken down, almost all OT systems might cause a negative impact if destroyed or tampered with.

Not only cyberattacks intended for ICS can affect the OT environment. In 2019 the Norwegian utility company Norsk Hydro was struck by a ransomware attack[38]. Parts of their operations were switched to manual mode and procedures to ensure safe and secure production. Other parts had to be shut down temporarily. The reasoning behind the automated operation and partial shutdown was the uncertainty as to which systems had been affected by the malware. The attack is estimated to have cost the company around 800 MNOK[39].

3.3.2 Dragos 2022 year in review

Dragos is a cyber security company focusing on operational technology and industrial control systems. Their 2022 yearly review on the status of cybersecurity for OT/ICS mentions, points out that, that[40]:

- 80% of customers had limited to no visibility into their OT systems. For the oil & gas industry, the number is 76%.
- The oil & gas sector has made a great improvement on interventions caused by poor network segmentation, decreasing by 39%, from 75% to 36%. The same goes for the number of external connections, where the industry is down 47% from 77% to 31%.
- 83% of all vulnerabilities reside in levels 0 to 3 in the Purdue model and include engineering workstations, PLCs, sensors, and industrial controllers.
- Of all reported vulnerabilities, 50% had the potential to cause loss of both visibility and controllability.
- There is recorded a heavy increase of 87% in ransomware attacks toward industrial organizations. The oil and gas industry is currently not among the most targeted victims making out only 21 of 605 incidents. 437 of the incidents were discovered in the manufacturing industry.

In 2022 the seventh ICS-specific malware was released. Piperdream is the first known scalable cross-industry attack framework. It exploits the industrial protocols Codesys, Modbus and OPC-UA to perform reconnaissance and attacks. The malware consists of five components. Mousehole is one of them and is made specifically for interacting with OPC-UA servers running a client-server infrastructure. The client-server structure is used in the case study described in Section 5, making this tool particularly relevant for anyone aiming to attack a system configured similarly to the one in the case study. It can read and write node attribute data, enumerate server namespace and brute force credentials. Dragos states that *"its* [the malware] *emergence is also indicative of the trend toward more technically capable and adaptable adversaries targeting ICS/OT"*[40].

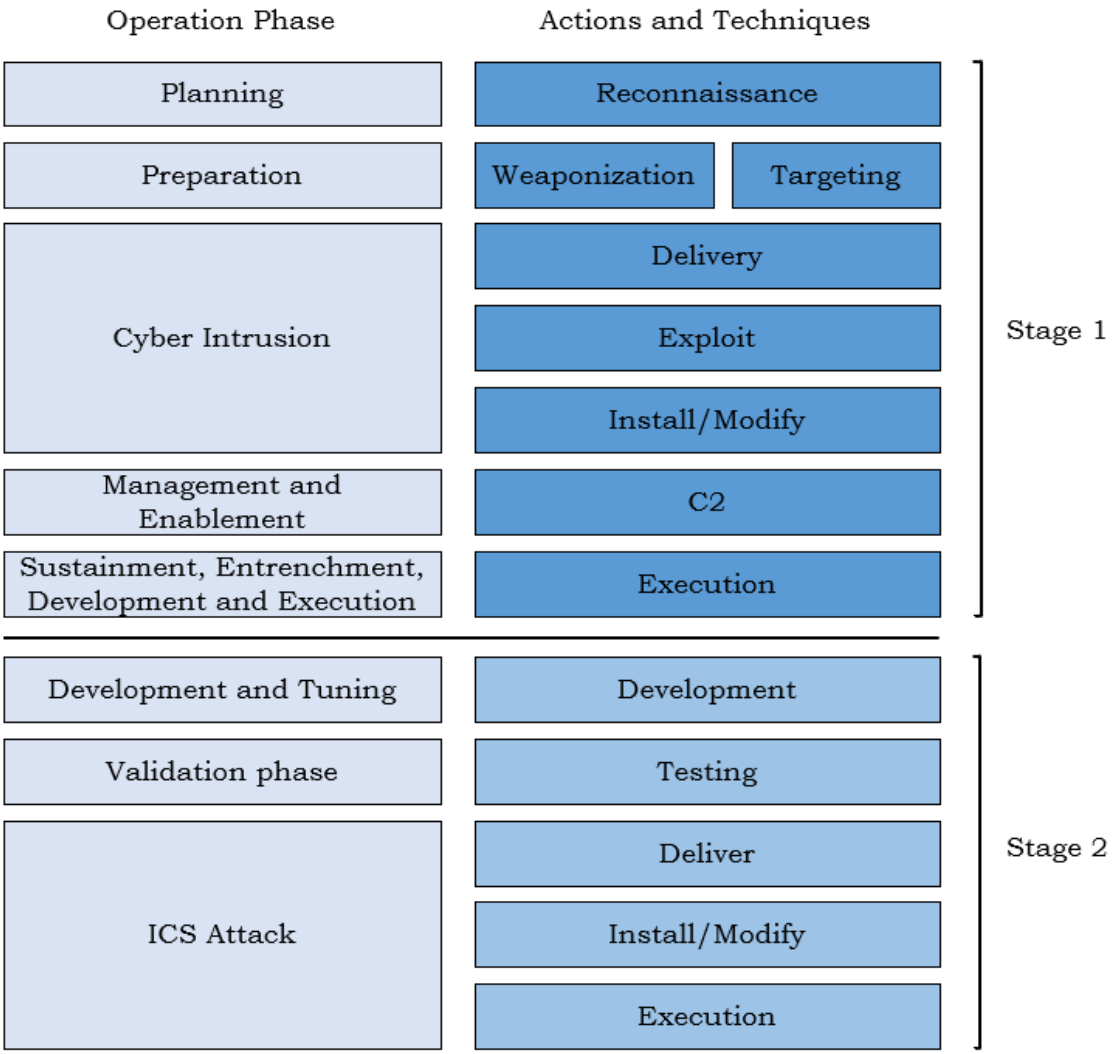


Figure 1: SANS ICS kill chain

3.3.3 SANS ICS Kill Chain

SANS Institute has published an ICS adaption of the cybersecurity kill chain[41]. The chain comprises two stages, each with several steps, illustrated in Figure 1. The operations phase column describes what phase of the operation is currently going on, and the actions and technique column describes the specific actions belonging to the phase.

Stage 1 of an industrial control system attack is similar to a conventional IT attack. Attacks are expected to propagate through the IT network unless they are carried out by an insider. Using separate layers, like in the Purdue model, security is enhanced by implementing several layers of countermeasures before the attacker can reach any physical components. However, directly connecting devices to the internet or using remote access can reduce the effectiveness of this segmentation.

In stage 2 of the kill chain, the adversary aims to develop and deploy an attack to the ICS. Attacks are developed for a specific ICS to obtain the desired effect. Finished malware is normally not tested in the operational environment. This is because the behaviour pattern of the system is familiar to system owners, and abnormal activity could more easily be

detected than in an IT network. Testing is normally done outside of the target system and is, in that case, impossible to detect.

The attacks may involve enabling, initiating, or supporting actions. The attack may also need to be performed simultaneously with evasion techniques to avoid detection by system operators. The level of complexity of the attack depends on the characteristics of the industrial control system, including safety and security protocols.

The difference between attacks towards IT and OT systems is that attacking an OT system requires more work. The attacks are often comprised of several stages, which of an organization's enterprise network might need to be breached before the adversary can even access the OT network. The adversary needs specific system knowledge to implement an attack that can have a physical impact successfully. The work of implementing these attacks is costly, both in the manner of time and resources. This means that attackers who succeed are highly motivated, persistent and likely to be very resourceful.

3.3.4 Mitre ATT&CK

The Mitre ATT&CK database[42] is a collection of known vulnerabilities, exploitations and adversary tactics. It has its own section for ICS and uses 12 different categories to describe tactics used by adversaries when conducting an operation. An operation is a coordinated series of actions taken by an individual or group to exploit vulnerabilities in a computer system, network, or device with the intention of causing harm or accessing sensitive information. Each tactic has a sub-page with a definition, a brief description and an overview of techniques.

3.4 Intrusion Detection Systems

An intrusion is a set of actions performed by an ill-intending actor that compromises the security objectives of a system. To avoid unwanted incidents, intrusion detection can be used to identify such intrusions. There are various methods to detect malicious activity. The most well-known are signature- or anomaly-based detection. Academical publications also introduce the concept of specification-based detection [43][2][44].

Signature-based detection matches behaviour towards a set of predefined rules called signatures. Anomaly-based detection performs detection using machine learning methods on a perceived model of the system. It can detect attacks signature-based methods do not but suffer from a higher degree of false positives. Specification-based detection mixes signature- and anomaly-based detection techniques to reduce the negative effects of the inability to detect new attacks and high false positive rates. Signature- and specification-based methods are discussed in this study, so they are described further in the following subsections.

Detection can be performed on a network- or host basis[45]. Network-based systems analyze packets or flow at different points in the network. As most attacks enter the OT environment through an external network connection or the enterprise network, this gives the opportunity to detect an intrusion before it has propagated through the whole network. Network-based methods are vulnerable to encryption and evasion techniques[46]. Host-based detection analyzes resources on the individual device to determine if there is an ongoing attack. Using host-based detection, insider attacks or high-jacking devices can be detected earlier than with network-based detection. Since the device decrypts all data,

encryption is not an issue.

3.4.1 Signature-based Detection

Signature-based detection, often also called misuse-based detection, is a technique where known malicious activity's signatures are matched toward system behaviour [47]. Figure 2 illustrates the general function of a signature-based IDS. A pre-processor processes packets or data before being sent into the actual detection module. Different matching algorithms can be applied, but exact pattern matching is usually implemented [48][49]. Exact pattern matching means that every piece of data is compared to all instances stored in the database. If suspicious activity is detected, it can be logged or raised as an alert depending on implementation.

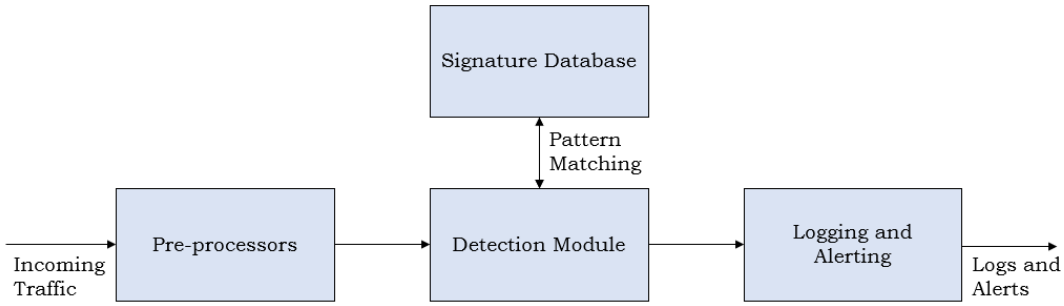


Figure 2: Working principle of signature-based IDS

If signatures are implemented in a satisfactory way, the IDS will not produce false positives, which is a concern in industrial and cyber-physical systems[5]. They are also easier to implement than anomaly- and specification-based techniques as they do not require any extensive modeling or learning stage. On the other hand, they require more maintenance as new signatures must be added to the database as new attacks occurs. This inability to detect zero-day attacks is considered the greatest disadvantage of signature-based IDS. Another drawback is that as the signatures database keeps growing, the demanded memory capacity and runtime of matching algorithms will increase. If the worst-case run time of the matching algorithm becomes too high, it can result in the IDS dropping packets or traces with malicious content, making it vulnerable to DoS attacks.

3.4.2 Specification-based Detection

Specification-based detection techniques were introduced to combine the strengths of misuse- and anomaly-based techniques. This means that the expected behaviour model is manually implemented as in signature-based methods, but the information of current system behaviour is analyzed as in an anomaly-based method. The key is adding expert knowledge about the system's behaviour and specifications[50].

Specification-based detection can detect zero-day attacks as anomaly-based methods while maintaining a relatively low false positive rate. The drawbacks of the two other detection categories are greatly reduced. A concern is that even if the false positive rate is reduced, it is not eliminated. In some systems, it might not be acceptable to have any false positives.

Developing a specification-based IDS comprises four stages, as Figure 3 illustrates. All the different stages can be performed using different methods[43]. First, the source of the specifications needs to be determined. The source of specifications is usually; protocol specifications, reference models, or observed behaviour. Then, the system specifications gathered from the source need to be analyzed to define correct behaviour. This process is called specification extraction and can be done manually or automatically. In the third stage, the behavioural model to be used in detection is implemented before the detection functionality is included in the final stage. Developing behavioural models in large, heterogeneous networks is costly. It requires knowledge about all system parts, including vendor-provided equipment.

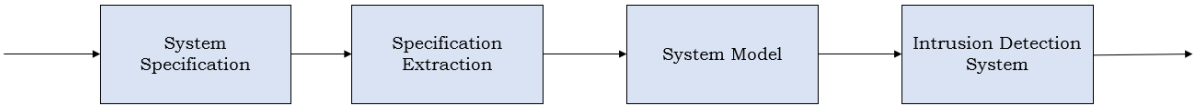


Figure 3: Development process of specification-based IDS

3.4.3 Detection Metrics for IDS

In order to assess the quality of an IDS in an OT envrionment, performance metrics should be used. Since the working principle and techniques used to implement detection systems differ, it is challenging to determine a common ground for quality metrics. This is an issue that has been addressed by several studies[51][52][2].

The characteristics of good detection systems can be summarized as; broad detection range, economy in resource usage, and resilience to stress[53]. Broad detection range means that the IDS should detect as many attacks as possible. Within the "broad detection"-metric are sub-metrics; accuracy, sensitivity, specificity, and computational time. Monitoring and detection should use a sustainable amount of resources, which is the economy in resource usage. An IDS should also function correctly in stressful situations or environments, e.g. during abnormal operations or periods of high packet load.

The most extensive studies on this IDS performance metrics are from the IT domain. To cover the OT domain sufficiently, a few additional points should be added. Reducing the probability of false positives in OT is more important than in IT. False positives can result in unnecessary escalation and disruption of production. Determining if an attack has been successful is important in recovery work to identify which assets might be affected[51]. The ability to interpret industrial protocols is essential to understand the current status of

physical processed. The ability to interpret a protocol increases the probability of attack detection, the ability to detect new attacks, and the ability to identify specific attacks[5].

Ease of use, number of installations, and upgrades indicate the competence required to handle the system can also be considered performance metrics for IDSs in operational environments[5]. External resource requirements like cabling and changes in network topology should also be considered[18]. For example; a poor IDS would require reconfiguration of devices serving a critical function in the system. Such reconfiguration might require temporary system shutdown or additional resources such as cabling.

4 Consequence-based Cyber-informed Engineering

This section introduces the Consequence-driven, Cyber-informed Engineering (CCE)[6] method. The method is applied on a flare system case study in Section 5. CCE aids organizations identify critical system components in ICS by examining the impact of cyber-security events. More specifically, the main goal is to identify and evaluate the worst possible outcomes, so-called high consequence events (HCE) and how they can be mitigated.

When implemented, it helps the organization[6]:

- Identify critical system components and subsystems.
- Assess the applicability of an attack scenario against critical system components.
- Gain insight into organizational and technical assets.
- Identify key technical information about the organization.
- Suggest recommendations to harden the system and operations against sophisticated adversarial attack/manipulation.
- IEC 62443[18] suggests that all organizations should have an initial cybersecurity assessment. The information collected can be used as a part of this assessment.

CCE was created by the Idaho National Laboratory (INL). INL is one of the national labs founded by the U.S. Department of Energy researching secure solutions for nuclear energy, clean energy options and critical infrastructure. The background for developing this method is that standard IT security systems and risk assessment methods are insufficient for ICS[54]. Most ICS is designed to meet automation- and safety requirements, which is not common in IT systems.

CCE comprises four stages; consequence prioritizations, system-of-systems analysis, consequence-based targeting and mitigations and protections. In consequence prioritization stage the goal is to identify high consequence events from potential cyber events. The system of system analysis analyzes the interconnectedness and dependencies of the organizations systems to understand their vulnerabilities and potential cascading effects of the HCE. The information database built in the system of system analysis is used in the system targeting stage to identify how an adversary could launch an attack toward the system initiating the HCE. Finally the mitigations and protection stage identifies strategies, measures and safeguards to protect the system from the attack scenario developed in the system targeting stage. Figure 4 illustrates the different stages and focus points for each stage. The following subsections introduce each of the stages more closely.

4.1 Consequence Prioritization

Consequence prioritization[54] is the first
stage, where the organization identifies its primary purpose and associated processes. Events that can compromise the organization's ability to fulfil its primary purpose are called cyber events. The organization's physical infrastructure and interdependence on other systems must be considered when developing different event scenarios. Cyber-physical systems have many components and sub-systems. A complete analysis of an

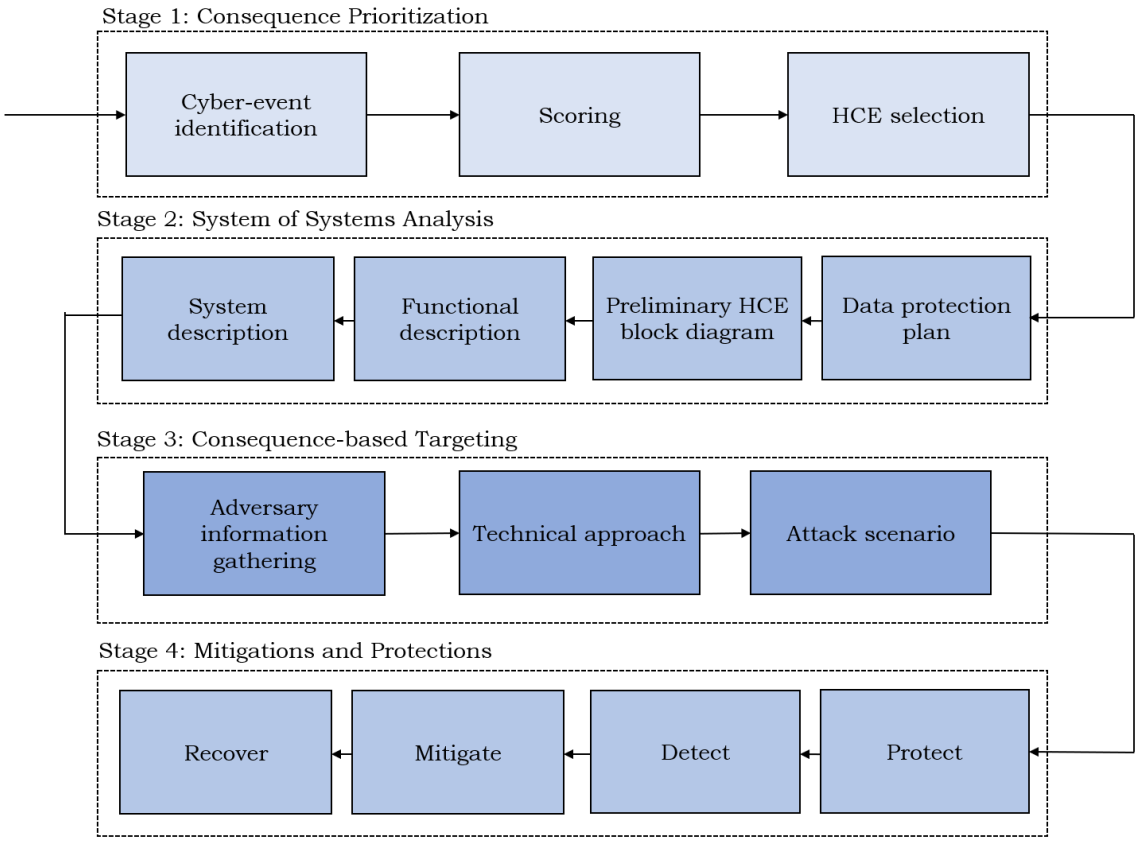


Figure 4: CCE assessment flow

organizations can be complex and time-consuming, therefore it is necessary to prioritize events. Prioritization is done by assigning scores according to different criteria within a category. Each organization must evaluate which categories are relevant for them, but INL suggests the following:

- Area impacted
- Duration
- Attack Breadth
- Safety
- System integrity confidence
- Cost including restoration

The scores are represented by the numbers 0, 1, 3, and 5, which respectively correspond to the consequence being ”none”, ”low”, ”medium”, or ”high”. The criteria thresholds are unique for each organization and need to be established as a part of the process. When the different scores have been appointed, weighting coefficients should be determined for each criterion, reflecting its importance.

$$\text{Scored Impact Points} = x_1(\text{area impacted}) + x_2(\text{duration}) + x_3(\text{attack breadth}) + x_4(\text{system integrity}) + x_5(\text{safety}) + x_6(\text{cost}) \quad (1)$$

$$\text{Maximum Impact Points} = x_1(5) + x_2(5) + x_3(5) + x_4(5) + x_5(5) + x_6(5) \quad (2)$$

$$\text{Severity Score} = \frac{\text{Scored Impact Points}}{\text{Maximum Impact Points}} \cdot 100 \quad (3)$$

Using Equations 1-3, the scored impact points is calculated for each cyber event. X is weighing coefficient for each category. To have a comparative basis the maximum impact points for all the categories is calculated in Equation 2. The severity score for each event is then obtained by scaling the scored impact points for each cyber event to the maximum impact points over all categories. The severity score is used to evaluate the different cyber events using a threshold decided by the organization. Events scoring above the threshold are classified as high-consequence events (HCE). The cyber events classified as HCEs are the ones brought onward to the following stages, and can be one or several events.

4.2 System-of-systems Analysis

System of systems analysis[55] is stage two. Here the organization evaluates all relevant material for the HCE. The goal is to obtain complete knowledge of systems involved. By having complete knowledge of the systems, devices and processes it is easier to assess all potential consequences. Potential consequences also includes unintended consequences. The stage involves analyzing infrastructure and operational processes. Specifically, the organization should identify the following for each HCE:

- Equipment
- Systems
- Processes
- Operations
- Maintenance
- Testing
- Procurement practices

The analysis should also include available information on subcontractors, vendors and suppliers. Sub-phases are used during stage two to formalize the process in a helpful manner. The sub-phases are illustrated in row two of Figure 4.

Establishing a *Data Protection plan* initiates the stage by deciding how the data aggregated should be protected. The goal of the data protection plan is to hinder the adversary from gaining access to information that can give them perfect system knowledge. *Preliminary*

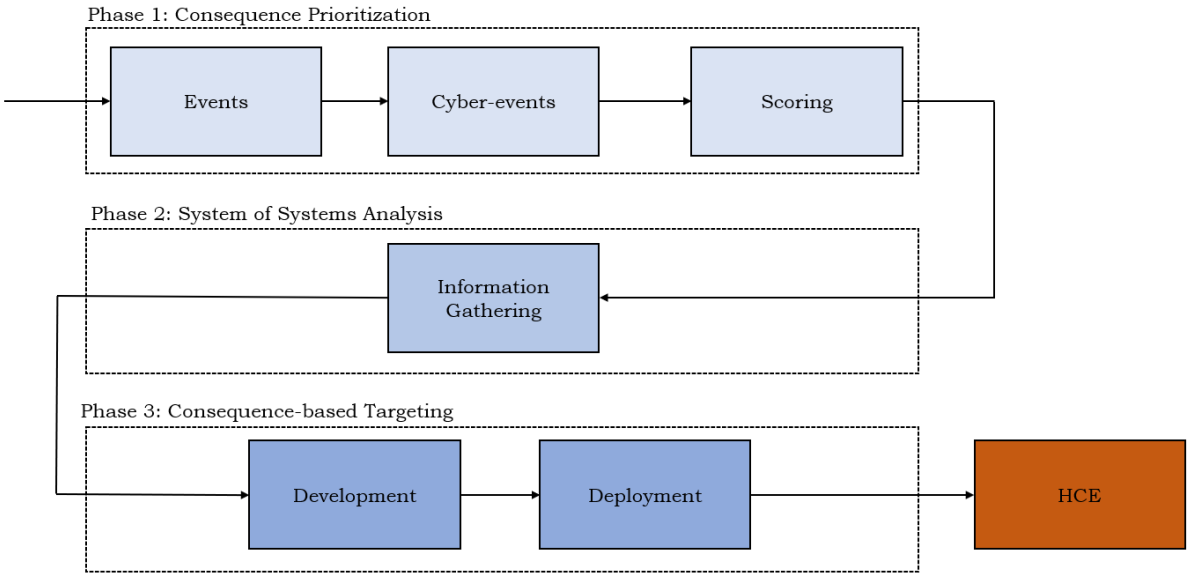


Figure 5: CCE kill chain

HCE block diagrams are high-level diagrams describing the HCE. These block diagrams provide a starting point for organizing the information gathered. *Functional description* connects the block diagrams and system information. *System description* is the final sub-phases which consists of identifying key details and information during a cyber attack. One technique to organize the information in the system description is to create a taxonomy of all enabeling and critical functions associated with the HCE.

4.3 Consequence-based Targeting

Using the system description produced in stage two, consequence-based targeting[56] aims to simulate how the adversary would attack the system. Each simulation of a possible attack is called an attack scenario. An attack scenario contains one or more technical approaches. The technical approach is the stages conducted to beach the system and obtain the HCE.

The focus in this stage is to use an attacker's perspective and identify essential information and resources for an operation to be successful; these are the critical needs. These critical needs are the ones that must be exploited obtain physical effects. Examples of critical needs are mechanical failure analyses documentation or access to an engineering workstation.

INL has developed its own CCE kill chain to illustrate adversary objectives during an operation. The kill chain is not the same as the flow of the CCE method as presented in Figure 4. It rather illustrates the adversaries actions corresponding to the different stages. The kill chain is used to identify so-called choke points, which are activities the adversary must perform to achieve success. The defender can use these points to narrow the implementation of protective measures.

All the technical approaches should be recorded in target details, which are the specifics of how every target could be exploited or manipulated. The target could be a device, process, logic circuit, etc. It can also be non-cyber related, like personnel. This information could

also help identify security aspects previously unknown to system owners and can encourage the implementation of better cybersecurity protocols. Each technical approach should contain a:

- Target access
- Required actions
- Timing for triggering
- Triggering factor

By the end of this analysis, fully developed attack scenarios should be passed on to the final stage. Each attack scenario should contain the following:

- Description of the HCE.
- Target details of involved system/components.
- Technical approach used to initiate the HCE.
- Critical needs to succeed in the technical approach.

4.4 Mitigations and Protections

Mitigations and protections[57] is the final stage of the CCE method. The main goal of this stage is to reduce or eliminate the risk of an HCE occurring. The stage is similar to the NIST framework for handling cyber risks in critical infrastructure[58]. The CCE mitigation stage comprises four functions; protect, detect, respond and recover.

- Protect: Protection techniques that can hinder the adversary from causing the HCE by using cyber means.
- Detect: Mitigating measures identifying malicious activity as fast as possible.
- Respond: Mitigating measure by having well-known concise plans on how to act if exposed to a cyber-attack.
- Recover: Mitigating techniques to restore normal operations after a cyber incident.

The framework suggests using tools like brainstorming to develop good mitigations and protections. The brainstorming could also include personnel not previously included in the CCE process. It might be necessary to prioritize which measures should be implemented where the framework specifies:

- Protection is the only function that can stop the HCE from happening. Mitigating measures either reduces the damage inflicted of the recovery process.
- Mitigating measures might have different efficacy.
- Some attack scenarios are more likely due to the threat landscape.
- Attack difficulty might affect the likelihood of an attack scenario happening.

5 Flare System Case Study

The ideas and main components of the the Consequence-driven Cyber-informed Engineering (CCE) method can be used to assess potential cyberevents in entire organizations or selected subsystems. To research if a consequence-based assessment method can improve intrusion detection a case study will be performed. The theory presented in Sections 2.1 and 3 form a comprehensible picture of the case study system as a part of an operational technology (OT) environment. The case study system is a flare system, as described in Section 2.4. It is a simplified flare system and OT network topology. The case is chosen and provided by Equinor because of its importance in safety procedures.

The section is divided into three main topics; description and introduction to the case study, CCE assessment, and detection parameter identification. First the case system and its components is described. Then, potential vulnerabilities and threat actors are identified using theory from Sections 2.4 and 3. These considerations are not related to the CCE assessment. The second part parforms the CCE assessment on the flare study, before the results are used to identify potential process parameters used for detection.

More specifically, the first three steps of the CCE method are conducted to produce an attack scenario. The consequence prioritization gives us an high-consequence event (HCE), used in the system of systems analysis to collect all the relevant information on the system. This is used to create a plausible attack scenario. The attack scenario is then used to suggest specification-based detection parameters.

5.1 Flare System Description

In the main production process oil, water and gas are fed into the separator through the feed pipe. After separation, oil, water and gas are sent down-process in separate pipes. Suppose over-pressure occurs, the pressure relief valve (PRV) is opened, and excess gas is diverted into the flare system. The flare system comprises a knock-out drum and a flare tower. The knock-out drum ensures that no liquids are sent into the flare tower. Excess liquids are sent back into the refining process through the return pumps. If there is an emergency situation, the excess liquids will be sent into the drain system. The gas is sent into the flare tower, where it is burned in a controlled manner. The flare tower has a pilot flame that constantly burns to ensure the relieved gas is ignited. The flare system is illustrated in Figure 6, where the red area is the flare system, and the blue area is one part of the main process included to provide the reader some system context.

The separator and knock-out drum have a level transmitter, level switch and temperature sensors for gas and liquids. The level transmitter provides a continuous analogue measurement of the tank. Usually, this measurement is given in percentage according to a specific interval. The level switch is a binary level measurement sensor indicating if the level is above or below a fixed point. The gas temperature sensor is placed at the top of the knock-out drum to measure the temperature of the gas flowing towards the flare tower. The liquid temperature sensor is placed as the bottom of the drum close to the return pumps and drain valve to measure temperature of liquids in the tank.

The network topology is set up according to the Purdue model and is illustrated in Figure 7. Devices are placed at their respective levels with; sensors and actuators at level 0, PLCs at level 1, HMI and EWS at level 2. Data and logs from the three down-most levels are collected and sent to an IDS and a historian residing at level 3. Moving from the OT to

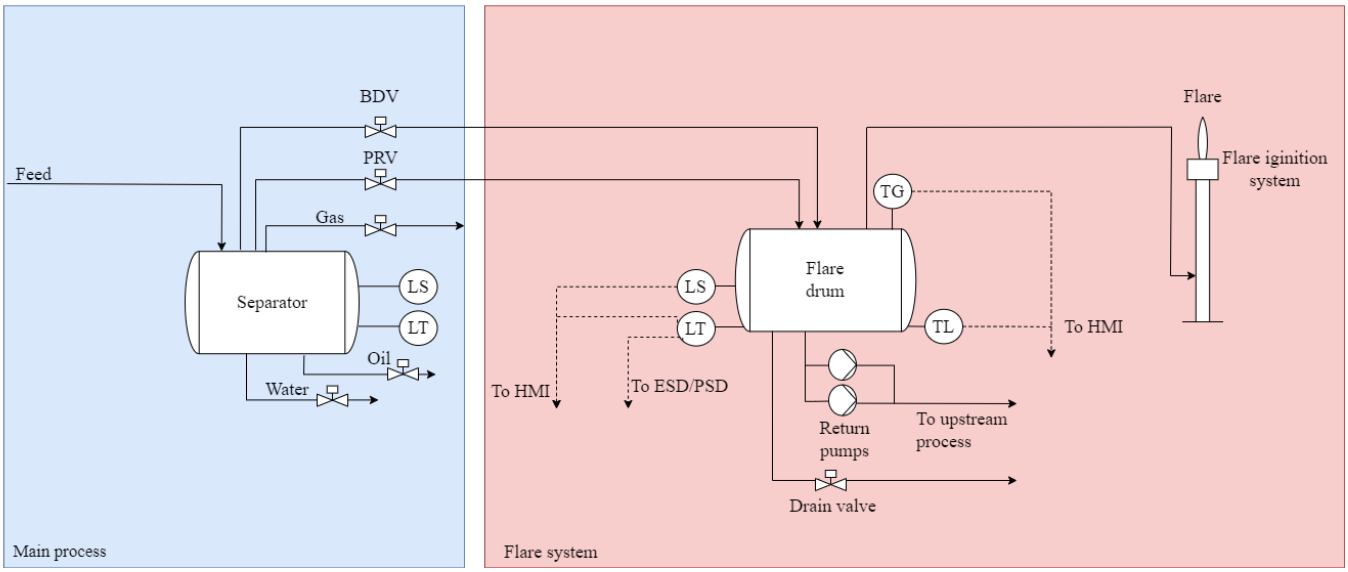


Figure 6: Flare system diagram

the IT domain at levels 4 and 5, data from the historian and IDS is compared and used as, e.g. input for SIEMs.

The IDS at level 3 is a passive signature-based IDS. Passive means that it will not actively respond to malicious traffic to avoid disturbing the process. The goal of this case study is to extend the detection range of the IDS. This will improve the IDS according to the performance metrics described in Section 3.4.3. The IDS gathers layers from the three lowest levels, performs detection and passes on any alerts up into the corporate network. The network traffic used in the existing IDS is gathered using a Switched Port Analyzer (SPAN), which provides a copy of the network traffic.

Levels 2 and 3 use conventional communication protocols for industrial applications. On level 2, OPC-UA is chosen (Modbus, Profinet or similar could have been optioned). Communications between levels 2 and 3 and up use Ethernet TCP/IP protocols. The more unconventional part of the network is Ethernet-based communication is used between levels 0 and 1. As mentioned in Section 2.1, these devices currently communicate using 4-20 mA electrical signals. Ethernet-based communication on the lower levels of industrial applications is currently being researched and developed. In this case study, the motivation for including it is to investigate whether it can be helpful in security applications. Equipment running OPC-UA over Ethernet-APL is under development from e.g. ABB and Mesco[59][60]. Devices on different levels can function as OPC-UA servers and clients depending on whether the device wants to poll or post a state.

5.2 System Vulnerabilities

Flare systems have vulnerabilities that can be exploited to initiate HCEs. This is clear from reading standards and reports (Sections 2.3 and 2.4.1), and collecting experiences from automation engineers employed at Equinor. This section explores what can be present in flare systems and which are relevant to the case study. These vulnerabilities will be used onward when discussing potential cyber events and attack scenarios. It also simplifies the work of identifying process parameters and data points to monitor.

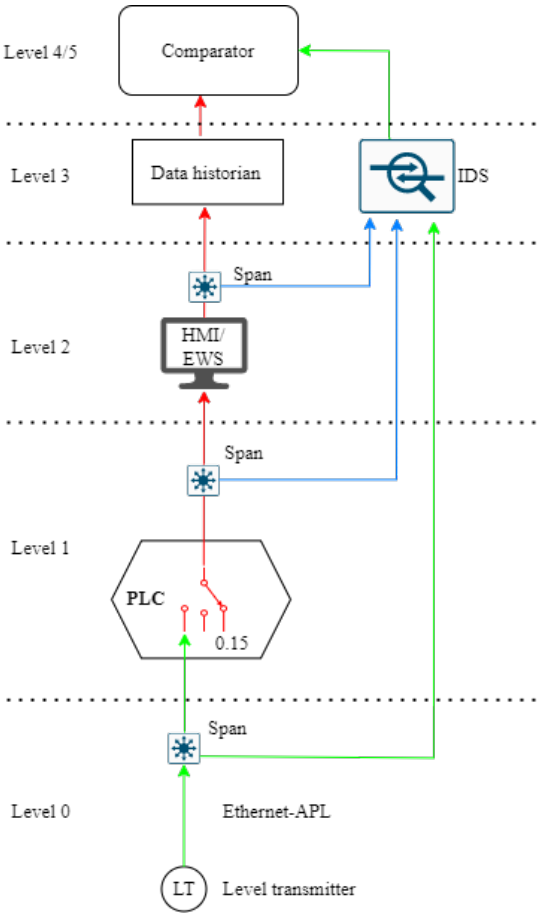


Figure 7: OT network topology

The Norwegian Oil and Gas Association has identified the main scenarios that can cause overfilling of the KO drum to illustrate necessary safety measures. A safety measure is defined as; *"measure intended to achieve adequate risk reduction"*[61]. As mentioned, a LAHH in the KO drum could cause a facility-wide process shutdown. The main scenarios according to Norwegian Oil and Gas Guidelines 70[19] are:

- BDV in open position.
- Oil BDV in open position.
- Overfilling of separators cascading to the flare system.
- Blowdown from flowlines.
- Condensation from spill-off flaring accumulating in the KO drum.
- Rupture disc failure in the external pressurized system causing the excessive flow to the flare system.

As discussed in Section 2.4.2, there are several hazards related to the usage of flare systems. Hazards will be the process causing damage during a cyber event. Some are less relevant for this case study as they are not reachable from the cyber domain. One example is the risk of flame-out due to off-shore wind strength and wave height. Others are very much

relevant, and by connecting them to the relevant cyber events, one can gain better insights into the mechanics of the incidents and maybe, in time, how they can be avoided.

From the audit performed by The Norwegian Petroleum Safety Authority on the Kårstø Equinor facility[24], there were some interesting remarks that an adversary could utilize. First and foremost, there were detected errors in the dimensioning of the system, which would allow over-pressure if mismanaged. Additionally, the system was not built for handling simultaneous pressure relief from several points. As seen in the Heimdal incident[25], poor dimensioning in combination with faults is a realistic scenario. It was pointed out that the pressure relief valve manual control was only available through a touch interface. From a cyber-oriented perspective, a touch-only interface is a significant vulnerability, as an attacker who has already gained system control could stop operators from manually initiating safety measures.

From the Eldfisk incidents[26], it was proved that faulty behaviour in the flare system could have cascading effects. In this case, the fault propagated to the drain system, causing acute oil spillage. Part of the reason this incident occurred was the confusion introduced by the origin of the alarms and trouble identifying the cause of the fault. The delay this confusion led to is an essential point as it proves that an adversary could evade detection by confusing or manipulating systems that are out of sight from the operator's perspective.

All flare systems have some way of igniting the gas being relieved, see Section 2.4. It can be a continuous pilot flare or a technique for igniting the gas when pressure relief occurs. If the system uses a continuous pilot flare, it would require a way to reignite the flare if choked. If an adversary gains control over the mechanisms controlling the flare ignition system, they could put it to cause a gas leak and then reignite after gas accumulation to cause jet fires or explosions.

5.2.1 Threat Actors

Revisiting the different threat actors described in Section 3.2., this section investigates who is more likely to want to attack a flare system and discusses the potential implications for the assessment. The discussion also takes into consideration who might have the means, resources and knowledge to successfully launching such an attack.

Hackers and consumers are the most unlikely adversaries. A hacker's goal is often to prove themselves capable of breaching security parameters or evading detection. Consumers of Equinor's services are several chains down the product chain and will have little to no opportunity to manipulate the system in a way that favours them. Characterized by low to medium motivation, low specific process knowledge, no possession and few resources, it would be improbable that these adversaries would have the motivation, knowledge and resources needed to implement an attack affecting the flare system of a facility.

Competitors is also an unlikely category. While they inherit a high level of knowledge. They are unlikely willing to spend the resources needed to develop and go through with an attack. A competitor would, in this case, be another producer of energy. Their objective could be sabotaging or tarnishing reputation to increase their market share. Additionally, competitors are unlikely to consider the risk of getting caught worth it compared to the potential gain.

The Norwegian National Threat Assessment[62] states that there is a possibility, however

unlikely, of experiencing sabotage from foreign state actors in 2023. These would have been the actors with the highest amount of resources, motivation and knowledge, making them the most dangerous with respect to impact. State nations are most likely to conduct sabotage or denial of service. Examples such operations are the attacks towards the Ukrainian power grid, Triton and Stuxnet.

Even if it is unlikely that state nations will perform sabotage or denial of service attacks on Norwegian soil in 2023, this threat actor is one of the most concerning. Such operations have been conducted outside the country. The energy industry, especially fossil, is currently of great geopolitical importance.

The threat assessment also presents that likelihood of left-extremists or environmental activists performing sabotage is highly unlikely. These attackers are highly motivated but lack the resources and knowledge to go through with such an attack. An attack from ideologists would likely be sabotage or denial of service to support their agenda.

Criminal organizations are the most active threat actor according to Dragos[40]. This is a diverse group where resources and knowledge vary. Cybercriminals' objective is almost exclusively financial gain, and they are usually the ones launching ransomware attacks. As these attacks become more common and sophisticated, this group of threat actors is relevant.

Insiders are employees or people with authorized access to the system. The objective is usually revenge or financial gain. It is considered most likely to be somebody acting on some other organization's behalf willingly or unwillingly, which makes it challenging to categorize this threat actor. An insider has a high knowledge of the process and high motivation. They might be in possession of physical devices and does not require as many resources as they already have access to the system.

To summarize one can say that state nations, criminal organizations, and insiders are the most likely threat actors to attack the system. When conducting the CCE method, the type of threat actor and their objective is irrelevant. Nevertheless, it is included in further discussions of results and the quality of the CCE method itself.

5.3 CCE Assessment

The goal of the CCE assessment is to collect information about the flare system in an organized manner, use it to develop an attack scenario and suggest mitigating measurements. This section assesses cyber events identifies an high-consequence event (HCE). The HCE description forms the basis for identifying critical components and attack tactics. These are then used to develop plausible attack scenarios. The mitigating measures are assessed within the scope of the study. This means that it only focuses on measures in the detect function.

5.3.1 Consequence Prioritization

During the consequence prioritization phase, the baseline assumptions are that access has been achieved and the adversary is knowledgeable and well-resourced. These assumptions mean that the adversary is able to implement the attack. The scope of the assessment is limited to the system described by the case study.

According to the CCE reference document[6], all targets fall into a category. The categories are; "physical infrastructure and interdependencies", "horizontal application of technology" and "reliance on automation and control capabilities". The the flare system mainly fall into the category of "reliance on automation and control capabilities" because it is not directly involved in the production but is necessary for safe operations. It can also be of the sort "physical infrastructure and interdependent" as the production can not continue if the flare system is malfunctioning.

Measuring criteria need to be chosen to evaluate cyber-events (potential HCEs). Below is a selection and reasoning behind the different criteria selected for this study. Table 2 includes examples of different thresholds that can be used for flare systems. The severity classification criteria for duration and cost are taken from the CCE reference document [54] to avoid disclosing sensitive data. The score is then given using the best of knowledge.

- Area impacted: **Not relevant criteria**, as an event will have a different effect depending on the importance of the facility. Incidents at junction point facilities like Heimdal or Kårstø will impact the organization more than isolated facilities like Visund. All events will be more severe if occurring at a juncture point facility. For some events, the area impacted is not applicable because the geographical location of the event will not aggravate the consequences. Because of this, the criteria are taken out of consideration, and the study only focuses on land-based juncture point facilities. However, the coefficient and scoring matrix thresholds are included in the reference matrix to illustrate the process.
- Duration: **Relevant criteria**, duration of outage equals higher financial losses and reputational damage.
- Attack breadth: **Not relevant criteria**. Deemed not relevant for this case study as the case system has very few components. Variability in a few components will have a disproportionately large effect on the event severity impact score. It is also assumed that the vertical movement needed to manipulate the system has been successful.
- Safety: **Relevant criteria**, as the industry has strict safety requirements, and a flare system is a safety measure.
- System integrity confidence: **Not relevant criteria**, it indicates whether the system owner can trust the system's integrity sufficiently to continue operations. The system integrity confidence scoring does not consider that the attacker still has a foothold within system perimeter after an incident, which will lead to unnuanced scoring.
- Cost (including restoration): **Relevant criteria**, as facility shutdowns can become extremely expensive. The main contributors are loss of income and cost related to breach of agreements. Thresholds connected to cost greatly depends on the size of the organization.

Additionally, environmental consequences are a significant concern in the oil and gas industry. Events resulting in oil spillage can have fatal consequences for marine life. Such an event could also cause significant reputational damage.

Based on the organization's concerns, the weighting coefficients are rated from 1 to 3. Duration (β) is given medium priority as it is connected to both costs and reputation. Safety (δ) is one of the primary concerns as the study regards a safety system. Cost (γ)

is given a lower priority. Environment (ϵ) is one of the more significant concerns and is therefore also set to a high.

Table 2: Cyber-event scoring matrix

	None (0)	**Low (1)**	**Medium (3)**	**High (5)**
Duration ($\beta = 2$)	Inconsequential	Return of all service in less than one day	Return to service in between 1 to 5 days	Return to service in greater than or equal to 5 days
Safety ($\delta = 3$)	Inconsequential	Risk of injuries onsite	Risk of injuries onside and offsite	Risk of fatalities onsite or offsite
Cost ($\gamma = 1$)	Inconsequential	The cost is significant, but well within recoverable	Significant cost that will take years to recover	The cost triggers liquidity crisis and potential bankruptcy
Environment ($\epsilon = 3$)	Inconsequential	Spills causing short term effects on biological life	Spills causing long-lasting effects on biological life	Spills causing permanent effect on biological life

A remark can be given about the scoring coefficient for the duration. As this research is conducted during the beginning of 2023, the geopolitical situation has increased the focus on stable power delivery to Europe. Not being able to provide this stability can be extremely costly and damaging to the reputation. Therefore, one could consider setting the duration to a high priority criteria, but it was decided against as this might be a more temporary status.

Using the theory from Section 2.4.2 and discussion about flare system vulnerabilities in Section 5.2 cyber events (potential HCEs) have been identified. These are the events listed in Table 3. For each event, the table gives a brief description and possible outcomes. As stated above, the use case was limited to only regard a land-based facility, which results in oil spillage not being as relevant. It is a highly relevant case for off-shore facilities and is therefore mentioned but not considered. The events are scored within the different categories using separate tables, respectively Tables 15-19, Appendix C.

As the assessment is limited to only land-based facilities, some additional considerations must be made. First and foremost, more space is available than in an off-shore facility. The larger amount of space makes evacuations easier and allows for more spacing between different systems. The flare system can therefore be placed further away from other critical equipment or pressurized hydrocarbons, decreasing the likelihood of, e.g. massive explosions. Leakages of liquids like oil would have a smaller impact than they would off-shore. On the other hand, land-based facilities are placed closer to civilians, even in remote locations. The risk of off-site fatalities is higher than off-shore, which is almost non-existent. There is also more biomass that could escalate potential fires.

Using a Python script, all event severity scores and severity score compositions are calculated based on Equations 1-3. The overall severity score for each event is presented in Table 4. The composition of the severity score based on each criterion is illustrated in Figure 8. The composition explains the impact of each category the cyber events are evaluated within. The highest-scoring events are jet fires or explosions, ignited liquid spread

Table 3: Cyber-events for flare systems

Event	Incident mechanics	Potential consequence
Ignited liquid spread	Overfilling of KO drum causing liquids to enter the flare tower, igniting and raining out over nearby area	Personnel injury
Process shutdown due to compromised integrity (CI)	LAHH in the KO drum activating ESD or PSD	Financial and reputational damage
Process shutdown due to destroyed equipment (DE)	Destruction of equipment	Financial and reputational damage
Oil spillage	Overfilling of KO drum causing oil to be drained out into the environment	Environmental harm
Jetfires or explosions	Flare-out in flare tower or leaks in pipe system in addition with ignition	Fatalities, personnel injury, environmental, financial
Toxic flareouts	Flare-out in flare tower while gas or toxins are being ventilated	Fatalities, personnel injury, environmental

and toxic flare-out. This does not mean that the other events are considered inconsequential, and the final choice depends on if a threshold has been set or if several events should be included in the further analysis.

The most natural event to choose would be jet fires or explosions. However, when designing the technical approach, it became apparent that one was dependent on several external variables for this scenario to happen. These are variables like the physical location of equipment breakage and wind direction which would expand the case scenario outside what is required to illustrate the research topic. Toxic flare-out was another candidate, but the technical approach would include choking the pilot flare or creating a pipe leak. The pilot flare is monitored using a CCTV camera, which would result in detection by image recognition. Image recognition is not covered in the scope of this study. A gas leak from a pipe in a confined space would best be detected using a gas detector, which is not included in the case study. So to illustrate the method working within the constraints of the research questions and case study, the ignited liquid flare event is chosen as the HCE.

Table 4: Severity scoring

Event	Severity Impact Score
Ignited liquid spread	55.6%
Process shutdown - CI	4.45%
Process shutdown - DE	42.22%
Jetfires or explosions	68.89%
Toxic flareouts	55.6%

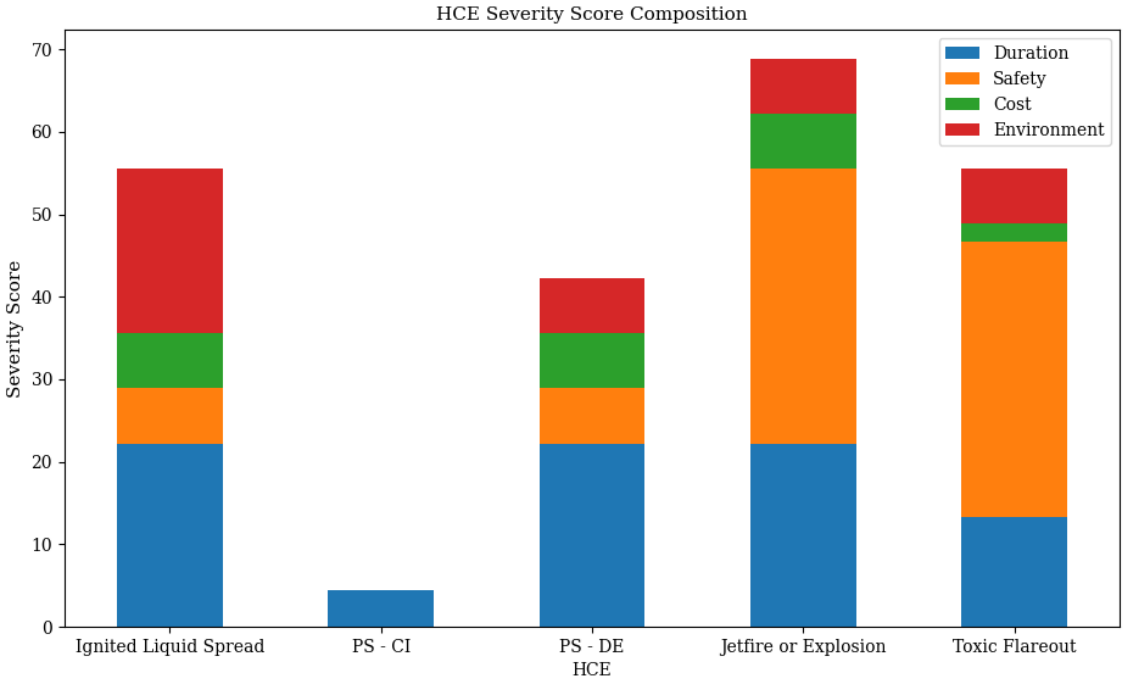


Figure 8: Severity scoring composition

HCE Description - Ignited Liquid Spread

The HCE consists of ignited liquids firing out of the mouth of the flame tower. The event would occur by overfilling the knock-out drum of the flare system with liquids, resulting in the liquids entering the flare tower. Water would evaporate as smoke, but as crude oil has a higher boiling temperature, residues would spread out as rain. The event would cause a production shutdown, hindering a safety-critical system from functioning. The flare system would have to be rinsed to restore production. The primary source of downtime would be forensic work. Production is not likely to precede until complete system integrity is regained. Besides shutting down production, ignited liquids could cause injuries to nearby personnel or fires in contact with flammable materials like biomass.

5.3.2 System-of-Systems Analysis

The system of systems analysis starts by considering who should be able to access which data in the data protection plan. This part is irrelevant in this case study as all data is synthetic. It is also skipped to allocate more time for the remaining assessment.

A preliminary HCE block diagram is created to better understand which components and processes are involved in the HCE. As pictured in Figure 9, one can see that the event starts by liquid entering the knock-out drum. When the knock-out drum overfills the liquids will enter the flare tower. When sufficient liquids fills the flare tower, it will be ignited by the pilot flare, resulting in ignited liquid spread through the top of the flare tower. This information can be used in the collection of relevant devices and systems in the HCE taxonomy, and for identifying important procedures in the system description.

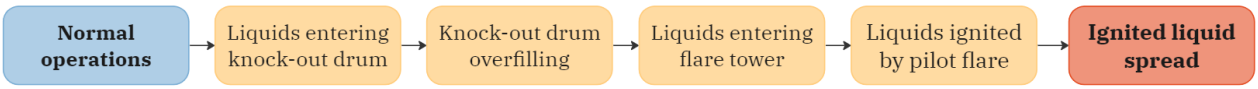


Figure 9: Preliminary HCE block diagram

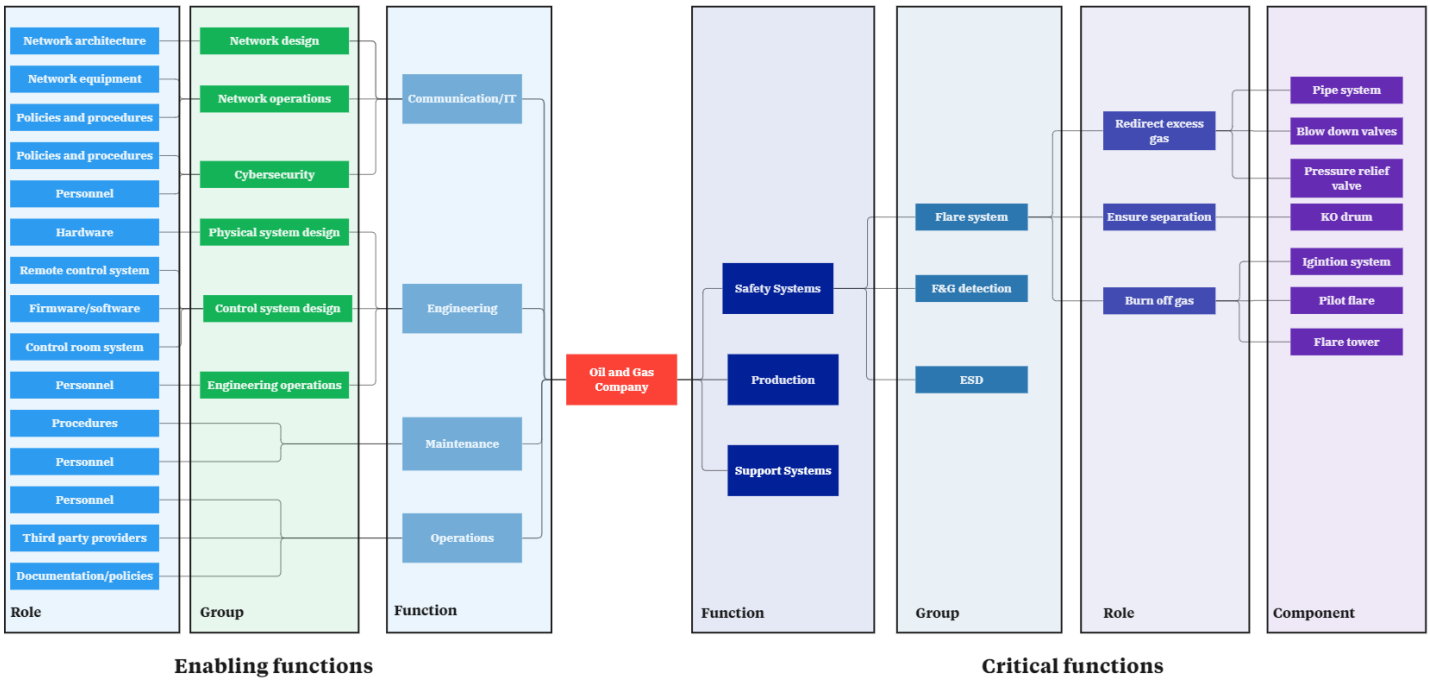


Figure 10: HCE taxonomy

The organization's taxonomy is built to better understand which components and functionalities are involved in the HCE. Figure 10 displays an executive view without the columns info object and terminal component. These columns are added in Figures 22-24, Appendix B. The taxonomy illustrates a fictional organization in the oil and gas industry. The figure is similar to the structure of a bow-tie diagram, with enabling functions on the left side and critical functions on the right. The enabling functions are used in the critical functions to obtain the organization's goals. Only relevant functions have been included on the critical functions side to avoid unnecessary information.

System Description: Operations

During an overpressure situation in the separator, the flare system is activated to safely dispose of the excess hydrocarbons and prevent the risk of explosion or equipment damage. The hydrocarbons can enter the knock-out drum through the PRV or the BDV, depending on the circumstances. Usually, the relief happens through the PRV as this is a mechanical and reliable solution. The BDV can be controlled from the control room and is used in

safety-related incidents like ESD. The operational flow can therefore be separated into the two following scenarios:

Relief through PRV

1. If sufficient pressure occurs in a tank or pipe, the spring of the PRV will be pushed to an open position and relieve pressure into the flare system. The opening is dynamical depending on the pressure.
2. The knock-out drum separates the liquids and gases, with the liquids being collected at the bottom and the gases leaving through the top.
3. The gas then flows through the flare header and reaches the continuously burning flare, where it is ignited and burned off.
4. If residual liquids are returned to the process using the return pumps.
5. If the amount of liquids accumulated in the tank is insufficient to initiate the return pumps, operators can increase the temperature of the liquids in the bottom of the tank to evaporate it and burn the gas through the flare.

Relief through BDV

1. ESD initiated.
2. The BDV is opened fully to empty the production line of pressure into the flare system.
3. The knock-out drum separates the liquids and gases, with the liquids being collected at the bottom and the gases leaving through the top.
4. The hydrocarbons then flow through the flare header and reach the continuously burning flare, where they are ignited and burned off.
5. Liquids separated from the knock-out drum are not sent back to the production line but sent out through a drain system or similar.

The temperature over the gas sensor will vary depending on the amount of gas flowing to the flare tower. An increased amount of gas will decrease the temperature. This means that in a blow-down scenario, where large amounts of gas flow through the flare system, the sensor will record a drop in temperature.

It is important to note that in an open system, the flare system will have a continuous flame as it will be burning off hydrocarbons at all times, even when there is no overpressure situation. The knock-out drum prevents liquids from reaching the flare, which could extinguish the flame and compromise system safety. The system has no additional seals, which makes the knock-out drum the last barrier to avoiding liquids in the flare.

System Description: Network and External Connections

The network topology is illustrated in Figure 7. Levels 0-3 are considered the OT network, and levels 4-5 are the enterprise network. Each facility has this entire stack and is interconnected with other facilities. Communications between levels 2-5 are Ethernet TCP/IP,

and levels 2-0 use Ethernet-APL running OPC-UA. SPANs copy communications between levels 0-3 to the IDS at level 3. Remote access into level 2 is possible through VPN to perform system programming or maintenance. The facility has remote access for third-party providers or vendors.

System Description: Components

All the inputs and outputs belonging to the subsystem are connected to one Siemens S7-1500 PLC. The PLC is communicating with one dashboard in the control room and one EWS. The EWS has the functionality to reprogram the PLC. The control room dashboard is a digital display with a fixed setup. The engineering laptops, EWS, file server and HMI, are all organization-standard Dell devices. Table 5 describes all relevant information about the system components.

System Description: Control Platform

The control room comprises programmable operator screens (also called human-machine interface) and a critical application panel (CAP). The programmable operator screen is the dashboard described in the system component description. The PLC provides the input and output. Of the states relevant to the flare system, the CAP displays activation of APS, ESD 1 and 2, and BD. If an alarm is triggered, the panel does not indicate the location or domain of the ongoing incident. It also has the ability to activate APS, ESD1, ESD2 and BD. The operator screen displays several states relevant to the flare system:

- Liquid level in knock-out drum
- Temperature of gas in knock-out drum
- Temperature of liquid in knock-out drum
- Pressure in knock-out drum
- State of BDV and PRV
- State of drain system valve
- State of return pumps
- State of valves going into the flare collection line

The flare status is monitored through CCTV cameras. This means the flare is observable, but there is no automatic alert or action in case of a flare-out.

5.3.3 Consequence-based Targeting

The consequence-based targeting uses the information gathered in the system-of-systems analysis to identify how an attacker would target the system to execute the HCE. The goal on the organization's side is to produce an attack scenario. This attack scenario identifies points or parts of the system the adversary must traverse, the choke points. To obtain the attack scenario, one must first create one or more technical approaches. These are

Table 5: System components

Engineer Laptop	Function	Data host and engineering functions
	Vendor	Dell
	Model	Precision 3630 Tower
	OS	x64, Windows 10
	Protocols	TCP/IP/Ethernet, SSH, SNMP, HTTPS, HTTP, Telnet
Engineering work station	Function	Data host and engineering functions
	Vendor	Dell
	Model	Precision 3630 Tower
	OS	x64, Windows 10
	Protocols	TCP/IP/Ethernet, SSH, SNMP, HTTPS, HTTP, Telnet, OPC-UA
Control room HMI	Function	Control room updates and simple system control
	Vendor	Dell
	Model	Precision 3630 Tower
	OS	x64, Windows 10
	Protocols	TCP/IP/Ethernet, SSH, SNMP, HTTPS, HTTP, Telnet
File server	Function	Data host
	Vendor	Dell
	Model	Precision 3630 Tower
	OS	x64, Windows 10
	Protocols	TCP/IP/Ethernet, SSH, SNMP, HTTPS, HTTP, Telnet
SPAN Switches	Function	Duplicate network communication
	Vendor	Cisco
	Model	IE-3200-8T2S-E
CCTV system	Function	Flare monitoring
	Vendor	Siemens
	Model	Surveillance Video Core
	Protocols	TCP/IP/Ethernet, SNMP, Modbus
Siemens PLC	Function	Flare system controller
	Vendor	Siemens
	Model	S7-150
	Protocols	OPC-UA

used in combination with the information collected in the system of systems analysis to extract a target description and critical needs. Techniques and tactics from the SANS ICS Kill Chain and Mitre, introduced in Section 3, has been used to develop the technical approach.

Development and Incident Mechanics

Analyzing the conditions necessary for an ignited liquid spread, one can back-trace how an attacker would approach the system to implement the attack. The approach is similar to reverse engineering, where the end-point is used to deduce possible causing events.

For a liquid flare spread to occur, one must create a large liquid overflow spilling out through the flare tower. It would be more circumstantial for an additional fire or explosion to occur, but such events have been reported, for example, at the Texas BP refinery incident[27]. It would depend highly on factors like wind strength and direction, facility layout and procedures. The goal is, therefore, to overflow the knock-out drum with liquids and utilize the continuously burning pilot flare to ignite the liquid hydrocarbon.

As described in system operations, the BDV is the only valve that does not have a mechanical function principle and can therefore be controlled by a PLC. The attacker must choose an approach similar to the "relief using BDV" without emptying the knock-out drum while avoid being detected by the detection system or operators. Given that the knock-out drum is sized normally within standard regulations it will take around four minutes to overfill it if there is a high amount of liquid in the system and all the inlets to the flare system is opened.

System Analysis for Targeting

The system analysis for targeting must include many of the same resources identified in the system-of-system analysis. Additionally, a system analysis for targeting must include knowledge of how to breach the system's perimeter and any vulnerabilities associated with the devices used.

It is natural for the adversary to assume that there is a remote access connection to the facility. Using techniques such as social engineering, one could identify the names of personnel with remote access authorization to the correct parts of the system. It could be some of the organization's process engineers or a vendor employee if they have access to the system. After singling out an employee with remote access, the adversary would have to gather intelligence on the employee's home network, security procedures and the device they use to access the remote connection.

Technical Approach

Information collected in the system analysis for targeting is used to create a technical approach containing specific steps and actions needed to attack the system. The presented technical approach is inspired by the approaches and techniques used in Stuxnet and Triton.

The entire operation includes several stages, much similar to what is presented in the ICS Kill Chain[41]. The initial stage is only briefly described in this report and does not go

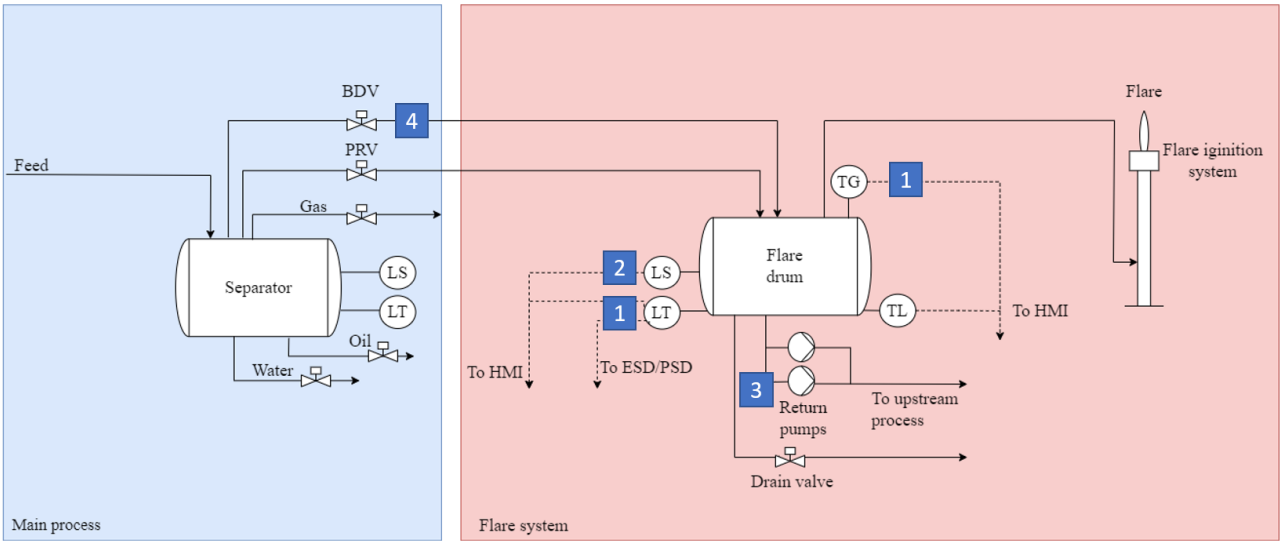


Figure 11: Technical approach - system drawing

into the details of the reconnaissance step. Two main targets are involved in executing the attack itself (stage 2 of the kill chain). The operation must attack the components controlling the physical parts of the flare system. It must also masquerade the attack from both the system alarms and the control room. Each technical approach should contain an overview of target access, required actions, and the timing for triggering and triggering factors. An overview of the devices included in the technical approach is given in Table 6.

Table 6: Technical approach

Target	**Access**	**Required actions**	**Timing/factor for triggering**
Engineering laptop	Compromised home Wi-Fi router access through remote access upon connection	Download malware for PLC and control room HMI	Immediately on connection to home Wi-Fi
EWS	Certificate-based authentication and VPN server configuration for remote access through OT firewall	The malware should reprogram the PLC controlling the flare system to open the BDV, disable the LAHH and return pumps and feed the LT, LS and TI with false readings	Immediately upon engineer laptop connection to OT network

Figure 12 illustrates how the operation can be executed by attacking the system's physical components while masquerading the adversary's actions. The technical approach event flow is similar to the HCE block diagram presented in the system of system analysis, but adds the required actions associated with the events. Figure 11 displays the same flow, but places it into the context of the case system drawing. The numbers corresponds to the same four stages as listed below. The chronological sequence of the attack is also listed in more detail below.

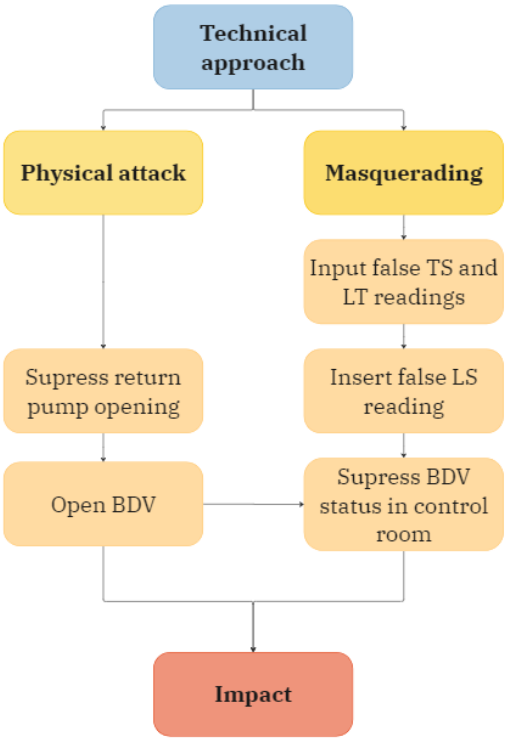


Figure 12: Technical approach - flow chart

1. Replay previously recorded tank level and temperature measurements.
2. Disable the level switch.
3. Ensure return pumps and drain system valve are closed and will not open.
4. Open all BDVs supplying the flare line.

It will be more beneficial to launch the attack when there are higher amounts of liquids present in the system. A larger liquid flow will cause the tank to overfill more rapidly. Therefore the attacker should gather intelligence on the procedures and operations of the entire facility.

Another technical approach not pursued further here would be remote accessing the EWS and reprogramming the PLC instantly. This removes all malware development and testing requirements but increases the chance of disclosing the operation.

Target Details and Critical Needs

The target detail describes what the adversary needs to know to design a successful attack on the system. This is a concentrated version of the system-of-systems analysis given in Section 5.3.2. The case study is limited to the extent that all the information gathered in that analysis is also considered essential for the attacker.

In order to access the system, the adversary must know the overall system structure. This knowledge is gathered during the reconnaissance phase of the attack. Schematic diagrams are generally stored on a file server in the OT or enterprise networks. One of

these file servers would have to be accessed depending on the approach chosen to breach system parameters. The attacker would specifically need to gather information about the following subjects:

- Schematic diagram of flare system
- PLC and EWS firmware and configuration
- Valve, pipe, flare and knock-out drum specifications
- Procedures and operations regarding ESD
- Procedures and operations regarding pressure relief
- Procedures and operations regarding remote access
- Network architecture diagram
- Personalia of employees or contractors with remote access

Some of these resources are not provided by case, e.g. specifications of valves and pipes. This is done for simplicity but would be crucial to design an attack that could successfully alter the system's behaviour.

Critical needs for deployment are requirements for the installation and execution of an attack. In the chosen technical approach, the malware installed through remote access is used to execute the attack. Therefore one critical need for deployment is access to the home network of an authorized engineer. As previously mentioned, the attack will be more effective if timed correctly. Another critical need is access to facility procedures and operations. The target details with critical needs for development and their location on target are presented in Table 7.

5.4 Detection Parameter Selection

The main objective of the performing the CCE assessment was to identify parameters that can be used to improve detection. Therefore, the mitigations and protections stage is not fully evaluated. Instead the focus is on the detect function, defined as "*Mitigating measures identifying malicious activity as fast as possible*"[6]. Specifically, the task is to extract system specification-based detection parameters. This section identifies such parameters that potentially can be used to detect the attack scenario.

Based on the technical approach and reasoning behind how an attack on a flare system must be conducted, several process parameters and data points can be suggested to detect the attack before it causes a significant impact.

- Mismatch between level and temperature - During the attack the tank level will increase. If the gas temperature sensor displays the same readings, it can indicate that the tank is filling up while gas is not flowing through.
- Repetitive patterns - A poorly designed masquerade might contain repetitive sensor reading patterns.

Table 7: Critical needs for development

Component	Critical Needs	Location
Engineer Home Router	Specifications/data sheet	Vendor
	Configuration file	On board
Engineer Laptop	Specifications/data sheet	Vendor
	VPN	On board
	OS	Open source
EWS	OS	Open source
	Specifications/data sheets	Vendor
	Flare system project file	On board/file server
	Software and install process	Purchase
File server	Schematics - Network topology	On board
	Schematics - Communication diagram	On board
	Documentation - ESD procedures	On board
	Documentation - Control room layout	On board
	Documentation - Remote access procedures	On board
	Documentation - Remote access authorization list	On board
	Documentation - Flare system procedures	On board
Siemens S7-1500 PLC	Product specs / manuals	Open source
	Configuration File	Organization file server
	Vendor I/O Module Pinouts/Wiring Diagrams	Vendor Website
	Control Logic Diagram	Organization file server

- PLC reprogramming - Reprogramming of PLCs is a known detection parameter for signature-based IDS[5]. The attack scenario discussed has PLC reprogramming as a critical need.
- BDV open while return pumps/drain are closed - This indicates that the knock-out drum is not being emptied.
- BDV open while PRV closed - This parameter would depend on knowing the ESD/PSD status, as the BDV should not be opened unless the situation is safety-critical.
- Sudden liquid drain in other parts of the system - This parameter would require monitoring of other parts of the system but should be recognized as an effective parameter. Sudden liquid drain during normal operations could indicate that the liquids are being redirected.

Monitoring the knock-out drum level is potentially one of the best indicators of a malfunction or ongoing cyber-attack. Looking back to Table 1, describing different hazards along with suggested mitigation. In this table, Denham and Donnelly points out that *"effective level measurement in the knock-out drum"*[22] would be a good mitigating measurement for hazards including liquid overfill and slugging.

The two latter parameters; BDV/PRV combination and sudden liquid drain - depend on monitoring parameters outside the flare system. They are included because they can be relevant for detecting the ongoing attack. BDV/PRV detection can be partially implemented. The sudden liquid drain would exclusively depend on external measurements, and is therefore not discussed further. Regarding the PLC reprogramming, existing IDS like Snort and Zeek can detect such activity[5] and it is not strictly related to the physical system state.

The technical approach uses masquerading to evade detection. Masquerading is one of the techniques mentioned by Mitre as common in ICS attacks[42]. Therefore, another possibility of detection attack would be monitoring data integrity. Utilizing the fact that communication is spanned out from several different levels, communication can be compared by the detection unit for deviations. This detection approach is widely applicable across attack scenarios if any form of masquerading technique is used. If, on the other hand, the attacker chooses a technical approach which maintains the data integrity, the approach would be less helpful. For this HCE to occur, quite drastic changes must be performed compared to the system's normal operations, so succeeding by only adjusting behaviour within the expected range would not accomplish the desired outcome.

6 Implementation and Testing

Section 6.1 starts by describing the proposed detection solution based on findings from the CCE analysis and suggested detection parameters from Section 5.4. This is followed by a description of how the suggested solution is implemented regarding OT and IDS theory. Section 6.2 presents the test setup, and scenarios and the produced datasets are described. The code base for both the suggested solution and the test environment is hosted on https://github.com/majasma/opc-ua-parser.

Section 5.4, identifies several process parameters and data points which will be affected by the attack approach. These parameters and data points are a part of the system specification. In such way, one can create a monitoring solution based on the knowledge gathered in the system of system analysis stage of the CCE assessment. Correlating the suggested monitoring solution to the CCE mitigations and protection stage, this would be a mitigating measure in the detection category (see Figure 4, Section 4).

6.1 Suggested Monitoring Solution

The goal is to implement the suggested monitoring solution using a specification-based detection approach, which is introduced in Section 3.4.2. The CCE analysis produced a system of system analysis and identified some process parameters that potentially can be used for detection. This correlates to the system specification and specification extraction phases of the specification-based development process.

The suggested solution monitors network traffic. This makes it a network-based monitoring solution. The suggested solution then compares the observed system behaviour to the expected behaviour in the implemented model. The comparison is done through exact pattern matching. Exact pattern matching is the technique most commonly used in signature-based detection. If abnormalities are detected, it should react to the event.

The suggested solution is a packet parser analyzing the flow and content of packets spanned out between the different Purdue levels. The packets are analyzed between levels 0 and 2, which means there is one set of packets between levels 0/1 and one set between levels 1/2 for each scenario provided by the test environment. This part first introduces how the parser is implemented to detect the features selected in Figure 5.4. After the implementation walk-through, design choices are explained and discussed.

6.1.1 OPC-UA Parser

The parser is implemented as an extension to the existing IDS/SIEM solution. A parser is a software component or program that analyzes and interprets structured data, and extracts relevant information or performs specific actions based on that data[63]. It is implemented using Python. IEC62443 recommends to only use passive detection in ICS[18]. Passive detection means that the detection system does not react to any detected malicious traffic. therefore the parser does only notify the overall IDS solution when abnormal traffic is detected. A notification is sent to the IDS if any irregularities are detected in traffic flow between levels or system states.

A parser implemented in an actual system would likely read live communication flows[47]. To ensure reproducability and consistency in implementation and testing the test environ-

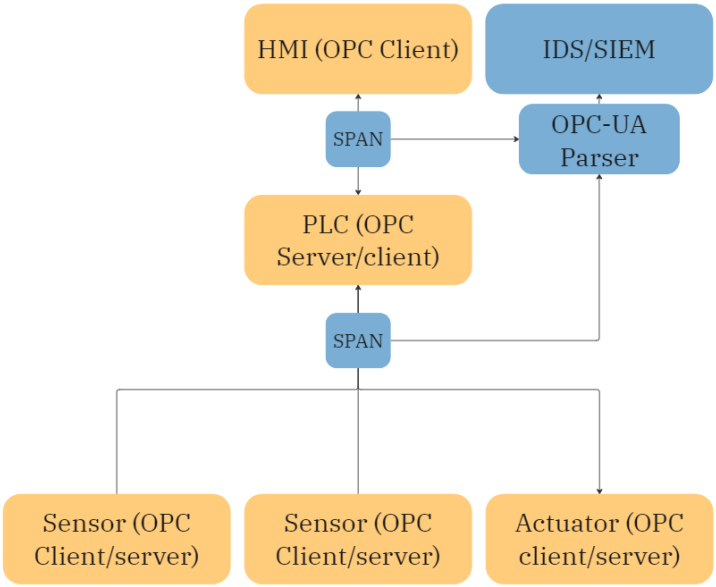


Figure 13: Suggested solution in case topology

ment produces a set of packet capture (pcap) files. The current implementation reads in the pcap files. Figure 13 illustrates where in the case network topology the parser would be placed. The yellow are devices providing system functionality. The blue are devices or systems associated with the monitoring strategy.

The parser monitors two aspects. The data integrity between levels, and the interpreted states towards the expected system behavoiur. The data integrity between levels is monitored by comparing packets sent between levels 0/1 to the ones sent between levels 1/2. A detection within this category is referred to as a packet mismatch. The other aspect is comparing the interpreted states towards the expected behaviour, this is the technique similat to specification-based detection. A detection within this category is referred to as a state irregularity.

The pcap files are read into the parser using the Python Pyshark package[64]. Pyshark is built on tshark, which is a part of the Wireshark network protocol analyzer suite. Wireshark is a well-known tool for network communication analysis and simplify future work and increases portability due to wide protocol support.

During development it became apparent that the OPC-UA packet does not have a unique ID, which made matching packets across pcap files challenging. If all packets are read into a data object sequentially, packets could be shifted into the wrong state. E.g. a temperature reading could be recorded as a level transmitter reading. The information about which state is being responded to is in the ”translate browse paths to node IDs request” packets. The state value is contained in the ”read responses” packets.

A filter is applied to filter out everything but the OPC-UA packets. The OPC-UA traffic is filtered further to only contain ”translate browse paths to node IDs requests” and ”read responses”. The state values are moved into a Pandas dataframe containing one column for each state and one row for each connection between server and client.

In case of packet loss, the translate browse path packets ensure that the correct states

are written to the correct columns. The parser expects the packets to appear in the same sequence as the servers were polled, and so if one poll cycle misses one or more state readings, a NaN is inserted, and a warning is written to the log. This process is performed on packets from each SPAN. A problem which remains unsolved in the implementation is the case where an entire poll cycle is missing.

The resulting dataframes containing all the OPC-UA traffic are analyzed for mismatches. Firstly all packets are counted for each connection between the client and server. If one dataframe is longer than the other, the longest is cut down; this is only to avoid errors during the processing of the files. This is a work-around for the unresolved problem mentioned above.

The result of the pre-processing is two pcap objects and two dataframes. The dataframes are compared, and if there is a mismatch, the state is logged, and a notification is sent to the decision-making unit. Further on, the dataframes are analyzed for state irregularities. These are the conditions presented in 5.4 which depend only on the flare system states. The condition "BDV open while PRV closed" is included as this can be matched towards the state of the ESD system in a simple manner. If any state irregularities are detected, a warning is written to the log containing the name of the irregular state. The state checks are:

- BDV open and PRV closed, check ESD status
- BDV open while RP/drain closed and level high
- Liquid filling but not large gas flow

As opposed to machine learning tools, the parser does not utilize any statistical methods. Instead, exact matching is used to perform all detection routines. As mentioned in Section 3.4.1, exact matching is often used in signature-based detection. The result is fewer false positives, which is required when working with safety-critical systems.

The parser implemented is a proof of concept. This means that the code is not implemented to be robust or optimized. A remark to be made is that exact matching can be time-consuming and lead to considerable delays or dropped packets. Time consumption is identified to be a potential challenge in future development.

6.2 Testing

A test setup is developed to test the effectiveness of the implemented solution. Test scenarios are developed by analyzing the expected operations from Section 5.3 and the chosen attack scenario to produce a communication dataset reflecting the processes' states. The section introduces the test setup and its reasoning before presenting the different test scenarios.

6.2.1 Test Setup

The case study uses OPC-UA as the communication protocol between the PLC and the control room. As described in Section 2.2.1, OPC-UA is both an architecture and a communication protocol. The test setup uses a client-server architecture. All devices can

act as both servers and clients based on the scenario. For the lowest level the test setup uses two servers initialized simultaneously to illustrate that the system might be using several PLCs. There is only one server in the level 1/2 communication.

The server is implemented using a Python script utilizing the opcua-asyncio package[65]. The Python script is preferred above a software simulator for simplicity and better control over test scenarios. The library utilizing asynchronous methods is chosen to simplify communication with the client. The server is first initialized as an object and registered on a URI. No authentication or encryption method is initialized. A node is added to the server, symbolizing the PLC itself. One variable is associated with the node for each in- or output of the PLC. Each variable is given a name, initial value and type. The initialization of each state is described in Table 8.

Table 8: Test setup state initialization

Name	Initial value	Data type	Unit
Level Transmitter	0	Float	% of capacity
Level Switch	0	Boolean	High(1)/low(0)
PRV	0	Float	% of opening
BDV	0	Boolean	On(1)/off(0)
Temperature Sensor - Liquid	14	Float	Degrees Celsius
Temperature Sensor - Gas	14	Float	Degrees Celsius
Drain System Valve	0	Float	On(1)/off(0)
Return Pumps	0	Boolean	On(1)/off(0)

The client is also implemented through a Python script using the asyncua package. The client opens an unencrypted connection to the server using the URI. All server states are polled with a 2-second interval representing standard system sampling. To illustrate the communication between layers 0 and 1, the client is polling states from two servers representing two groups of sensors. The servers in this scenario communicate on two different ports of the loopback interface.

The communication between the client and server is captured with Wireshark. The data is then exported to a pcapng file. The tests are conducted using the raw data to avoid missing potential false positives. The setup data extraction process is illustrated in Figure 14, where the full line would be extraction of Levels 1/2 communication and a combination of the full and dotted line would be Levels 0/1 communication.

6.2.2 Test Adaptations

The case is not based on an existing system, so the tests are all simulated. Simulations can introduce faults that affect the result of the tests. This section describes how the test setup deviates from a real-world setup.

The main difference is the architecture and topology. Realistically the server and client applications would run on different devices. During the tests, all are run on localhost. The OPC-UA applications do not interact with actual sensors or actuators, and the applications do not interact through a wired physical medium. In that way, most sources of delay and chances of data corruption have been removed. In a real-life setup, both delays and data corruption can occur for several reasons other than malicious activity.

The case study uses a SPAN switch to capture packets. SPANs copy the incoming traffic

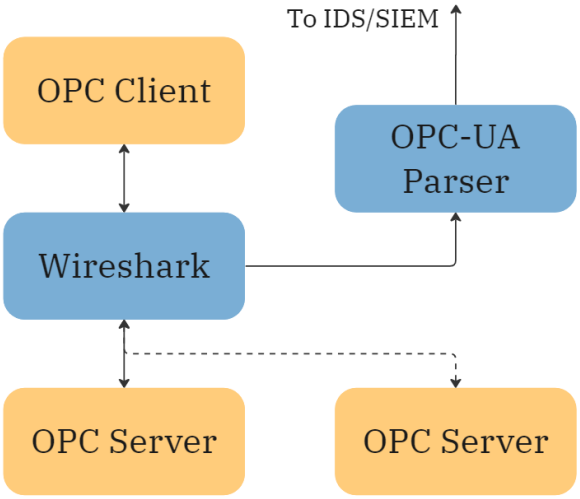


Figure 14: Test setup

and send it out in duplicate to the parser or IDS. As all switches and routers have a maximum capacity, the SPAN will drop packets if oversubscribed. This might not be critical for the system's functionality, but it will create discrepancies between the packet flows between the different levels and the historian.

6.2.3 Test Scenarios

The test scenarios are implemented as Python scripts used to produce a dataset of communication packets between the server and the client. The datasets are produced from three different scenarios reflecting the normal operations of the system and the attack scenario. Each script is implemented as pictured in Figures 15-19.

All the scenarios are implemented in a loop, where all states are calculated and set depending on the number of iterations passed and each other. Then the states are written by the server as well as stored in a separate file for later representation. All scenarios are terminated when the overall system state is once again neutral.

The system targeting in Section 5.3.3 mentions that overfilling a knock-out drum will take around four minutes. The scenarios are either ended when the system has returned to a normal state after a pressure relief or after 160 seconds ($2 * 80$ polls, with one poll every two second) in the attack scenario. 160 seconds is considered to be in the same order of magnitude as four minutes.

PRV Scenario

The PRV scenario comprises two parts, as Figure 15 illustrates. The figures containing sequence diagrams reflects how the sequence of code works, mainly how different events or transitions affect the system. The end case where the level switch is not activated, but liquids are manually evaporated is not included. One illustrates the system behaviour in the case of a considerable relief and one of a smaller relief. In the case of considerable relief, the PRV is modelled to have a dynamical opening between 40-55%. This causes

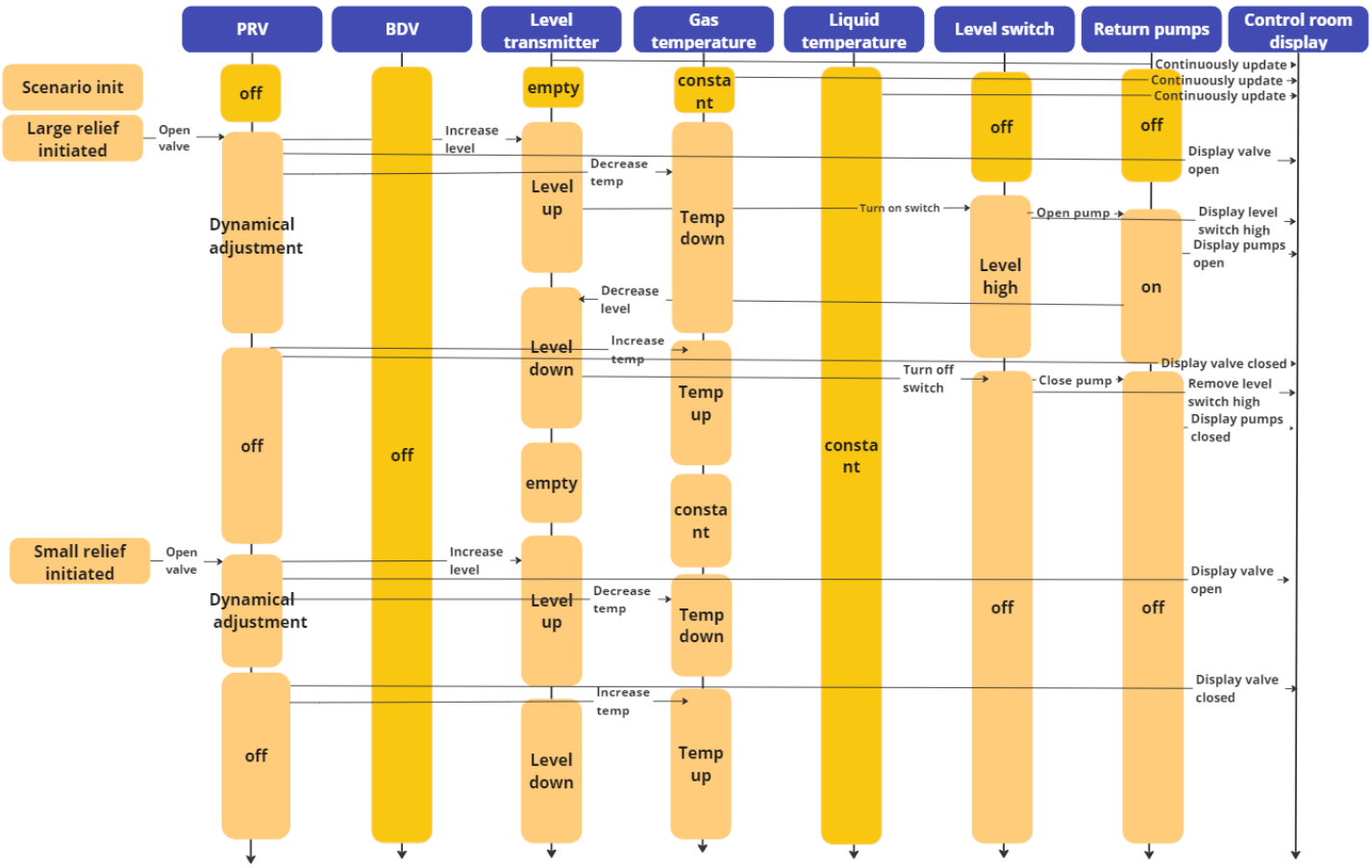


Figure 15: Sequence diagram - relief through PRV

the level transmitter readings to increase. When the level transmitter increases above the threshold, set to 10, the level switch turns on, and the return pumps open. After 180 seconds, the simulated large relief is over, and the PRV closes. The closing of the valve shuts off the gas flow. When the return pump is open, the tank level decreases. Once under 10, the level switch is turned off. The large relief scenario is terminated when the PRV is closed and the knock-out drum is empty (LT = 0).

The termination of the large relief scenario initiates the small relief scenario. The goal is to illustrate the case where the PRV is opened slightly, and the level of the tank is also barely increased above the level switch threshold. For 50 seconds, the PRV is opened in a dynamic interval between 3-5%. The level is only momentarily above 10 % of tank capacity, but the return pumps are activated to empty the tank. The scenario is terminated when the PRV is turned off and the tank is empty. A development of states over time is illustrated in Figure 16. The x-axis is number of poll cycles, and the y-axis is dependent on the unit described by the label in the upper right corner. States which are unaffected by the scenario is not included in the plot. Arrows denotes events according to the list below:

1. LS high/RP on
2. LS low
3. PR off

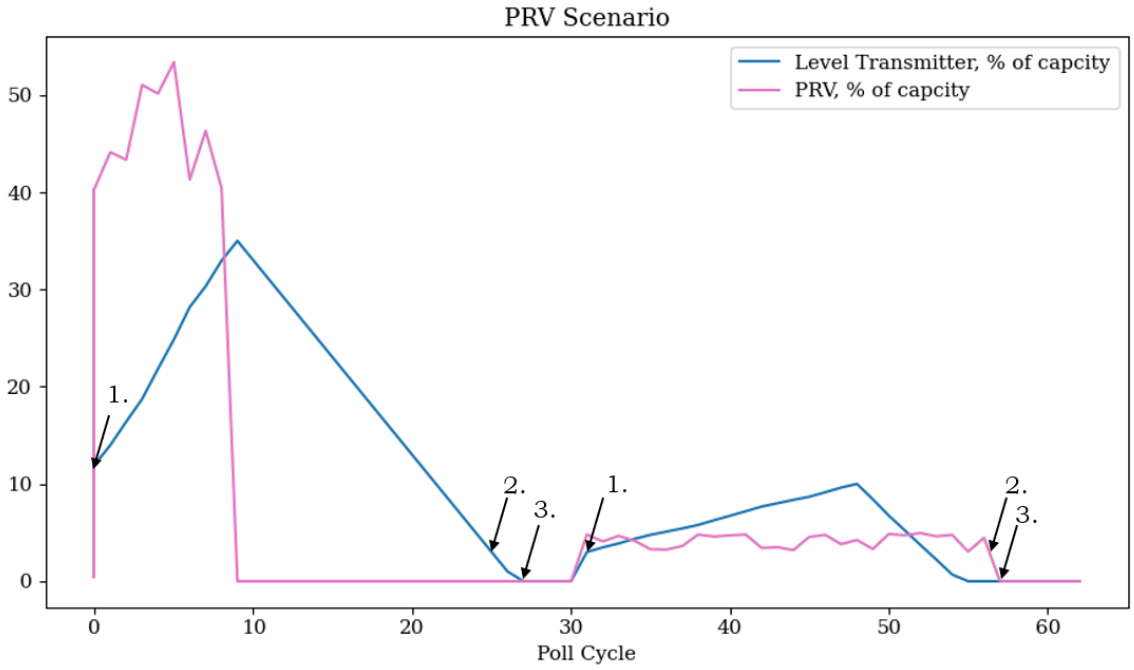


Figure 16: Plotted scenario - relief through PRV

The is one remark about all plots illustrating the state development (Figures 16, 18, 20 and 21). The level switch threshold is set to 10% of full tank capacity. From the figures it appears to be switched to high at around 5%. That is because the effect of the drain or return pump is written in the same poll cycle as the level switch is set to high. Meaning if there is a 5 % increase in tank level at the start of a poll cycle, increasing the total amount to 12 %, the drain or return pump will be turned on instantly to decrease the level making it appear lower than 10%.

BDV Scenario

The BDV scenario covers the process- or emergency-shutdown use case and is illustrated in Figure 17. The BDV is opened fully and releases gas and liquids into the system for 40 seconds. The large flow of gas and liquids decreases the temperature over the gas temperature sensor and increases the tank level. As the level increases above the threshold, the level switch is turned high. Given the prerequisite that the system performs an ESD/PSD, the drain system is turned on instead of the return pumps, like in the PRV scenario. This causes the level to decrease and the gas temperature to increase. The scenario is terminated when the BDV is turned off, and the tank is empty. A development of states over time is illustrated in Figure 16. The x-axis is number of poll cycles, and the y-axis is dependent on the unit described by the label in the lower right corner. States which are unaffected by the scenario is not included in the plot. Arrows denotes events

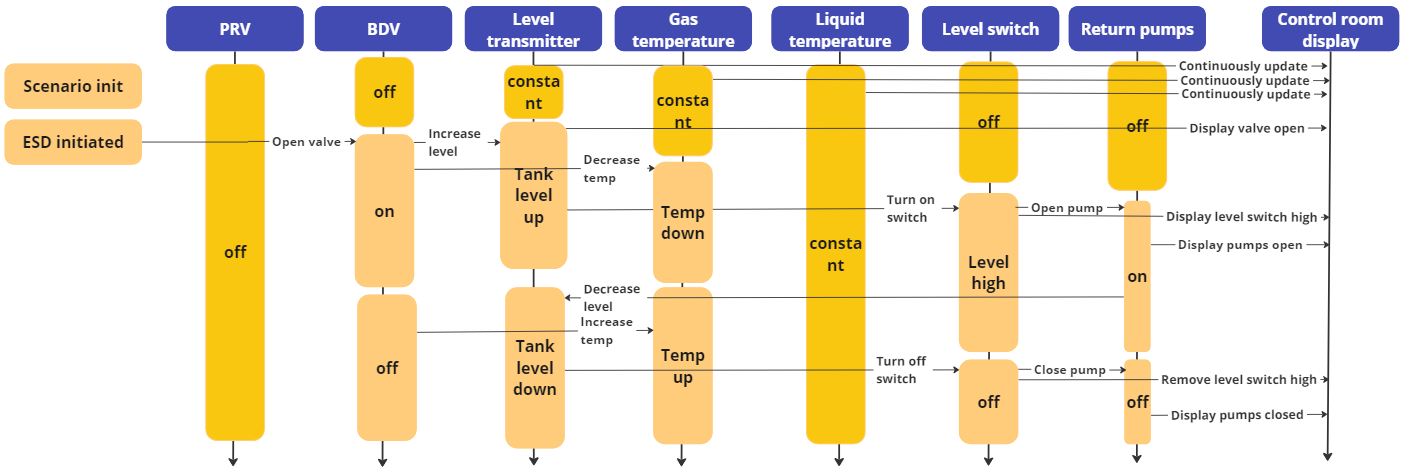


Figure 17: Sequence diagram - relief through BDV

according to the list below:

1. BDV open
2. LS high/drain on
3. LS low
4. Drain off
5. BDV close

Attack Scenario

Producing the datasets for the attack scenario is more complex. In the former examples, the communication between sensors and PLC will be identical to that between the PLC and the HMI. As the attacker seeks to mask the attack by replaying old process measurements, the communication between levels will not be identical. Two separate scripts are implemented to represent this. The sequence diagram describing the situation across levels is illustrated in Figure 19.

The actual attack is a simple scenario to model. First, BDV is set to open, rapidly increasing the tank level. Simultaneously, the temperature over the gas temperature sensor decreases due to the large gas flow. The level switch measures the high level, but the return pumps or drain system is not activated. When the level transmitter reaches 100% (full capacity), the value is continuously repeated as the sensor cannot measure the amount of overflow. The overflow is set to occur after polls, corresponding to 92 seconds. The state changes over different poll cycles is illustrated in Figure 21. States which are unaffected by the scenario is not included in the plot. Arrows denotes events which are named in the plot. This goes for both Figure 21 and 20.

The second is the communication received by the HMI - the masked traffic. As the increased gas flow will be visible on the CCTV camera system, it is favourable to pretend

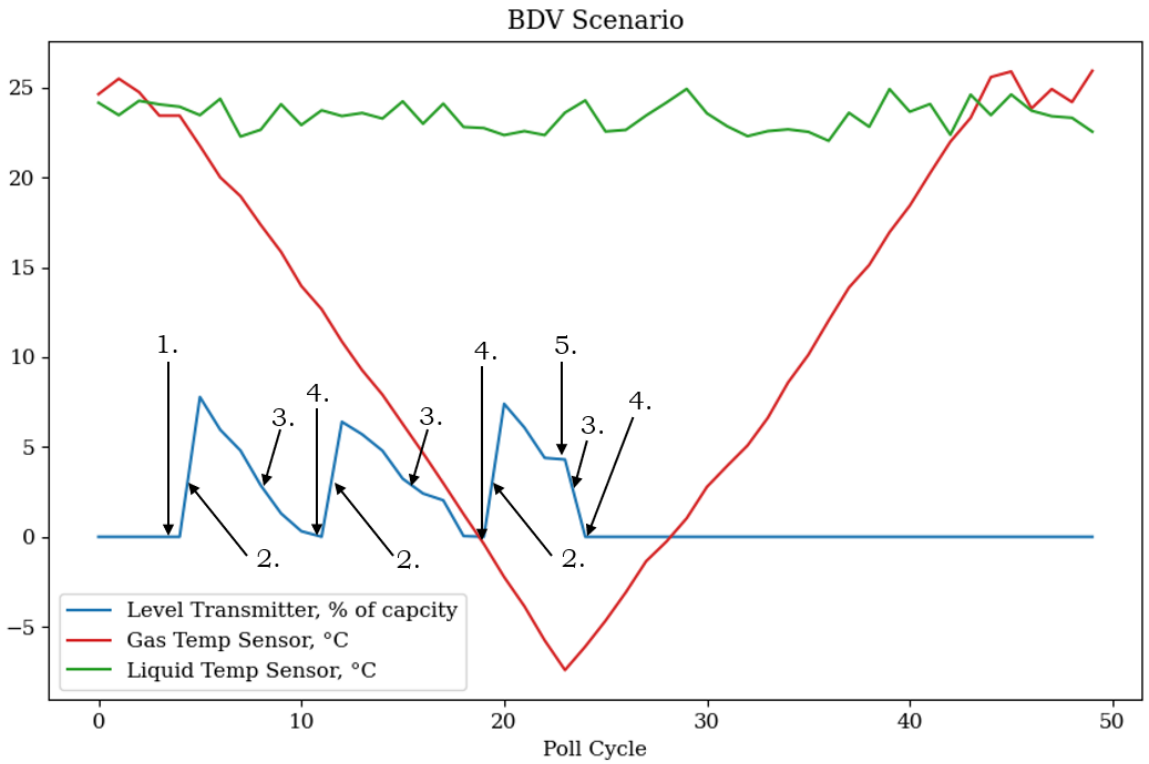


Figure 18: Plotted scenario - relief through BDV

a minor relief occurs. Therefore, the server writes values for the PRV and level transmitter according to a sine curve alternating between 0 and 2. Previously recorded records would have been used as a replay attack in an actual attack. After one minute, the values for the level switch are written high, and the return pump value is written true. The server then goes on writing as if the level in the tank was decreasing. The state changes over the different poll cycles is illustrated in Figure 20.

6.3 Test Dataset

The datasets are captured using the Wireshark software[66]. The following steps were performed to record the communication:

1. Access the interface for "Adapter for loopback traffic capture".
2. Start the client and server.
3. When the scenario is done, terminate the recording.
4. The capture is exported to a .pcapng file.

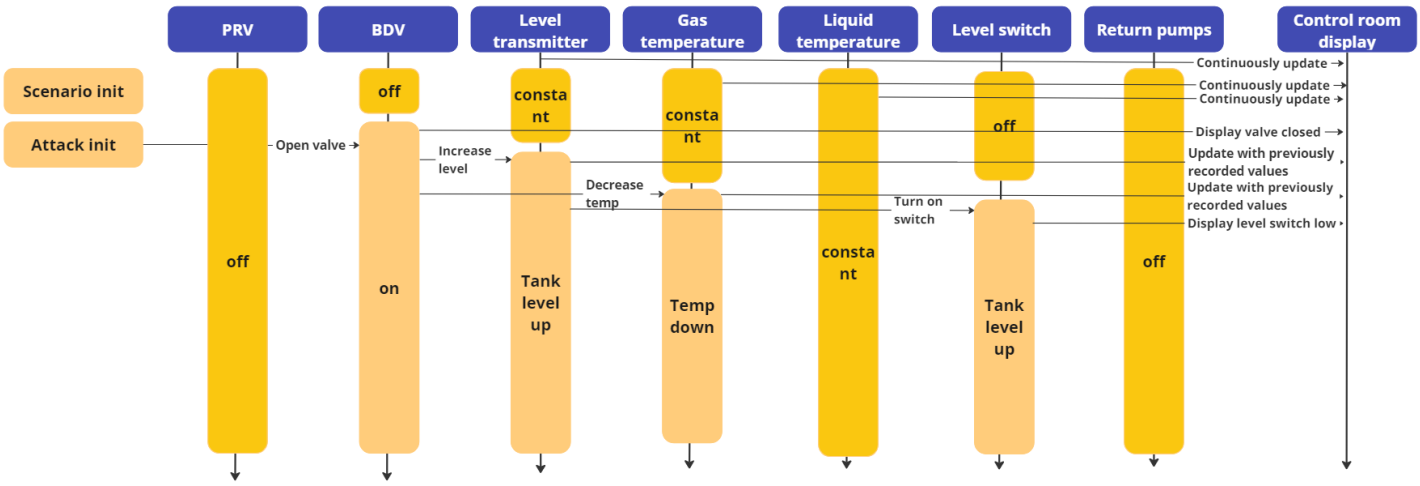


Figure 19: Sequence diagram - attack scenario

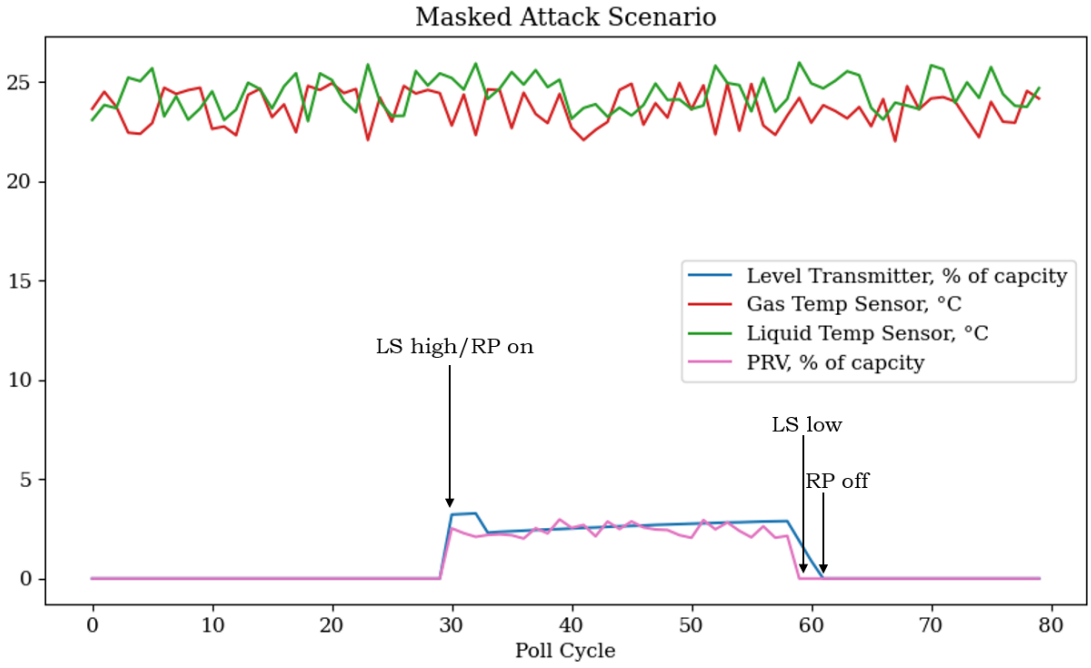


Figure 20: Plotted scenario - masked attack

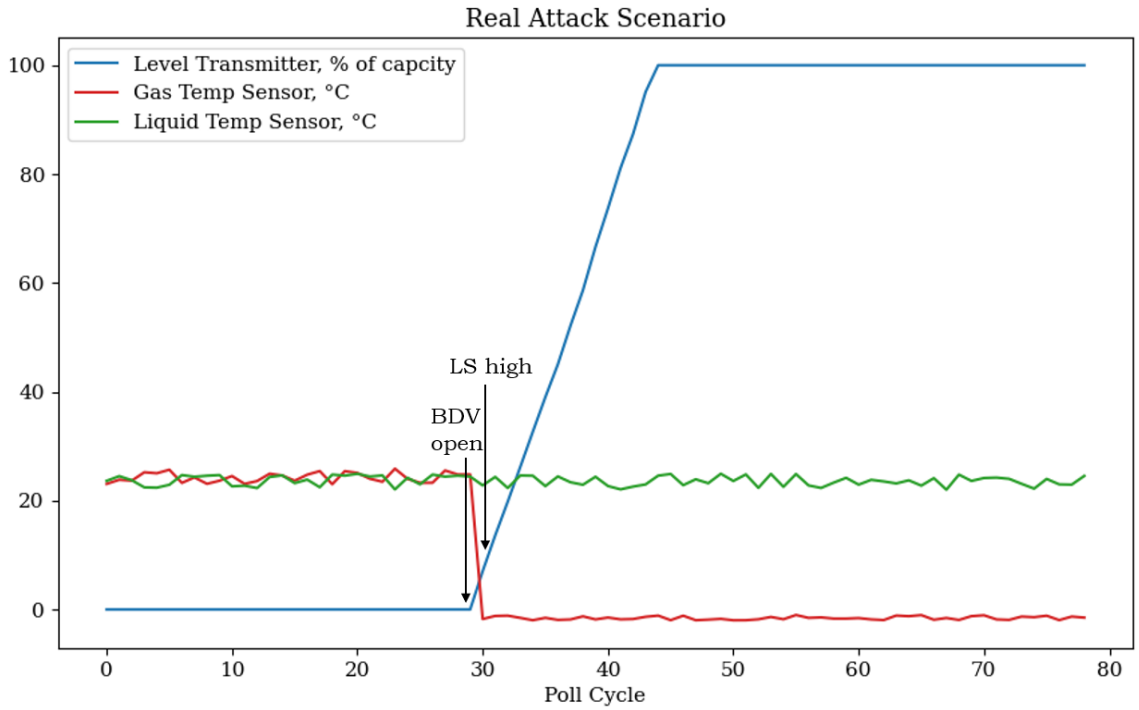


Figure 21: Plotted scenario - real attack

The raw dataset contains a repetitive communication pattern between the server(s) and the client in the format of the list below when the Wireshark 'opcua' filter is applied. The session is re-opened and -closed for every poll because of the client-server structure, which is not a permanent connection. A permanent subscription model might be more likely in stationary equipment like those in the flare system. The messages of type; read request/response and translate path request/response are repeated for each server variable every polling.

- Hello message
- Acknowledgment message
- Open Secure Channel Request
- Open Secure Channel Response
- Create Session Request
- Create Session Response
- Activate Session Request
- Activate Session Response
- Read Request
- Read Response
- Translate Browse Paths To Node IDs Request
- Translate Browse Paths To Node IDs Response

- Close Session Request
- Close Session Response
- Close Secure Channel Request

The data in the dataset are all sent on the loopback interface, meaning the source and destination addresses are the same. This would not have been the case in a real scenario. The address and port information is handled in the TCP/IP header and is not a part of the OPC-UA Binary Protocol.

7 Results

The results section presents the study's findings clearly and concisely, without interpretation or discussion. Each test scenario is presented with the results of packet analysis using the OPC-UA parser, and then the individual results are summarized by comparison.

PRV Scenario

From level 0/1, 488 read responses were collected, and 496 read responses from level 1/2. This corresponds to 61 and 62 poll cycles, respectively. 100 mismatches between levels were detected, which corresponds to 20.16%. The distribution of mismatches across states with its corresponding least detected difference, highest detected difference and average difference are presented in Table 9. The most frequent mismatch was the level transmitter state, with an average difference of 23.02 %. No irregular state readings were detected.

Table 9: Packet capture irregularities PRV scenario

Name	Count	Delta min	Delta max	Delta μ	Delta σ
Level Transmitter	38	3.82%	100%	23.02%	33
Return Pumps	3	100%	100%	100%	0
Level Switch	3	100%	100%	100%	0
BDV	0	-	-	-	-
PRV	16	2.34%	100%	24.11%	0.46
Drain Valve	0	-	-	-	-
Temperature liquids	20	0.25%	6.84%	2.92%	0.3
Temperature gas	20	0.21%	7.98%	3.14%	0.21

BDV Scenario

From level 0/1, 374 read responses were collected, and 400 read responses from level 1/2. This corresponds to 49 and 50 poll cycles, respectively. 12 mismatches between levels were detected, corresponding to 3% of the packets. The only mismatches were detected in temperature sensor readings. The distribution of mismatches with its corresponding least detected difference, highest detected difference and the average difference is presented in Table 9. The highest average delta is 5.39 %. One state irregularity was detected. From the moment the BDV is turned on, while the PRV remains open, the parser logs the "BDV open and PRV closed, check ESD status" event.

Attack Scenario

From level 0/1, 624 read responses were collected, and 640 packets from level 1/2. Corresponding to 78 and 80 poll cycles, respectively. 284 mismatches between levels were detected, which corresponds to 44.30 %. The distribution of mismatches across states with its corresponding least detected difference, highest detected difference and average difference are presented in Table 11. The most prominent discrepancy is in the level transmitter readings, which have an average difference of 2021.65%. The gas temperature sensor readings are also noticeable as they average a 104.27% difference compared to the liquid temperature sensor with an average of 3.88%.

Table 10: Packet capture irregularities BDV scenario

Name	Count	Delta min	Delta max	Delta μ	Delta σ
Level Transmitter	0	-	-	-	-
Return Pumps	0	-	-	-	-
Level Switch	0	-	-	-	-
BDV	0	-	-	-	-
PRV	0	-	-	-	-
Drain Valve	0	-	-	-	-
Temperature liquids	6	0.38%	9.96%	4.14%	0.12
Temperature gas	6	1.21%	9.73%	5.39%	1.02

According to the capture file, all state checks are detected at the correct timings. In other words, all of the following events were logged; "BDV open and PRV closed", "BDV open while RP/drain closed and level high", "Liquid filling but not large gas flow", and "Tank level exceeds 80%".

Table 11: Packet capture irregularities attack scenario

Name	Count	Delta min	Delta max	Delta μ	Delta σ
Level Transmitter	48	100%	11135.96%	2021.65%	182.86
Return Pumps	27	100%	100%	100%	0
Level Switch	47	100%	100%	100%	0
BDV	78	100%	100%	100%	0
PRV	28	100%	100%	100%	0
Drain Valve	0	-	-	-	-
Temperature liquids	8	1.26%	8.27%	3.88%	0.68
Temperature gas	48	2.40%	108.38%	104.27%	4.09

Runtime

The runtime of a program is the time spent executing all of the operations. The parser code is interpreted as it is implemented in Python. Interpreting code means that it is not a pre-compiled software running the tests. The runtime of a program is highly dependent on the structure of the program and the hardware it runs on. The runtime of the analysis of each test scenario was recorded and interpreted using Pythons built-in cProfile and pstats. Table 12 describes; the total number of OPC-UA packets recorded, the total runtime in seconds, the time used by the Pyshark "load_packets" function in seconds, and the time spent loading packets as a fraction of the total runtime. All other functions and subroutines are not included due to negligible time consumption.

Table 12: Parser runtime analysis

Scenario	Number of packets	Total time [s]	Packet load time [s]	Load time fraction [%]
PRV	1968	7.962	6.650	83.52%
BDV	1594	6.830	5.64	82.58%
Attack	2528	9.885	8.317	84.14%

8 Discussion

Before entering the discussion, the main objectives are revisited. The first objective was to "*identify critical data points and process parameters in a flare system using a CCE-inspired assessment method*". As the goal was to identify parameters that could improve detection, an adapted version of the CCE method was performed. It identified a worst-case event (HCE) and gathered all relevant data for the attack and how it could be conducted (attack scenario). This information was used to identify potential detection parameters.

The second main objective was to *"improve detection in flare systems using techniques inspired by specification-based detection and process parameters from the conducted assessment"*. Section 6 suggests a passive monitoring solution based on the outputs from the CCE method and chosen process parameters from Section 5.4.

This section assesses the method, development process, and results and evaluates it within the context of the research questions. The section critically analyses the test results and identifies any limitations. The discussion is separated into two parts, the first discussing the CCE method, how the assessment was conducted and which implications it had for the suggested solution. The second part discusses the results and how the test setup might impact the results. This separation invites the reader to take a step back and follow the process from the start of the project, where CCE was the main topic, over to the case study and then eventually discuss the results of the improved detection solution. All the numerical results from the OPC-UA parser tests are presented without interpretation in Section 7.

8.1 On the Usage of CCE

The discussion points about the CCE method and how it is used to research project objectives can be parted into three topics. First, the CCE method is compared to risk assessment methods and how the usage of either could have affected the results. Then, the potential relationship between CCE and threat actors is discussed. Lastly, the experiences from conducting the CCE method on a flare system case study are presented, and how the results from this analysis contributed to shaping the suggested monitoring solution from Section 6.

8.1.1 CCE and Risk Assessment Methods

The common factor between consequence-based and risk assessment methods is the goal of implementing mitigating measures or strategies to protect a system or organization based on the risk. The consequence-based methods differs from risk assessment methods by focusing on a set of worst-case scenarios and how they potentially could occur, ignoring the likelihood of occurrence.

Ease of attack deployment and little requirement for specific system knowledge would increase the number of potential attackers able to successfully launch an attack. In turn, this increases the likelihood of such attacks. These characteristics are often seen in attacks launched towards the higher levels of the organizational topology. The attacks targeting devices or subsystems operating on the lowest levels of the Purdue model[41] require much time, knowledge, and resources. They are therefore considered to be harder to succeed with. These attacks will typically be less likely than e.g. ransomware or denial of service

attacks. To cause a physical impact on the environment, it is these systems that mainly have to be targeted.

Major accidents will occur eventually, even if the likelihood is low. The argument above can be backed up by, for example, the law of large numbers. The law of large numbers is a theorem that describes the outcome of performing the same experiment many times. As the number of trials increases, the likelihood of a result occurring also increases. Given the knowledge that major events can occur, and that major events have high consequences, safety measures mitigating such events should be implemented.

By performing a risk assessment including likelihoods, the results might not indicate that monitoring system parameters on the lowest part of the system should be one of the focus areas. This conclusion is almost unavoidable when using a consequence-based method, as the physical components of critical systems have the greatest potential to affect the environment. Leaving out the probability factor appears plausible for this case study.

Trisis and Stuxnet are examples of malware targeting devices at level 1 of the Purdue model. Succeeding with such an attack is extremely difficult according to the SANS ICS Kill Chain[41]. Usage of these malware compose a negligible amount of all attacks towards organizations operating within the utility and energy industry. As a result, the likelihood of such attacks is minor compared to attacks like ransomware or DoS. Omitting the implementation of security strategies for physical subsystems is unwise as the outcome can be similar to the incidents described in Section 2.4.1. Therefore, using CCE to identify detection parameters for critical components to use in detection routines is a suitable choice.

The statement above does, however, come with reservations when discussing entire organizations. A CCE assessment is ineffective as the sole framework for developing protection measurements and -strategies. Every organization should know the threat actors and -landscape for their industry. As some attack approaches are more common than others, one could expect a more rapid development in complexity for the respective tools and malware. An organization should be able to protect themselves against such attacks to avoid reputational and financial losses. The CCE assessment would be of best use as a supplement to the existing risk assessment framework.

8.1.2 Consequence-based Methods and Threat Actors

Threat actors are a consideration not included in the CCE assessment. In this study, state nations, criminal organizations, and insiders were considered the most probable to conduct an attack on the system. All of these are resourceful, motivated and do, at least to some degree, inherit the necessary knowledge. Assessing which actors are most likely to attack a system might help narrow down possible technical approaches based on adversary behavioral models. On the other hand, by not focusing on specific threat actors, mitigating measures do not pose the same risk of becoming biased in favour of the most likely attacker.

The CCE method does incorporate some considerations of attacker behaviour through system targeting. However, a behavioral assessment is dynamic and might change as new tactics or attack approaches are discovered. If adversaries are rapidly advancing in knowledge or resources, the system targeting process might need to be reevaluated. Keeping track of the threat landscape may not be directly linked to the CCE method but can indicate when it is relevant to perform a reassessment.

8.1.3 Flare System CCE Assessment

Cost-effectiveness has been a focus throughout the study. This focus, in addition to the study being limited to a flare system, does have some implications. The monitoring is limited to the flare system. As mentioned in Section 2.1, operational technology environments contain many systems, several of which are safety instrumented systems. These systems and others that could lead to high-consequence events should also be modeled. Which systems to assess and to what extent would be a decision the organization should make before initiating their assessment process.

Conducting the assessment proved more challenging than anticipated during planning. The main contributors to this fact were; sparse related work, lack of a well-defined environment in the case study, and lack of diverse competence. Firstly, few available resources at the time of literature collection performed a full CCE assessment[7][8]. All the resources are published by INL, which also has developed the method, meaning no independent assessments have been included. Assessing a subsystem and not an entire company still demands much information about the overall structure and operations, as proved in the system-of-system analysis in Section 5.3. This stage could have been less time-consuming using an exhaustively defined subsystem or case study.

The lack of diverse competence is somewhat connected to the extend of use case definition. The assessment required knowledge about several disciplines, mainly automation and cybersecurity, but also networking, embedded computing, and behavioural psychology. The CCE reference document[6] recommends using a diverse group knowledge-wise while conducting the assessment, which is an essential aspect. This is one of the main reasons the research was limited to one safety-critical system. Completing the assessment and expanding the case study singlehandedly leaves room for errors and subjective opinions or decisions. After receiving feedback on the realism of obtaining specific scenarios and effects, multiple iterations between stages, consequence prioritization, and system targeting had to be done.

The most useful stage of the assessment was system targeting. It was mainly this part used for the implementation of detection routines. The system of system analysis provided the information used to implement the system behavioral model, but this information could have been extracted directly from the case and system description. On the other hand, it did speed up the modeling process by providing the information in a structured format. Given a more extensive system, the first stage would be more useful, as the possible HCE could reside in different parts of the system.

The implemented solution would have likely detected several of the HCEs considered during the consequence prioritization for the flare system. If an organization inhibits either a consequence- or risk assessment, it could be reused to quickly identify potential bottlenecks or critical points. It would introduce the possibility of only conducting the system targeting and mitigations stages.

Considering time consumption and the system targeting being the most valuable component, one could contemplate whether applying a less time-consuming method would be more effective. Take the bow-tie technique for comparison. As mentioned in Section 3, the bow-tie technique is a visual assessment method where one places an event or fault in center, with causes on the left and consequences on the right[34]. Comparing it to the CCE method, the left-hand side would be the technical approach and system targeting, the HCE the event in the center, and the right-hand side would be the mitigating measures and strategies.

The downside of using a simpler method is that it provides less nuance and details. Especially during the selection of the HCE, the CCE method offers a much more structured approach to event evaluation. Event evaluation is not a part of the bow-tie technique and would require the organization to have an opinion of which events should be assessed. Nor does the system-of-system analysis have an equal to the bow-tie method. This might result in insufficient protection of critical devices if the organization does not have a good overview of its system and components. Looking back at the Dragos report stating that 76% of all oil and gas companies have little visibility into their OT systems, one could assume that the previous statement is unrealistic. Therefore, the CCE method is a more fool-proof and structured assessment, even if it is more time-consuming than its alternatives.

8.2 On the OPC-UA Parser Performance

By analyzing the results from Section 7, this section discusses whether the research's second main objective has been obtained. The second main objective was to "improve detection in operational technology using techniques from specification- based detection and process parameters identified in the conducted assessment". Simultaneously, unanswered questions or potential problems for future work should identified.

This section starts by looking at the overall functionality of the parser and how it can be used. Then Section 8.2.2 takes a closer look into the numerical results to determine the parser's performance. As the test environment and dataset are produced by this study, a subsection is dedicated to assessing the setup's quality and identifying potential issues.

8.2.1 Parser Applications

A parser is chosen as the suggested monitoring solution for the benefit of a more holistic solution. It is used as an extension to the existing IDS solution to eliminate the need for reconfiguration. In that manner, the IDS or SIEM remains the centralized decision unit. That way, function of the IDS will not have to be specializing in specific systems or protocols but instead optimizing the decision-making process. Adding service or functionality increases the complexity of security solutions. Increased complexity is a drawback usually best avoided, but in this case, the better option from a cost-benefit perspective.

Unexpected states are detected by analyzing the state updates sent between levels and comparing them with the expected system behaviour. The parser does not include any model for attacker behaviour or variable assessing the likelihood of an attack. Therefore, the parser is blind to whether it is notifying of an attack. Such a variable or model must be added in future work. It could be implemented in the parser or the IDS/SIEM logic. Alternatively, operators must perform this work manually, but it would require sufficient knowledge within the respective system and networks to determine if there is an ongoing attack.

Revisiting the mechanic of the Stuxnet and Triton malware from Section 3.3.1. Stuxnet utilized legal commands between the PLC and the centrifuges to obtain abnormal behaviour aiming to destroy the centrifuges. Even if the commands were valid within the protocol, the communication was masked so the operators would not react to the abnormal inputs. The Triton malware aimed to disable safety instrumented systems to either cause a process shutdown or send the system into an unsafe physical state exploiting a zero-day vulnerability. Triton also used masquerading to avoid detection. The masking of activity

and system measurement is common for both attacks. If the facilities struck by these malware had a tool for ensuring data integrity across levels and detecting inconsistencies, they might have been able to detect the attacks sooner.

In the academic literature, there is a distinct focus on detecting zero-day attacks[5]. This is, on the other hand, not considered a focus point by Equinor. As previously mentioned, many organizations utilize unpatched or even legacy devices. Unpatched devices expose an organization to more significant risks of attacks using existing tactics and malware. It also proves that tools that mitigate the risk of being attacked by malware that has existed for several years, like Triton and Stuxnet, are still highly relevant. That is also why this study omits further discussions on the mitigation of zero-day attacks.

The Eldfisk incident[26] discussed in Section 2.4.1 is one example of an incident where the entire platform was shut down because of fault indication, but the cause or location was not certain. Facility-wide shutdowns are rare, but it creates significant changes in pressure and temperature, which increases the probability of malfunctions and spills. A comparison can also be made to the Hydro cyber incident discussed in Section 3.3.1, where the uncertainty regarding the attack extent lead to partial shutdown and large financial losses. Precautionary shutdowns can also happen if a facility loses observability on systems or devices below level 2 of the Purdue model. Worst case, poor situational awareness can lead to fatalities, like in the Texas BP incident[27].

By having full observability and an interpretation of critical system states, one can avoid precautionary system shutdowns caused by poor situational awareness. As long as the device on level 0 or 1 functions and posts state updates that the parser receives, the parser can function as a redundant observer. Both in the case of a non-malicious incident like Eldfisk or a malicious incident where critical systems may not be infected like at Hydro. The parser can improve the detection of non-malicious system failures and system health monitoring. One of the main findings is therefore that the parser can function both as an integrity monitor and supplement for the intrusion detection system.

There are also some challenges discovered during the design process. The parser depends on the ability to retrieve and parse communications between levels 0 and 1. If OPC-UA communication is not implemented on these levels, one would need another parser to interpret the existing communication protocol. When the states are extracted, states could be compared in the same manner as in the parser.

An attacker could potentially access communication between levels 0 and 1 before the communication is spanned out to the parser. This approach would be a man-in-the-middle (MitM) attack. In a MitM attack, the attacker intercepts and relays communication between two parties, allowing them to eavesdrop, manipulate, or impersonate legitimate participants without their knowledge. In such case, the parser would be oblivious to the manipulation. One suggestion to tackle this problem is to extract the state updates directly from the OPC-UA server or the sensor. The suggestion is undesirable as it requires separate cabling or alteration of existing device configuration. As discussed in Section 3.4.3, required alterations to the existing setup and external resource usage is an IDS performance metric that ideally should be minimized.

8.2.2 Numerical Test Results

Moving on to the recorded results presented in Section 7, one of the first noticeable points is that all the datasets produced to represent level 0/1 traffic are missing one or two poll

cycles. This is likely because the servers are initialized sequentially and run on relative poll timings rather than fixed timings. It illustrates a point that should have been considered during the design of the parser. The fact that a system is likely to operate with multiple servers can cause inconsistencies or drift in timing, causing the client to receive old measurements.

Regarding packet mismatch detection, several different conclusions and points can be made. For the states with boolean values, the information about the measured difference, mean, and variance provides little insight. For all cases, the difference is 100%, and the variance is zero. This is caused by the fact that the dataset only contains 0 and 1. A mismatch would occur if a 0 is expected to be a 1 and vice versa. The difference between the two values will always be 100%.

In the case of the BDV scenario, the 3% mismatch rate is to be expected, as seen in Table 10. The PRV case, on the other hand, displays a mismatch rate of 23.02% (Table 9). There are two likely reasons for this. The first possible reason is a topological consideration not considered during design and implementation. In a client-server OPC-UA architecture, the packets do not have any unique identification, meaning no message ID or timestamp. It is challenging to confidently match packets captured on different levels without knowing a synced point in communications. Packets would have to be synced or marked with an identifier to eliminate this source of errors.

State mismatches between temperature sensor readings are present in all the scenarios. For all the temperature sensor states except the gas temperature in the attack scenario, the mean delta is lower than 6%. Given that the system has an expected operating temperature of 20-30°C, 6% of 25°C corresponds to $\pm$ 1.5°C. The temperature variance is considered well within reason, as temperature readings can be affected by several external sources.

The delta columns in the matched state values provide information about the recorded mismatch's significance, and the variance tells how large the spread of the values is. Even if there are frequent mismatches in the scenarios for normal operation modes, the average size of the mismatch is lower than in the attack scenario. For example, the average difference in Level transmitter recordings for the PRV case was 23.02% (Table 9), while in the attack scenario, it was 2021.65% (Table 11). The same goes for the gas temperature sensor readings, respectively 3.13%, 5.39%, and 104.27% for the PRV, BDV, and attack scenarios.

In other words, the relative amount of mismatched packets and the average recorded difference in the level transmitter and gas temperature sensor can be helpful in the development of more precise signatures. A concise example of this would be to have the parser analyze the size of the recorded mismatch over a period of time. If mismatched packets exceed 30%, it could indicate an attempted masquerade. If the recorded delta exceeds 50% for the level transmitter or 20 % for the gas temperature sensor, it could indicate malicious activity or a malfunction in communications between levels 1 and 2.

The count of mismatches in the attack scenario over double the amount for the PRV case, which is significant even if the test execution could have affected the results. If non-boolean values had been used to represent the remaining states, these might also have proved useful in detecting irregular traffic.

The test scenarios' design could also cause the unbalance in mismatches between state parameters using float and boolean values. There are fewer state changes in the BDV

scenario, as seen by comparing Figures 17 and 16. The BDV scenario uses more boolean values, which would not appear mismatched even if two packets with a state reading "1" are compared. This corresponds to a false negative, where the parser does not detect a mismatch which is actually there. Conversely, the state of the temperature changes more frequently and the mismatch is therefore easier to catch. One way to tackle the effect could be to implement some memory into the parser. The memory could allow the parser to only notify about a mismatch if the value deviates by some predetermined factor. E.g. for the temperature sensor readings, one could allow a deviation of 5% or some other reasonable value based on historical recordings.

The PRV test did not exhibit any proof of state irregularities. This was expected, as the test represents the normal operation of the flare system. The BDV test indicated a positive for the "BDV open/PRV closed" combination. This was expected as the BDV is open while the PRV is closed in an ESD scenario. This detection signature would require knowledge about the state of the ESD system to be helpful. Nevertheless, the signature was included because the ESD case should occur rarely, and any occurring false positive could indicate a faulty system component. All the expected signatures were indicated during the attack scenario, meaning the parser can successfully detect state irregularities based on the attack scenario developed in the CCE assessment.

One remark to be made about the parser logging of irregular state behaviour is that every irregular state is logged every poll cycle. This means that if the level is above 80% of tank capacity for X number of poll cycles, this will be logged X times. The issue might be solved more elegantly by e.g. logging the first instance and then logging only every tenth consecutive instance to avoid notification overflow.

Per now, all detections are logged as a file format. In future work, this should be implemented as a messaging or notification structure where the parser communicates notifications or alarms directly to the decision-making IDS. During that stage of development, it would also be natural to add identification codes or prioritization to the different signatures.

It would also be an advantage to implement more of the techniques used in specification-based detection. The prioritization should be based on the likelihood of an attack is occurring or if a routine change or malfunction likely causes the detection. Like all specific numerical values, this should be based on device configurations and historical data. During this study, all values have been set given the best knowledge, which would not be sufficient in a real system.

A concern during the testing was that the exact pattern matching of states and comparison of packets could prove time-consuming. Time spent matching packets toward signatures is known to be a possible problem in signature-based detection, as mentioned in Section 3.4.1. As displayed in Table 12, it is not the pattern-matching consuming the majority of the time. The time spent loading packets makes out more than 80% of the runtime for all cases. As mentioned in Section 6, a real parser would not load packets but read communication directly and consume considerably less time in total.

Another remark about runtime is that the packet capture file contains packets for intervals of several minutes, while the runtime ranges between 6.8 and 9.8 seconds. This means the parser is not currently using too much time to keep up with the scenarios even if the load_packet routine is included.

8.2.3 Test Environment and Dataset Considerations

There is a possibility of errors as the suggested solution, test setup, and dataset are produced by the same study. The possibility of missing important aspects or lack of nuances is high. The ideal scenario would be to receive a dataset from a real system and instead customize the implementation based on differences in configurations, as the solution does not rely on any specific devices other than knowledge of sensor-, ip-, and port configuration.

When implementing a detection tool, signatures or thresholds for notification should be tuned according to expected system behaviour. E.g. if latency is a known issue, and does not normally indicate malfunction or malicious traffic, the detection solution should take this into consideration. Accuracy and low false positive rate is important IDS metrics, as mentioned in Section 3.4.3. The detection solution should be tested on a representative dataset to eliminate false positives.

The dataset contains in total of 6044 OPC-UA packets. These packets correspond to around 755 poll cycles, so all the system states are updated 755 times. The packets are distributed across the three test scenarios. The attack scenario does not reflect behaviour that can be expected in everyday operations. As mentioned in Section 6.1, the parser does not go through a learning or training process. Therefore, there is no required dataset size to obtain reliable results. According to Wujek et al. [67], using machine learning methods on small datasets can lead to insufficient accuracy. Training a machine learning model on a dataset where rare events have a disproportionately large part of the dataset can lead to biasing. Bias in machine learning refers to the systematic error or unfairness that can occur when algorithms make decisions or predictions that favor certain classes or outcomes[67]. The dataset is, in other words, not suitable for the development of machine learning models. This remark is irrelevant to answering the research question but is included for future development.

9 Conclusion

The project has identified critical data points and process parameters in operational technology using a CCE-inspired assessment method. It has also suggested improving detection using specification-based detection techniques and process parameters identified in the CCE assessment. These were the project's main objectives. More specifically, the study presents an OPC-UA parser as a cost-effective supplementary tool to increase detection with limited resources using specification-based detection parameters.

The comparison between consequence- and risk-based assessment methods highlights the difference in their approaches, with consequence-based methods focusing on worst-case scenarios without considering likelihoods. CCE is such a consequence-based method. Attacks inflicting physical damage are often considered to be less likely, and can therefore be neglected in a risk assessment. An approach leaving out likelihood is particularly relevant for critical infrastructure systems, where preventing physical damage is essential. It makes the CCE method suitable for identifying critical system parameters for specification-based monitoring. However, the CCE assessment should be used in conjunction with existing risk assessment frameworks to address the entire threat landscape of an organization.

The assessment process proved more challenging than anticipated due to the need for more available resources, the need for a well-defined case study, and the requirement for diverse competencies. Despite these challenges, the assessment successfully identified an HCE and potential detection parameters. The most useful stage of the assessment was system targeting, which could be reused in a more extensive system to quickly identify potential bottlenecks or critical points. The CCE method could have been simplified to resemble a bow-tie assessment, but this would be a trade-off between nuance and simplicity.

This thesis introduces OPC-UA parser, which can function as a supplement to existing monitoring solutions and as a monitor for data integrity and system health. It can detect unexpected states by analyzing and comparing state updates with expected system behavior. It improves system integrity confidence in normal operations if combined with TCP/IP-based communication between levels 0 and 1. Situational awareness is raised during incidents, which may save the organization precautionary shutdowns.

The parser tests revealed some limitations and potential sources of errors, including the possibility of inconsistent timing between servers and the lack of unique packet identification. The assumption of OPC-UA communication between levels 0 and 2 is a weakness in portability. The parser could be vulnerable to MitM-attacks. These limitations and challenges should be considered if the parser is to be developed further.

The test setup is likely to be the primary source of errors during the test execution. Testing parsers or IDS supplements on a real dataset would be crucial to eliminate unintended errors and model the system properly.

The study suggests that the overall facility security and system integrity confidence can be improved by including specific system parameters in the monitoring solution. Improved system integrity confidence also extends to monitoring and detecting unintentional system failures. The CCE assessment method has proved to be valuable in the process of identifying these system parameters.

9.1 Further Work

Future work could be divided into short-, mid-, and long-term. In the short term, the parser test framework should be improved. Improvements should include timing synchronization and adding unique packet identification. This would improve the accuracy and reliability of the test results and enable a more detailed analysis of the communication patterns of the system. Performing tests on a non-synthetically produced dataset would also be valuable to confirm that a parser can detect attacks.

On a mid-term basis, the parser's detection signatures could be improved regarding accuracy and processing capacity. Memory usage could be included to implement the ability to detect patterns or evaluate if a change in state values is justifiable or not. The ability to assess the likelihood of an attack or malfunction should also be included, improving the quality of the IDS further according to detection metrics presented in Section 3.4.3.

The case study assumes TCP/IP communication is installed on Levels 0 and 1 of the Purdue model. Such installments must become more common for this attack- and integrity detection to be valuable. In the long term, research should be devoted to further developing such technology and identifying potential challenges and usage areas.

Bibliography

[1] International Electrotechnical Commission. *Electropedia: International Electrotechnical Vocabulary - Part 903-01-19: Safety*. Online. 2013. URL: https://www.electropedia.org/iev/iev.nsf/display?openform&ievref=903-01-19.

[2] Robert Mitchell and Ing-Ray Chen. 'A Survey of Intrusion Detection Techniques for Cyber-Physical Systems'. In: *ACM Comput. Surv.* 46.4 (2014). ISSN: 0360-0300. DOI: 10.1145/2542049. URL: https://doi.org/10.1145/2542049.

[3] Marie Baezner and Patrice Robin. *Stuxnet*. Report 4. Version 1. Zurich, 2017. DOI: 10.3929/ethz-b-000200661.

[4] Mitre ATT&CK. *Triton*. 2022. URL: https://attack.mitre.org/software/S1009/ (visited on 4th Oct. 2022).

[5] M. Markusson. 'Intrusion Detection Systems for Operational Technology'. In: (2022).

[6] Sarah G. Freeman, Curtis St Michel, Robert Smith et al. 'Consequence-driven cyber-informed engineering (CCE)'. In: (Oct. 2016).

[7] Jeffrey R. Gellner and Curtis P. St. Michel. 'CCE Case Study: Baltavia Substation Power Outage'. In: (May 2020). DOI: 10.2172/1645032. URL: https://www.osti.gov/biblio/1645032.

[8] Matt Reif, Jeffrey R Gellner, Curtis P St Michel et al. 'CCE Case Study: Stinky Cheese Company'. In: (Oct. 2020). URL: https://www.osti.gov/biblio/1696803.

[9] Eunhyeong Shin and Sangcheol Hyung. 'Title of the Paper'. In: *Transactions of the Korean Nuclear Society Virtual Spring Meeting* (2020). URL: https://www.kns.org/files/pre_paper/43/20S-369-%5C%EC%5C%8B%5C%A0%5C%EC%5C%9D%5C%B5%5C%ED%5C%98%5C%84.pdf.

[10] Eric Gyamfi and Anca Jurcut. 'M-TADS: A Multi-Trust DoS Attack Detection System for MEC-enabled Industrial loT'. In: *2022 IEEE 27th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. 2022, pp. 166–172. DOI: 10.1109/CAMAD55695.2022.9966900.

[11] *Computer Security Resource Center OT*. https://csrc.nist.gov/glossary/term/ot. (Visited on 19th Oct. 2022).

[12] T. Williams. 'The Purdue Enterprise Reference Architecture'. In: *IFAC Proceedings* 26.2 (1993).

[13] Yan Hu, An Yang, Hong Li et al. 'A survey of intrusion detection on industrial control systems'. In: *International Journal of Distributed Sensor Networks* 14.8 (2018), p. 1550147718794615. DOI: 10.1177/1550147718794615. eprint: https://doi.org/10.1177/1550147718794615. URL: https://doi.org/10.1177/1550147718794615.

[14] Matthias Niedermaier, Florian Fischer and Alexander von Bodisco. 'PropFuzz — An IT-security fuzzing framework for proprietary ICS protocols'. In: *2017 International Conference on Applied Electronics (AE)*. 2017, pp. 1–4. DOI: 10.23919/AE.2017.8053600.

[15] Carlos E. Pereira and Peter Neumann. 'Industrial Communication Protocols'. In: *Springer Handbook of Automation*. Ed. by Shimon Y. Nof. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 981–999. ISBN: 978-3-540-78831-7. DOI: 10.1007/978-3-540-78831-7_56. URL: https://doi.org/10.1007/978-3-540-78831-7_56.

[16] OPC Foundation. *Unified Architecture*. 2008. URL: https://opcfoundation.org/about/opc-technologies/opc-ua/ (visited on 8th Mar. 2023).

[17] Karl-Heinz Niemann. 'Planning, installation and commissioning of Ethernet-APL networks'. In: *Ethernet-APL - Engineering Guideline* 1.14 (2022). URL: https://www.ethernet-apl.org/wp-content/uploads/APL-Engineering-Guideline-V114_1.14.pdf.

[18] *Cybersikkerhet for industrielle automatiserings- og kontrollsystemer*. Standard. Geneva, CH: Norsk Elektroteknisk Komite, 2021.

[19] *Application of IEC 61508 and IEC 61511 in the Norwegian Petroleum Industry*. Standard. Norwegian Oil and Gas Association, 2018.

[20] *Technical safety*. Tech. rep. The Norwegian Oil Industry Association, The Federation of Norwegian Industry, 2008.

[21] Petroleum Safety Authority Norway. *Terms and expressions*. 2023. URL: https://www.ptil.no/en/technical-competence/terms-and-expressions/ (visited on 14th Feb. 2023).

[22] Paul Denham and Alan Donnelly. 'Managing the Hazards of Flare Disposal Systems'. In: *Hazards Symposium* 25.160 (2015).

[23] International Electrotechnical Commission. *Electropedia: International Electrotechnical Vocabulary - Part 903-01-06: incident*. Online. 2013. URL: https://www.electropedia.org/iev/iev.nsf/display?openform&ievref=903-01-06.

[24] Einar Ravnås. 'Storulykketilsyn og tilsyn med teknisk sikkerhet på Kårstø'. In: (2014). URL: https://www.ptil.no/contentassets/d0b8a07926f44bd7855e679cffa4865d/2014_246-rapport.pdf.

[25] Bjarte Sandvik. 'Rapport etter gransking av hydrokarbonlekkasje på Heimdal 26.5.2012'. In: (2012). URL: https://www.ptil.no/contentassets/a4aca96b3ccd4727a5eb346497a5c41c/granskingsrapport---statoil---heimdal.pdf.

[26] Erik Hørnlund. 'Rapport etter granskning av utilsiktet nødavstengning og akutt oljeutslipp til sjø på Eldfisk kompleks i perioden 6. - 8. august 2014'. In: (2015). URL: https://www.ptil.no/contentassets/6fd70255daa2400990fdd8ca8ee54934/granskingsrapport---conocophillips---eldfisk.pdf.

[27] *Investigation Report*. Tech. rep. U.S. Chemical Safety and Hazard Investigation Board, 2007.

[28] International Electrotechnical Commission. *Electropedia: International Electrotechnical Vocabulary - Part 903-01-07: Risk*. Online. 2013. URL: https://www.electropedia.org/iev/iev.nsf/display?openform&ievref=903-01-07.

[29] International Electrotechnical Commission. *Electropedia: International Electrotechnical Vocabulary - Part 903-01-10: Risk Assessment*. Online. 2013. URL: https://www.electropedia.org/iev/iev.nsf/display?openform&ievref=903-01-10.

[30] National Institute of Standards and Technology. *Framework for Improving Critical Infrastructure Cybersecurity*. NIST Cybersecurity Framework. National Institute of Standards and Technology, 2014. URL: https://www.nist.gov/publications/framework-improving-critical-infrastructure-cybersecurity.

[31] International Organization for Standardization. *Information technology - Security techniques - Information security management systems - Requirements*. ISO Standard ISO/IEC 27001:2013. 2013.

[32] Yulia Cherdantseva, Pete Burnap, Andrew Blyth et al. 'A review of cyber security risk assessment methods for SCADA systems'. In: *Computers & Security* 56 (2016), pp. 1–27. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2015.09.009. URL: https://www.sciencedirect.com/science/article/pii/S0167404815001388.

[33] Per Håkon Meland, Karin Bernsmed, Christian Frøystad et al. 'An experimental evaluation of bow-tie analysis for security'. In: *Information & Computer Security* (2019). DOI: 10.1108/ICS-11-2018-0132. URL: https://doi.org/10.1108/ICS-11-2018-0132.

[34] Mary Mulcahy, Chris Boylan, Samuella Sigmann et al. 'Using bowtie methodology to support laboratory hazard identification, risk management, and incident analysis'. In: *Journal of Chemical Health and Safety* 24 (Nov. 2016). DOI: 10.1016/j.jchas.2016.10.003.

[35] Ioannis Zografopoulos, Juan Ospina, Xiaorui Liu et al. 'Cyber-Physical Energy Systems Security: Threat Modeling, Risk Assessment, Resources, Metrics, and Case Studies'. In: *CoRR* abs/2101.10198 (2021). arXiv: 2101.10198. URL: https://arxiv.org/abs/2101.10198.

[36] 'Industrial Control Systems: Cyberattack trends and countermeasures'. In: *Computer Communications* 155 (2020), pp. 1–8. ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2020.03.007. URL: https://www.sciencedirect.com/science/article/pii/S0140366419319991.

[37] Robert Lee. *TRISIS Malware: Analysis of Safety System Targeted Malware*. Online. 2017. URL: https://www.dragos.com/resource/trisis-analyzing-safety-system-targeting-malware/.

[38] N. Mathur. *Norsk Hydro shares a 4-minute video on how its employees stood up for the firm post an extensive cyberattack*. 2019. URL: https://securityboulevard.com/2019/04/norsk-hydro-shares-a-4-minute-video-on-how-its-employees-stood-up-for-the-firm-post-an-extensive-cyberattack/ (visited on 13th Feb. 2023).

[39] Norsk Hydro. *Cyber-attack on Hydro in brief*. 2020. URL: https://www.hydro.com/en/media/on-the-agenda/cyber-attack/ (visited on 13th Feb. 2023).

[40] Dragos. *ICS/OT cybersecurity year in review 2022*. URL: https://www.dragos.com/year-in-review/#section-report (visited on 22nd Feb. 2023).

[41] M. Assante and R. Lee. *The Industrial Control System Cyber Kill Chain*. SANS Institute, 2015.

[42] *ICS tactics*. URL: https://attack.mitre.org/tactics/ics/ (visited on 29th Oct. 2022).

[43] Livinus Obiora Nweke. 'A survey of specification-based intrusion detection techniques for cyber-physical systems'. In: *International Journal of Advanced Computer Science and Applications* 12.5 (2021).

[44] Robert Mitchell and Ing-Ray Chen. 'Effect of Intrusion Detection and Response on Reliability of Cyber Physical Systems'. In: *IEEE Transactions on Reliability* 62.1 (2013), pp. 199–210. DOI: 10.1109/TR.2013.2240891.

[45] INCIBE-CERT. *Design and Configuration of IPS, IDS and SIEM in ICS*. 2017. URL: https://www.incibe-cert.es/sites/default/files/contenidos/guias/doc/certsi_design_configuration_ips_ids_siem_in_ics.pdf (visited on 4th Oct. 2022).

[46] Andrew Hay, Daniel Cid, Rory Bary et al. 'Chapter 1 - Getting Started with OSSEC'. In: *OSSEC Host-Based Intrusion Detection Guide*. Ed. by Andrew Hay, Daniel Cid, Rory Bary et al. Burlington: Syngress, 2008, pp. 1–27. ISBN: 978-1-59749-240-9. DOI: https://doi.org/10.1016/B978-1-59749-240-9.00001-6. URL: https://www.sciencedirect.com/science/article/pii/B9781597492409000016.

[47] R. Bace. *Intrusion Detection*. Macmillan Technical Publishing, 2000.

[48] J. Koziol. *Intrusion Detection with Snort*. Sams Publishing, 2003.

[49] *Suricata*. URL: https://suricata.io/ (visited on 28th Oct. 2022).

[50] Prem Uppuluri and R. Sekar. 'Experiences with Specification-Based Intrusion Detection'. In: *Recent Advances in Intrusion Detection*. Ed. by Wenke Lee, Ludovic Mé and Andreas Wespi. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 172–189. ISBN: 978-3-540-45474-8.

[51] Mauro Conti, Denis Donadel and Federico Turrin. 'A Survey on Industrial Control System Testbeds and Datasets for Security Research'. In: *IEEE Communications Surveys & Tutorials* 23.4 (2021).

[52] Xuelei Wang and Ernest Foo. 'Assessing Industrial Control System Attack Datasets for Intrusion Detection'. In: *2018 Third International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC)*. 2018, pp. 1–8. DOI: 10.1109/SSIC.2018.8556706.

[53] M. N. Petersen. 'Detecting network intrusions'. MA thesis. Technical University of Denmark, 2014.

[54] Sarah G. Freeman, Curtis P. St. Michel and Nathan Hill Johnson. 'CCE Phase 1: Consequence Prioritization'. In: (May 2020). DOI: 10.2172/1617458. URL: https://www.osti.gov/biblio/1617458.

[55] Doug Buddenbohm, Sarah G. Freeman and Curtis P. St. Michel. 'CCE Phase 2: System-of-System Analysis'. In: (May 2020). DOI: 10.2172/1617457. URL: https://www.osti.gov/biblio/1617457.

[56] Stacey Cook, Sarah G. Freeman and Curtis P. St. Michel. 'CCE Phase 3: Consequence-based Targeting'. In: (May 2020). DOI: 10.2172/1617456. URL: https://www.osti.gov/biblio/1617456.

[57] Theodore Miller, Sarah G. Freeman and Curtis P. St. Michel. 'CCE Phase 4: Mitigations and Protections'. In: (May 2020). DOI: 10.2172/1617455. URL: https://www.osti.gov/biblio/1617455.

[58] *Framework for improving Critical Infrastructure Cybersecurity*. Framework. National institute of standards and technology, 2018. DOI: https://doi.org/10.6028/NIST.CSWP.04162018.

[59] Stefan Bollmeyer and Francisco Mendoza. *ABB brings Ethernet-APL with OPC UA to the field*. 2021. URL: https://new.abb.com/news/detail/80779/abb-brings-ethernet-apl-with-opc-ua-to-the-field (visited on 22nd Feb. 2023).

[60] Mesco. *ETHERNET-APL & OPC FOUNDATION COOPERATE*. 2021. URL: https://mesco-engineering.com/en-us/news/2021-06-25/ethernet-apl-opc-foundation-cooperate (visited on 22nd Feb. 2023).

[61] International Electrotechnical Commission. *Electropedia: International Electrotechnical Vocabulary - Part 903-01-17: Safety measure*. Online. 2013. URL: https://www.electropedia.org/iev/iev.nsf/display?openform&ievref=903-01-17.

[62] Etterretningstjenesten, Nasjonal sikkerhetsmyndighet and Politiets sikkerhetstjeneste. 'National threat assessment 2023'. In: (2023).

[63] *Parsers*. https://www.ibm.com/docs/en/app-connect/11.0.0?topic=overview-parsers. Accessed: May 28, 2023. 2023.

[64] K. Huseyin and P. Kumar. *Pyshark: Python wrapper for tshark, allowing python packet parsing using wireshark dissectors*. https://github.com/KimiNewt/pyshark. 2021.

[65] E. Owen, J. Coenders, X. St-Onge et al. *Asyncua: an OPC UA Python library built on asyncio*. Version 1.0.2. 2023. URL: https://github.com/FreeOpcUa/python-opcua.

[66] The Wireshark team. *Wireshark*. https://www.wireshark.org. Version 4.0.5. 2023.

[67] Brett Wujek, Patrick Hall and Funda Günes. 'Best practices for machine learning applications'. In: *SAS Institute Inc* (2016).

A Attack Scenario

An attack scenario is all the resources needed by an adversary to launch an successful operation initiating an HCE. This appendix contains the attack scenario used in the flare system case study. The scenario contains; a description of the HCE, target details, technical approach and critical needs. All components are described more closely in sections Sections 5.3 and 5.3.3.

HCE Description

The HCE consists of ignited liquids firing out of the mouth of the flame tower. The event would occur by overfilling the knock out drum of the flare system with liquids, resulting in the liquids entering the flare tower. Water would evaporate as smoke, but as crude oil has a higher boiling temperature residues would spread out as rain. The event would cause a production shutdown as it would hinder a safety-critical system from functioning. To restore production the flare system would have to be rinsed, but the main source of downtime would be forensic work. Production is not likely to precede until full system integrity is regained. Besides shutting down production, ignited liquids could cause injuries of nearby personnel or fires if in contact with flammable materials like biomass.

Target Details and Critical Needs

To deploy the attack, the attacker would have to access the system through a remote access. A critical need is therefore foothold in or access to a computer or home network of a authorized employee in addition to large parts of the information collected in the system-of-systems analysis. This means that all the information listed in Table 13 is required to design and deploy the attack. Timing is not strictly a critical need, but will greatly impact the effect of the attack and should therefore be considered.

Technical Approach

To make an impact the attacker would design a malware that overfills the knock-out drum without being detected or triggering the ESD system. The approach chosen consists of opening all BDV leading to the knock out drum while forcing the drain and return pump systems to remain closed. LAHH would have to be suppressed. The entire operation would have to be masked by sending fake sensor data between the PLC and HMI. The technical approach for deployment is summarized in Table 6.

Table 13: Critical needs for development

Component	Critical Needs	Location
Engineer Home Router	Specifications/data sheet	Vendor
	Configuration file	On board
Engineer Laptop	Specifications/data sheet	Vendor
	VPN	On board
	OS	Open source
EWS	OS	Open source
	Specifications/data sheets	Vendor
	Flare system project file	On board/file server
	Software and install process	Purchase
File server	Schematics - Network topology	On board
	Schematics - Communication diagram	On board
	Documentation - ESD procedures	On board
	Documentation - Control room layout	On board
	Documentation - Remote access procedures	On board
	Documentation - Remote access authorization list	On board
	Documentation - Flare system procedures	On board
Siemens S7-1500 PLC	Product specs / manuals	Open source
	Configuration File	Organization file server
	Vendor I/O Module Pinouts/Wiring Diagrams	Vendor Website
	Control Logic Diagram	Organization file server

Table 14: Technical approach

Target	**Access**	**Required actions**	**Timing/factor for triggering**
Engineering laptop	Compromised home Wi-Fi router access through remote access upon connection	Download malware for PLC and control room HMI	Immediately on connection to home Wi-Fi
EWS	Certificate-based authentication and VPN server configuration for remote access through OT firewall	The malware should reprogram the PLC controlling the flare system to open the BDV, disable the LAHH and return pumps and feed the LT, LS and TI with false readings	Immediately upon engineer laptop connection to OT network

B HCE Taxonomy

Figure 22: Enlargement of enabling functions upper part

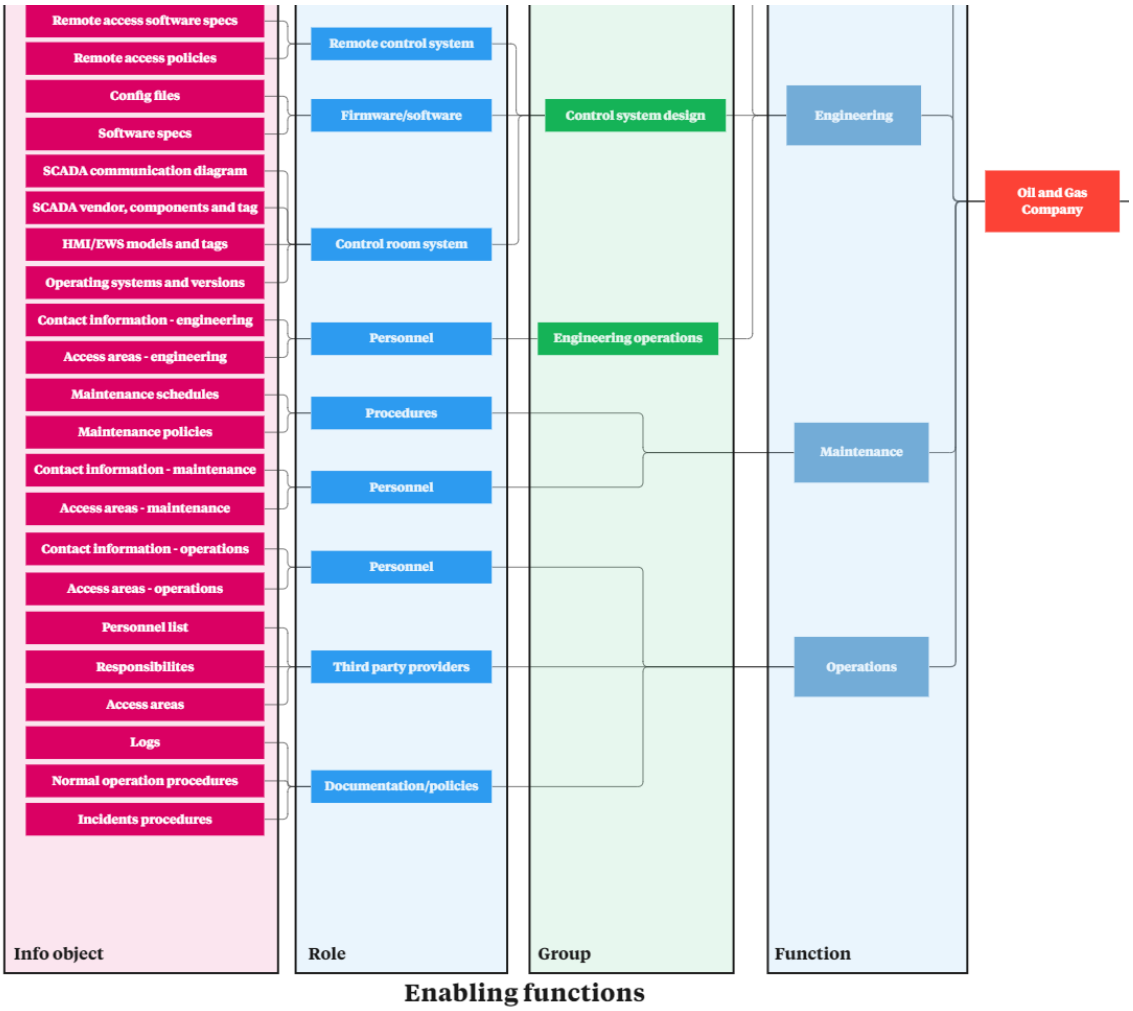


Figure 23: Enlargement of enabling functions lower part

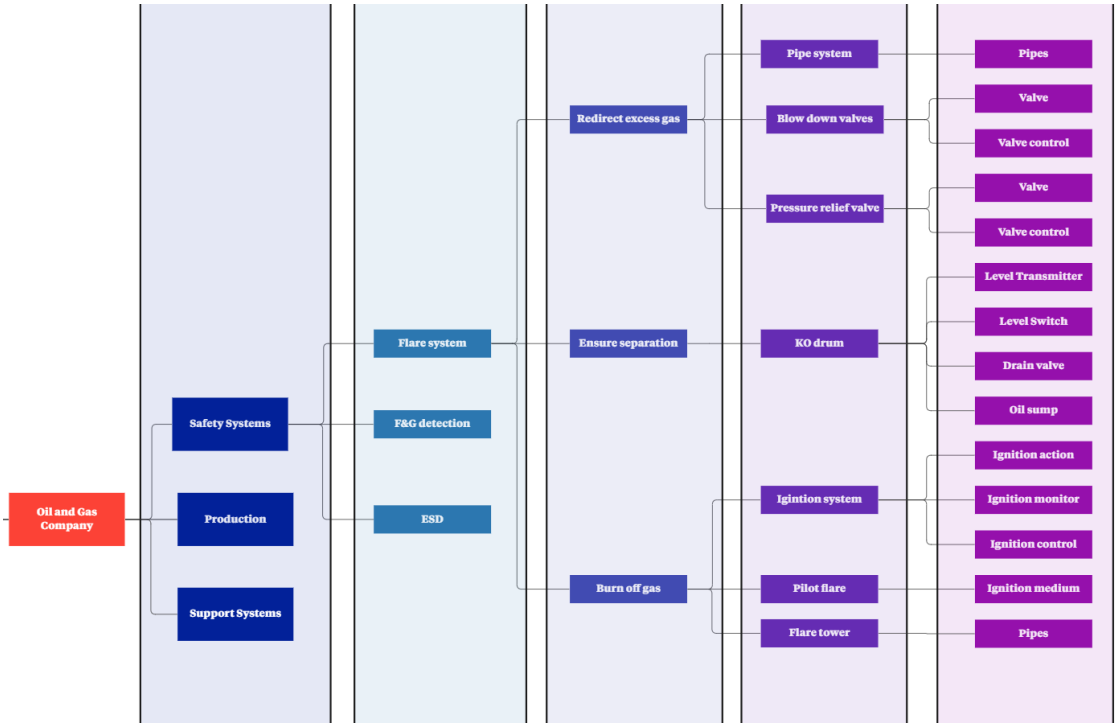


Figure 24: Enlargement of critical functions

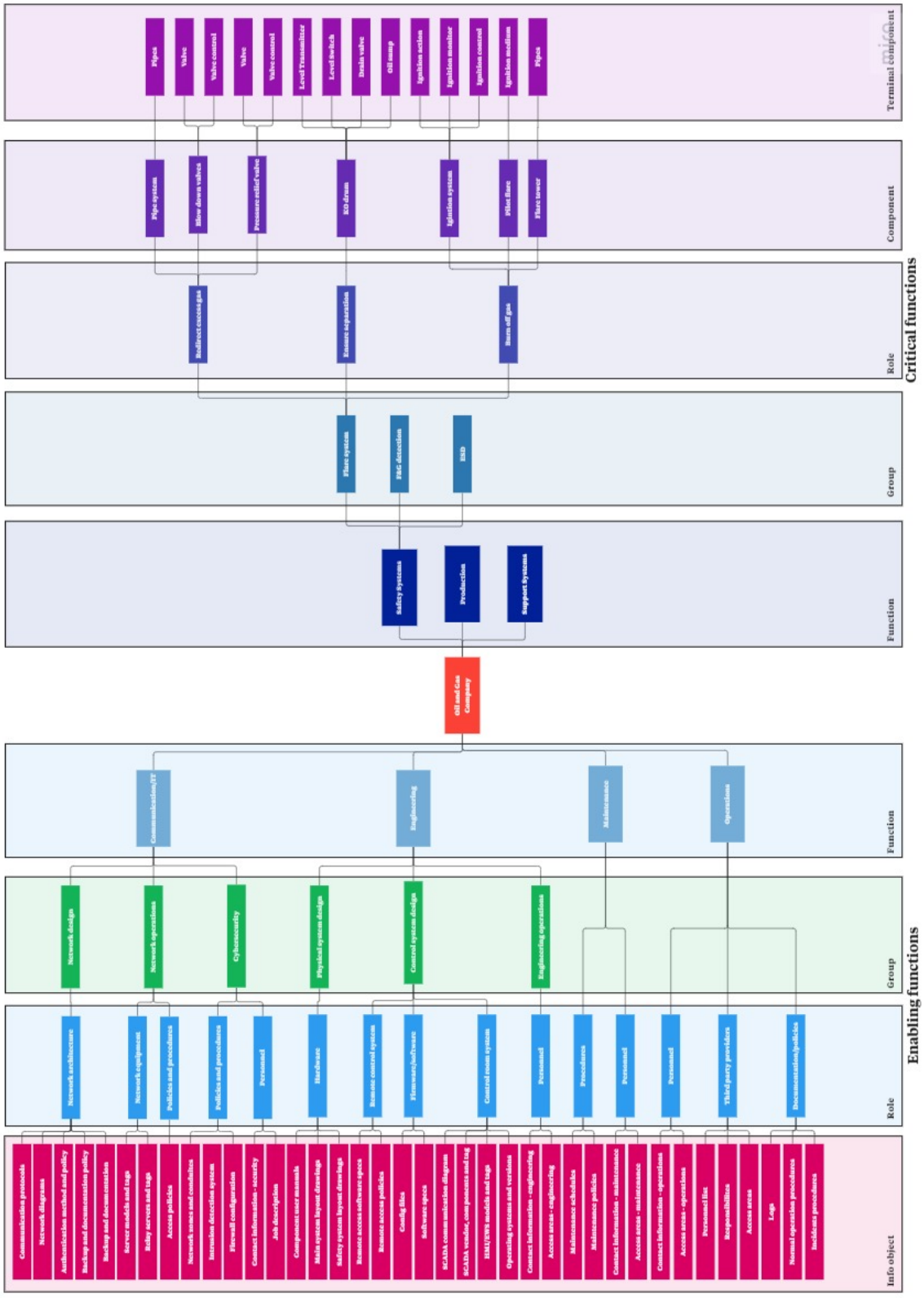


Figure 25: Complete HCE taxonomy

C Event Scoring Tables

Table 15: Cyber event scoring: ignited liquid spread

	None (0)	**Low (1)**	**Medium (3)**	**High (5)**
Duration ($\beta = 2$)				The event would likely trigger a production shutdown to rinse the flare system for liquid or burn off residues, as well as a longer outage due to administrative work like e.g. forensic investigations
Safety ($\delta = 3$)		Given that the spread of flares are within site parameters this event could cause burn injuries to personnel onsite		
Cost ($\gamma = 1$)			The cost of production shutdown is considered significant but not sufficient to cause liquidity crisis	
Environment ($\epsilon = 3$)			If the ignited liquid is spread out of land-based facility parameters it could cause wildfires	

Table 16: Cyber event scoring: process shutdown - compromised integrity

	None (0)	**Low (1)**	**Medium (3)**	**High (5)**
Duration ($\beta = 2$)		Loss of integrity would likely result in a shutdown but if no physical components need replacement, production can resume relatively quickly		
Safety ($\delta = 3$)	Given that no safety critical events are going on at time of shutdown, there is no safety consequences			
Cost ($\gamma = 1$)	Less than 5 days outage would have a cost, but well within recoverable amounts			
Environment ($\epsilon = 3$)	Inconsequential			

Table 17: Cyber event scoring: process shutdown - destruction of equipment

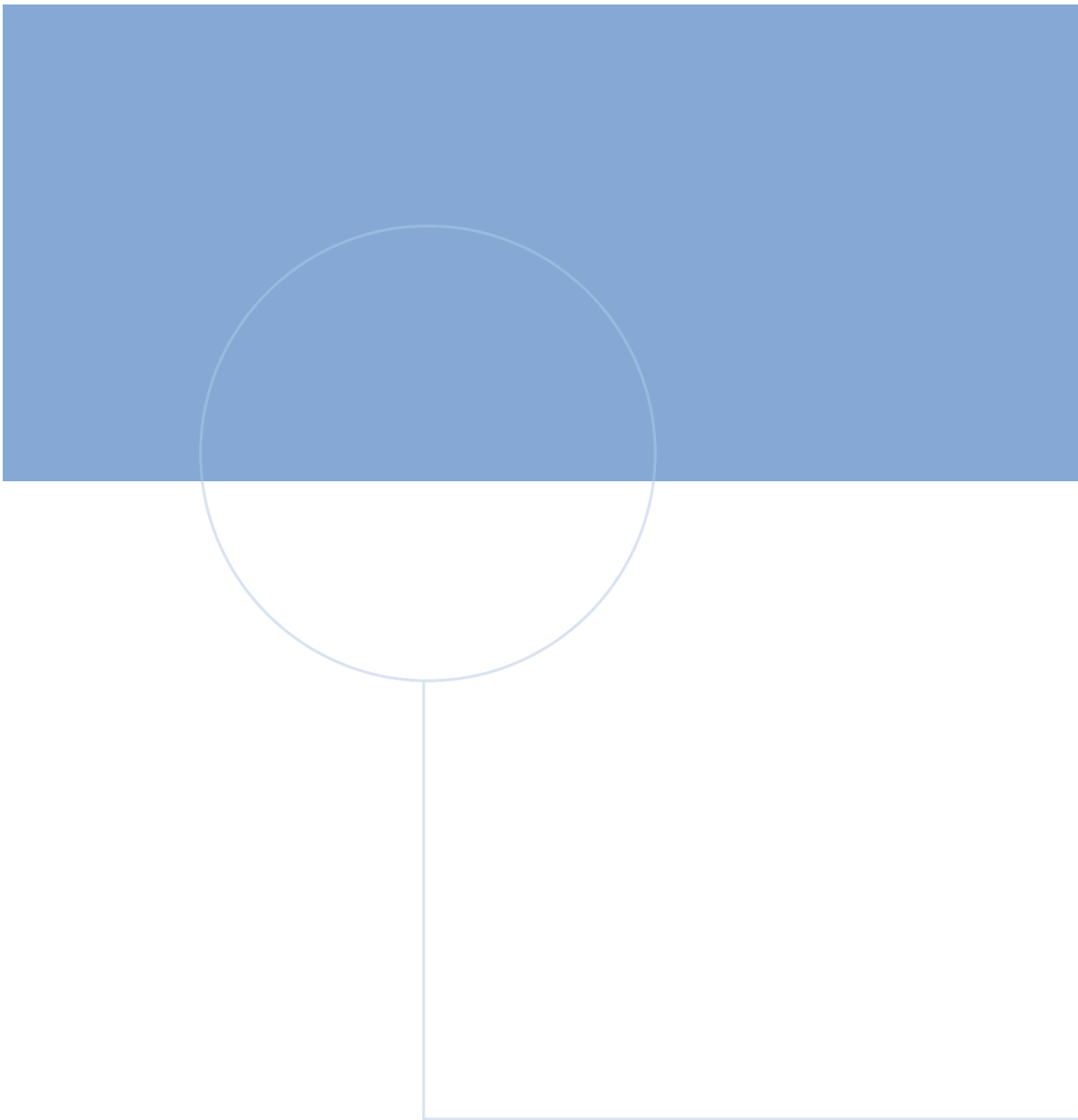
	None (0)	**Low (1)**	**Medium (3)**	**High (5)**
Duration ($\beta = 2$)				Destruction of equipment could cause restoration work lasting longer than 5 days
Safety ($\delta = 3$)		If personnel reside in proximity to the destroyed equipment it could lead to injures		
Cost ($\gamma = 1$)			Process shutdown at a juncture point facility for more than five days will introduce significant cost but not enough to trigger a liquidity crisis	
Environment ($\epsilon = 3$)		Depending on the destroyed part, one could expect minor leakages of liquids or hydrocarbons but not sufficient to make an severe impact		

Table 18: Cyber event scoring: jetfire or explosion

	None (0)	**Low (1)**	**Medium (3)**	**High (5)**
Duration ($\beta = 2$)				Destruction of equipment could cause massive restoration work
Safety ($\delta = 3$)				Fires and explosions could cause fatalities both within facility parameters but also spread of fires and projectiles off-site
Cost ($\gamma = 1$)			Restoring a juncture point facilities and long production stop would be very costly but not enough to trigger a liquidity crisis	
Environment ($\epsilon = 3$)		Projectiles and fires could impact surrounding environment and wildlife		

Table 19: Cyber event scoring: toxic flareout

	None (0)	**Low (1)**	**Medium (3)**	**High (5)**
Duration ($\beta = 2$)			The event would likely cause process shutdown to preform investigation and mitigating work, similar to the ignited liquid flare	
Safety ($\delta = 3$)				Fatalities caused by toxic gas both onsite and offsite [21]
Cost ($\gamma = 1$)		The event could cost the organization incomes caused by reputational damage		
Environment ($\epsilon = 3$)		Toxic gasses could damage surrounding animal life		



NTNU

Norwegian University of
Science and Technology