

Last line of defence

Cyber security of industrial control systems

M. Luchs

Delft University of Technology

LAST LINE OF DEFENCE

CYBER SECURITY OF INDUSTRIAL CONTROL SYSTEMS

by

M. Luchs

in partial fulfillment of the requirements for the degree of

Master of Science

in Offshore and Dredging Engineering

at the Delft University of Technology,

to be defended publicly on Wednesday October 26th, 2016 at 14:00 PM.

Supervisor:	dr. ir. C. Doerr	
Thesis committee:	Prof. dr. C. van Rhee,	TU Delft
	dr. ir. S. A. Miedema,	TU Delft
	Ir. E. van der Heijden,	Heerema Fabrication Group

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

PREFACE

Before you lies the thesis "Last line of defence: Cyber security of industrial control systems". This work investigates the state of cyber security within the offshore and dredging industry, the result of which has led to the proposal of a novel intrusion detection system for industrial control systems. It is written to complete the graduation requirements of the MSc program Offshore and Dredging Engineering at the Delft University of Technology.

The project has been undertaken in collaboration with Heerema Fabrication Group whom where looking to increase their awareness on cyber security. Investigating the state of cyber security within the offshore and dredging industry has led to the research question, which was formulated together with my supervisor from the TU-Delft, Christian Doerr. The work has proven challenging at times, in part because the subject is fairly unexplored terrain, and also my missing of a background in cyber security and computer networks. Nonetheless it has provided me with many avenues for growth and learning, especially since both the TU-Delft as HFG provided me the option to freely explore and thus gain insights broader then in one area of focus alone. Additionally both Mr. Doerr and my advisor from HFG, Mr. van der Heijden, were always available to provide valuable feedback and insight, and were willing to assist me when hitting roadblocks or I lost traction.

I would like to thank Frank van der Heijden, my supervisor at Heerema Fabrication Group for the support and many opportunities granted to me while working on-site in Zwiindrecht. Significant gratitude and appreciation also go towards Christian Doerr, my supervisor at the TU Delft, for his excellent advice and guidance during this process. For without his counsel this work would not exist in its current shape or form. I also wish to thank all those whom responded to my many questions, or helped me along my path while conducting this work. To other colleagues as both HFG and TUD: Thank you for all the pleasant cooperation.

Then there are some others whom I want to specifically mention. To my family: Thank you for your believing in me, your love and support has served me well, as always. Charlie Wolters, thank your for a listening ear, your kind words, and loving hugs. You provided a safe harbour whenever I felt lost. As for who helped me review my work and provided significant and valuable feedback on my work: Robbie Luchs, Erwin Junge, Nikol Guljelmović, and Sander Dragt. I am thankful and indebted for your efforts!

I hope you enjoy your reading.

*M. Luchs
Delft, October 16, 2016*

ABSTRACT

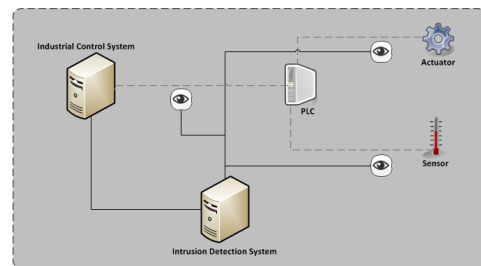
The world is rapidly embracing networked technology and transitioning into one of hyperconnectivity, a term first coined by social scientists Anabel Quan-Haase and Barry Wellman. Increased connectivity provides benefits such as automation and, remote access and control of networks and equipment, thereby decreasing operational costs. Maritime and offshore companies are increasingly automating their vessels and platforms to reduce the required workers on-board and centralise platform control.

With this tight coupling of complex ICT and industrial control systems however comes an increase in risks. These risks are further increased due to the application of security controls. Where in mechanical and structural engineering the focus lies on failure (e.g. safety factors), this is not necessarily the case for ICT related systems, which are often only verified to be working as specified and expected. Unexpected behaviour is not taken into consideration. Thus, while most vessels and platforms depend on automated systems, it seems little is being done to protect them from cyber incidents and attacks. The impact of security breaches on these systems can be disastrous due to the potential for physical damage to people and planet. This is especially true within the oil and gas industries. For example a fire at the Piper Alpha production platform in the North Sea in 1988, caused by an oil and gas leak, resulted in the loss of 169 lives. While computer viruses or worms might not directly injure people, or destroy equipment, automated control systems can.

This work thus focusses on the area where mechanical systems meet automation systems, a field called industrial control systems. An investigation into the current state of cyber security within the dredging industry has been conducted, which was followed by a threat analysis on industrial control systems. These systems operate at the heart of the dredging industry. This has revealed that malicious software can cause physical damage to equipment and injury to people. In an effort to improve the current state and help prevent cyber incidents from occurring the following research question has been formulated:

Can adversaries operating on Control System infrastructures be detected by an Intrusion Detection System which is monitoring the physical state?

To answer this question a novel intrusion detection system is designed which takes advantage of the physical state of the processes. This new concept deviates from other systems in that they obtain information from the network, as opposed to the physical process, where the data cannot necessarily be trusted. Additionally, when malicious events or cyber incidents occur within or behind the controller (PLC), the control network does not necessarily contain the required information detailing ongoing attacks. Looking at the physical system then allows for malicious attacks and cyber incidents to be detected by observing anomalous and unexpected behaviour of the monitored physical process. This enables the detection of advanced malicious threats which would be missed otherwise. The required information on the physical state of the process is obtained on the last line, between the controller and field devices.



CONTENTS

1	Introduction	1
1.1	Focus on industrial control systems.	2
1.2	Aim, Research Questions and Approach.	2
1.3	Intended Audience	3
1.4	Thesis Outline	3
2	Preliminary topics	5
2.1	Understanding the Cyber Security Landscape.	5
2.1.1	Malicious actors	5
2.1.2	Motives driving actors	6
2.1.3	Common Attack Methods	7
2.1.4	Consequences of security incidents	8
2.2	Understanding Industrial Control Systems	9
2.2.1	Control systems	10
2.2.2	Components	10
2.2.3	Architecture	11
2.2.4	Communication Protocols	12
2.2.5	Users.	13
2.2.6	Differences between ICT and ICS systems	13
2.3	Challenges securing Industrial Control Systems.	14
2.3.1	Traditional ICT solutions not always applicable	14
2.3.2	Legacy systems.	14
2.3.3	Patching and updating	15
2.3.4	Control Protocols	15
2.3.5	Commercial Off The Shelf hardware and software	15
2.3.6	Malware	16
3	Cyber security in offshore and dredging	17
3.1	Cyber incidents are on the rise	18
3.2	Cyber Security Awareness	19
3.3	Industrial Control Systems	20
3.4	Cyber security vulnerabilities and incidents.	22
3.4.1	Offshore and dredging related	22
3.4.2	Industrial control system related.	24
3.5	Discussion and Recommendations	25
3.5.1	Recommendations.	25
3.5.2	Security initiatives	26
4	Malicious software	27
4.1	Background.	27
4.2	Documented malware incidents	29
4.2.1	Conficker	29
4.2.2	Stuxnet.	30
4.3	Impact on industrial control systems	34
4.4	Obtaining malware	36
4.4.1	Control flow	37
4.4.2	Exports.	38
4.4.3	Implementation	39

4.5	Malware as a service	39
4.6	Conclusion	43
4.6.1	Recommendations.	43
5	A novel intrusion detection system	45
5.1	Introduction to intrusion detection	46
5.1.1	Type classification	46
5.1.2	Common detection techniques	47
5.1.3	Limitations.	48
5.2	Related work	49
5.3	Threat analysis	50
5.3.1	Assessment Scope	51
5.3.2	Actors and capabilities.	52
5.3.3	Existing security controls	53
5.3.4	Vulnerabilities	55
5.3.5	Risks	56
5.4	First principles monitoring system	57
5.4.1	Design objectives and assumptions	57
5.4.2	Control system decomposition.	59
5.4.3	Conceptual design	60
6	Building a prototype	67
6.1	Prototype design	67
6.2	Prototype implementation	69
6.2.1	Message bus	69
6.2.2	Event block	70
6.2.3	Analysis block	70
6.2.4	Response block	71
6.3	Feasibility study.	71
7	Evaluation	75
7.1	Strategy	75
7.1.1	Lab environment.	76
7.1.2	Experiments	77
7.1.3	Evaluation metrics	77
7.1.4	Challenges	79
7.2	Source model	82
7.2.1	Trailing Suction Hopper Dredger.	82
7.2.2	The model	85
7.3	Experiments	87
7.3.1	Experiment I: Cyber incidents	88
7.3.2	Experiment II: Manipulation attack	90
7.3.3	Experiment III: Envelope escalation	92
7.4	Evaluation and improvements	95
7.4.1	Value analysis	95
7.4.2	Consistency analysis.	97
7.4.3	Envelope analysis	97
7.4.4	Sample rate	98
7.4.5	Improvements	99
7.5	Dynamic Bayesian experiments.	100
7.6	Discussion	105
7.6.1	Future work	105
8	Concluding remarks	107
8.1	Key finding and their implications	107
8.2	Future avenues	109

A	Glossary and basic definitions	111
B	Prototype UML	113
B.1	Message bus	113
B.2	Event block	114
B.3	Analysis block.	115
B.4	Response block	116
	Bibliography	119

1

INTRODUCTION

The world is rapidly embracing networked technology and morphing into one of hyperconnectivity [1], a term first coined by social scientists Anabel Quan-Haase and Barry Wellman [2]. This relates to the trend that all things with the potential for networked communication will do precisely that: become interconnected.

Increased connectivity provides a large range of benefits to those making the transition. Such benefits are direct access to increasing troves of knowledge and information, improving efficiency, reducing costs and increasing profit margins. By connecting automation systems to these larger networks such benefits can be improved even further. Maritime and offshore companies are increasingly automating their vessels and platforms to reduce the required workers on-board and centralise platform control.

Maritime research facilities and businesses are starting to realise the potential for fully autonomous vessels and platforms. In the near future the world will see fully autonomous ships and platforms come to light, especially since research endeavours to realise this are fully ongoing [3]. One step in this direction is seen at Heerema Fabrication Group, whom recently constructed the Helwin Beta HVDC¹ offshore platform which will only be manned during maintenance. Within the dredging industry automation and networked technology is increasingly put to use, partly due to the profit margins, which are often based on the specific layers of sediment to be removed. This change in technology enables contractors to troubleshoot vessels and even change production parameters on the fly. All from shore-based locations. Interconnectivity of technology is leaving few - if any - industries untouched.

This hyperconnectivity between devices brings more than benefits alone. A fact that most are unaware of. Connecting devices to networks exposes them to the same risks and vulnerabilities that have plagued ICT equipment for years [4, 5]. In 2015 the world economic forum has identified cyber attacks as an emerging global threat [6]. Unfortunately those operating connected devices and (physical) systems remain blissfully uninformed about this rising problem, or simply turn their attention away by passing the problem on to another. Interestingly, speaking to CBS' 60 Minutes, FBI Director James Comey pointed out the following: *"there are two kinds of big companies in the United States. There are those who've been hacked by the Chinese and those who don't know they've been hacked by the Chinese"* [7]. Resembling the statement made in 2013 by U.S. Attorney General Eric Holder [8]. Both of these statements are indicative of the overall lack of awareness when it comes to risk posed by networked technology.

Looking at the maritime industry the 2011 report from ENISA, investigating the state of cyber security with relation to the maritime industry, concluded the same lack of awareness amongst professionals within the European maritime industry [9]. Knowledge on the potential consequences is absent, there appears to be a gap between industry cultures, and there is no cross pollination between cyber security professionals and those operating in other fields. This trend is reflected also by industry organisation, journals and media outlets related to the offshore and dredging industry. Only limited information and articles on the topic at hand are available, if any at all. With the maritime industry being classified as a critical infrastructure within Europe, their continuous operation is essential for the functioning of European society and economy. Safekeeping of maritime businesses and assets is of major importance. This is especially true since it has undeniably been shown that consequences of a cyber incident can affect the safety of automated processes, and with it the physical world [10].

¹High voltage direct current

1.1. FOCUS ON INDUSTRIAL CONTROL SYSTEMS

This work focusses on the area where mechanical systems meet automation systems, a field called industrial control systems. More specifically its focus lies on the cyber security of industrial control systems, a relatively unknown field that has only recently started to gain traction with a rising amount of news coverage.

Not only is the world rapidly becoming more networked, it is also becoming more automated. This is also true for the maritime and offshore industries that have highly technical equipment at their center. Automation and networking of this equipment reduces the need for human supervision and improves the bottom line of businesses. With this tight coupling of complex ICT and industrial control systems however comes an increase in risks [11]. Such risks are further increased due to the application of security controls. Where in mechanical and structural engineering the focus lies on failure (e.g. safety factors), this is not the case for cyber security which is often only verified to be working as expected. Unexpected behaviour is not taken into consideration. Thus, while most vessels and platforms depend on automated systems, it seems little is being done to protect them from cyber attacks.

The potential impact of security breaches on these systems can be disastrous due to the potential for physical damage to people and planet. This is especially true within the oil and gas industries. For example a fire at the Piper Alpha production platform in the North Sea in 1988, caused by an oil and gas leak, resulted in the loss of 169 lives [11]. While computer viruses or worms might not directly injure persons or destroy equipment, automated control systems can.

1.2. AIM, RESEARCH QUESTIONS AND APPROACH

With the rising importance and increasing prevalence of computer systems their safe keeping is becoming ever more important. This work aims to contribute to this safe keeping to which end the following endeavours have been undertaken:

State of cyber security Conduct an investigation into the current state of cyber security within the offshore and dredging industry and to get an impression of the current levels of awareness. This insight is then to be used to further improve the current state and inform industry professionals with respect to the importance that cyber security holds in an interconnected world, especially where control systems are concerned.

Malware and industrial control systems Most, if not all, maritime ships and platforms use automation to control processes aboard and their correct and safe operations are of prime importance to the safe operations of these vessels. There have been reports that malicious software found its way onto both interconnected and not connected automation systems. Within the industry there seems little knowledge on the effects that such infections can have on the safety and continuity of operations. This leads to the second part of the work, which is to investigate malicious software and in particular how it relates to control systems. Both this, as the first investigation are to be accomplished through desktop research and interviews with industry experts.

A novel intrusion detection system for industrial control environments Countering malicious software and other cyber incidents can be done using a variety of tools. Two examples of such tools are firewalls and virus scanners. Another such method is to use "defence in depth" strategies, which focusses on building resistant systems and build security in layers as opposed as a one layer solution. As part of the defence in depth strategy the concept of intrusion detection is often named and provides a valuable tool in detecting comprises. At the same time intrusion detection can provide critical information during forensics to prevent future compromises along the same path. Traditionally networked intrusion detection targets the network traffic between connected devices, searching for malicious behaviour, or data. This leads to the third part of this work and research question:

RQ *Can adversaries operating on Control System infrastructures be detected by an Intrusion Detection System monitoring the physical state?*

The first step in answering this research question is a literature study on IDS systems, their background and current solutions. From there an novel approach to intrusion detection system with respect to control systems is proposed, which is subsequently implemented and an experiment ran to both evaluate and demonstrate this new approach.

1.3. INTENDED AUDIENCE

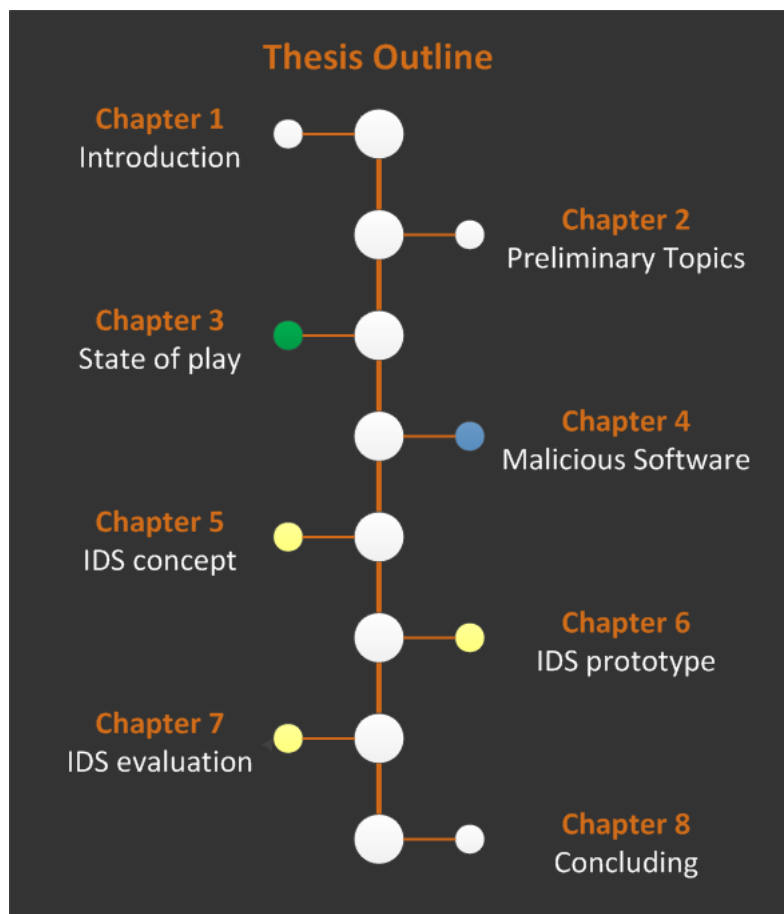
The broad goals driving this thesis created a rather broad audience who are somewhat opposing to one another. As such, a generalisation has been made which resulted in two main audiences, which are defined in the two paragraphs below. Building on these generalisation, the report overview is build in a way to indicate which chapters might be of interest to each group and which chapters might prove irrelevant. It does stand that the whole report has been written with a split personality, it is recommended to be read entirely by those with a clear interest in both the offshore and dredging industry combined with control system cyber security.

Offshore and dredging industry professionals These are the individuals that have a professional binding to the offshore and/or dredging industry, both technology oriented (e.g. engineers) and people oriented (e.g. leadership and management).

Cyber security professionals and researchers These are people who have a professional relationship to the cyber security sector, this applies both to researchers as corporate professionals.

1.4. THESIS OUTLINE

This work can be subdivided into three core topics: *State of play*, *malicious software* and a novel *intrusion detection system*. Each reflects a different research goal which are all related to the security of industrial control systems. The image below represents the outline of this work, depicting each of the chapters and their correlation by colour. Where the colors represent the following: green - the state of play, blue - malware in an industrial environment, yellow - a novel intrusion detection system of industrial control systems. This figure is followed by a short abstract of each chapter, including an indication to its primary intended audience.



Chapter 2 - Preliminary topics This chapter provides background information that is useful to understand the remainder of the thesis, especially for readers with a low working knowledge in cyber security, control systems and the combination of both fields: cyber security of control systems.

Chapter 3 - Cyber security in offshore and dredging This chapter discusses the current conditions of cyber security within the offshore and dredging industry. Various topics touched upon are the lack of awareness, examples of breaches and the efforts which have been undertaken to help increase awareness within the industry. This chapter is based on information available from literature, industry magazines and interviews with experts. The main audience targeted is the offshore industry professionals.

Chapter 4 - Malicious software This chapter presents available research that investigates the effect that malicious software, both traditional and designed for industrial control systems, has and can have on industrial control systems (ICS). This is followed up by the efforts to create a malicious software sample for demonstration purposes. For the literature study the target audience is both the offshore industry professionals as well as security researchers, whereas the efforts into a malicious software example are mostly aimed at security researchers and those putting efforts into raising security awareness.

Chapter 5 - A novel intrusion detection system This chapter proposes a novel intrusion detection system (IDS) concept to protect against advanced threats in industrial control systems. First the background of intrusion detection systems is presented with traditional approaches and solutions, which is followed by related research investigating intrusion detection specifically targeting control systems. This then leads to the introduction of the new conceptual intrusion detection system.

Chapter 6 - Building an IDS prototype With a conceptual intrusion detection system in place, a prototype is to be built which can be used for evaluation of the concept. First the basis for the design will be investigated, determining the inner workings of the prototype. With a firm base the main implementation details will be discussed.

Chapter 7 - Evaluation of the prototype A novel concept and prototype have been created. The final stage is the evaluation which will determine its performance. First a strategy will be developed and a testing environment selected. Next the experiments will be designed and then executed. The results from these experiments will be presented and then used to discuss the prototypes performance and feasibility of the proposed concept.

Chapter 8 - Concluding This chapter wraps up this work by summarizing the main findings and makes suggestions for directions of future work.

2

PRELIMINARY TOPICS

This chapter provides the reader with background knowledge that will help the further reading of this work. It starts out with an introduction to the field of cyber security in section 2.1 to provide background to those unfamiliar with the field. For those unfamiliar with the field of control systems a short introduction to industrial control systems is given in section 2.2. This is followed in section 2.3 by common challenges encountered in the securing of industrial control systems, as this outlines the importance of research into the security of control systems and thus this thesis work. Finally a glossary with basic definitions can be found in appendix A.

2.1. UNDERSTANDING THE CYBER SECURITY LANDSCAPE

This section is intended for those new to the field of cyber security and will provide a basic understanding on the security landscape. The intention here is to make the reader more aware of cyber security and the important role it plays within a computerised and networked world. For an act of (cyber) crime to occur there are three requirements that have to come together: 1) method, 2) opportunity and 3) motive [12]. This section uses these requirements to discuss the various *malicious actors*, the *motivations* driving them and the *attack methods* they utilise. The section ends by discussing the consequences malicious acts have and their negative impact on individuals and businesses.

2.1.1. MALICIOUS ACTORS

The rise of networked devices and equipment is accompanied by an increased number of cyber security incidents (see also chapter 3), which makes it important that society is aware of the larger picture. This is especially true for those directly responsible. Cyber threats and incidents are more than exploits of system vulnerabilities and malicious software (malware; discussed on page 8), they are executed by human actors with motives of their own. A significant element when investigating mitigating measures to prevent cyber incidents is a risk analysis. When creating such an analysis, it is important to take the human actors behind cyber incidents into account because they can have a direct effect on the required mitigating measures and involved costs. Protecting a system against a random script kiddie¹ who is trying to deface a website, is far from comparable to the protections required when facing a career criminal with eyes on the prize.

Computer criminals, like other criminals, come in many shapes and guises. They might have a university degree, wear suits, or contribute to their communities. This section then provides a stereotypical overview on the human actors who are responsible for a multitude of cyber incidents.

Amateurs To date, most of the computer crimes reported have been committed by amateurs [12] and insiders [13]. These are the people observing a weakness in the security of a system which gives them access to a thing of value and decide to use, instead of report, the weakness for their personal gain. Such a vulnerability can be as simple as accessing poorly secured Wi-Fi access points while waiting for a personal internet connection after moving to a new apartment. Employees that are denied a promotion, or are reprimanded, might become disgruntled and decide they want to get even. If they gain access to an ICT system, they can potentially wreak havoc on this and connected systems. Another form of amateur is the bored kid (or any

¹In hacker culture a script kiddie is an unskilled individual who uses scripts or programs developed by others to attack computer systems and networks, and deface websites.

person for that matter) that tries to run random scripts off the internet, or just copies and pastes tutorials, without really knowing what they are doing nor being aware of potential side effects of their actions [14]. These amateurs are often called "Script Kiddies". Mentioned so far are people who act in bad faith, but this is not always the case. Often people are simply unaware of the consequences of their actions. For example the charging of malware infected phones through corporate ICT systems, playing games -in some cases even on critical systems-, or falling victim to a social engineering attacks.

Hackers Hackers (traditionally known as crackers²) are those who attempt to gain access to systems for which they have no authorization. They tend to have more than a working knowledge of computer systems and are often responsible for writing the scripts that are used by amateurs. Often they are driven by the pure challenge of gaining access, curiosity and other forms of personal gain. Hackers are often placed into one of the following three stereotypes: Black-hats, white-hats and grey-hats. Black-hats are often looking for a score and bragging rights [15], enjoying the ensuing chaos that results from their actions. No common ground or motivation binds these actors [12]. White-hats on the other end walk the thin line of breaking into systems (at times without authorization) for the benefit of the public and greater good. To achieve this goal they disclose their discoveries to the manufacturer to have them fixed before incidents occur, or otherwise releasing the vulnerabilities to the public to inform about the issue. Grey-hats then are those individuals whom exhibit traits seen by both black- and white-hats.

Professional Criminals The professional criminals, also known as the career criminals, use their knowledge of computer systems for monetary gain and status. They are willing to put in considerable work and effort to gain access to systems while aiming to remain undetected for as long as possible. This enables them to extract what they need over a period of time while the target remains unaware of any criminal activity [15]. Because of their requirements, professional criminals invest in methods and tools that are neat, robust and able to remain undetected [12].

Nation States With an every increasing number of devices gaining networking capabilities, the internet will increasingly act as the stage for various missions conducted by nations states. Such missions are often conducted by three letter agencies or the cyber division of the armed forces. In the past, the intention behind these missions was said to be the collection of intelligence which was used for the protection of the nation state. These days an expansion on this purpose has been taking place making networked systems an important aspect of modern warfare. To achieve their goals this often means that a large part of the internet traffic is stored and investigated. To get a feel for the scale of these missions one only has to look at the revelations by Edward J. Snowden or the sophistication of the Stuxnet malware. Examples range from eavesdropping on a large number of people their communications to disabling physical devices by sophisticated malware [10, 16].

Other miscreants There are many more stereotypical generalisations that could be discussed or expanded on in this topic. Protesters, terrorists or activists for example are but a few of them. The aim of this section is to give a general introduction to diverse flavours. More detailed information and research is available in this field. For a more in depth discussion and potential starting point for further research the reader is directed towards the paper by Holt and Kilger [17].

2.1.2. MOTIVES DRIVING ACTORS

The motivations that are driving the people behind security breaches are as diverse as they themselves are. Nonetheless, an indication to what could drive them is given in this section. The complexity of what motivates people is a domain of psychology which can shine a light on what drives the actors behind cyber incidents. These can be as diverse as the actors themselves. This section aims to give a high level overview of these motives.

Political gain When looking at groups like nation states, or more generic militant movements, there is the potential for political gain. If access can be gained into the digital systems of ones adversary, one might be able to learn about their plans, resources, or other relevant information streams. Besides political espionage

²Historically hackers build things, and crackers break them.

one would be able to infiltrate adversary systems and disabling them without even being near the actual equipment, which is precisely what Stuxnet was created for [18].

Economic gain Participants in the 2012 World Economic Forum declared data to be a new class of economic asset, much like currency or gold. A company might want to learn of the research that a competitor has in the pipeline. Whether this is to know how to counter new products, beat them to a copyright claim or to be used by insider trading. All of it is related to a type of financial gain. Career criminals engage in phishing attempts to collect credit card information, trick users into installing malware on their devices that assimilate them into a botnet, and run a plethora of other scams. These actions are often based on the financial benefit it provides to the perpetrator.

Civil disobedience In the physical world people have to convene at locations when protesting for, or against a specific cause. With the arrival of the Internet this location has changed to the comfort of ones own home. Hacktivists argue that through the defacement of websites (digital graffiti), or the launching of denial of service attacks³, one is able to protest a cause and express his or her opinion by means of cyber squatting. More worrisome though, is the risk that an activists gains access to the control systems of, for example, an offshore oil platform. This would give them the power to control the platform and stop production or cause physical destruction. Even though a platform resides miles offshore in the most inhospitable environment, an activist can operate from the comfort of a local Starbucks. Another benefit for protesters which is gained by these digital protests is that it is virtually impossible to trace the attack back to its original source. That is, presuming the attacker is knowledgeable enough to hide and cover her, or his, tracks.

Research and awareness There are those who are trying to improve the overall security of our lives, be it corporate, governmental or individual. While this is not the place to debate on the ethics of doing research without consulting the involved parties, these actions have often improved the security of the investigated products. To some this increase in security, and their personal interest in the field is enough motivation.

Lulz A lesser known term among most people, but well established with those actively involved in online communities, is Lulz. *The Urban Dictionary* defines Lulz as:

Beginning as a plural variant of lol, Lulz was originally an exclamation but is now often used as a noun meaning interesting or funny internet content. Lulz is the one good reason to do anything, from trolling to rape. After every action taken, you must make the epilogic dubious disclaimer: "I did it for the lulz".

For some people the pure enjoyment of their actions, and the inevitable ensuing chaos, is enough to justify their actions.

2.1.3. COMMON ATTACK METHODS

This subsection will give an idea into the various kind of attacks methods employed by malicious actors who try to compromise computer systems. It is important to note though that this text does not provide a full overview of all potential attack vectors and approaches.

Phishing Phishing is the act of enticing a user to click on some link that brings them to a malicious website. This is often done by incorporating the link into an email, which is then send to a large group of (targeted) users. The website is often created to resemble an official website, such as a bank, but has the purpose to steal login credentials, install malicious software, or another act of cyber crime. Spear phishing is a variation on phishing where only a specific person or a select group of people is targeted. The messages in a spear phishing attack are often highly detailed and resemble a sender that is known to the target(s). Information required to create such messages are often learned during by closely investigation of a target, through which as much information as possible is gathered, e.g. by using social media.

³Overloading a server with more requests than it can handle, also known as (D)DOS.

Malware Malicious software, commonly known as malware, are programs that are defined by their malicious intent to go against the requirement of the user. Malware is used for the disruption of computer operations, gathering of sensitive information and even to provide access to systems for nefarious actors. This definition thus means that software causing unintentional harm due to some imperfection does not fall under this definition. Malware will be further discussed in chapter 4 on page 27.

Weak Authentication Authentication is often required to access systems before they can be used. While there are many forms of authentication the most known is arguably the combination of user name and password. When authentication systems, or the used credentials, are weak it is possible for nefarious actors to gain unauthorized access. An example of a weak user credential is the use of an easy-to-guess password such as "password", or having login credentials lingering around ones workplace as demonstrated by figure 2.1.

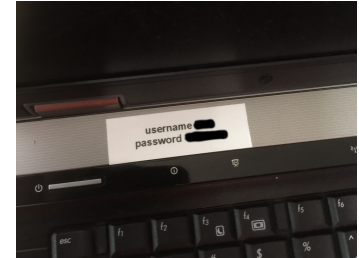


Figure 2.1: Example of weak authentication

Weak physical security Why would an attacker go through the trouble to get access to a target's networks, when it is much simpler to simply walk into the offices and sit down behind a work station, drop USB sticks, or attach various devices to computer systems and power outlets, which are then used to monitor personnel and relay this information back to the attacker. Relaying of information can be done by connecting to the network of the target and establishing an connection to a command and control server on the internet. An attacker might even go as far as stealing easy to carry equipment such as laptops and tables and investigate them for useful information in a safe location.

Social Engineering Social engineering is the act of psychological manipulation of people such that they perform actions or provide (confidential) information to an attacker (also known as a social engineer). Social engineering is about hacking the human mind, something that is quite often significantly easier then finding a vulnerability in a target's network or software.

Reconnaissance Reconnaissance is the act of gathering intelligence on a target. Much like a thief planning a bank heist who will start by investigating the bank, counting the number of guards and possible exits. So too will a clever cyber criminal spend time investigating his or her mark. There are multiple ways to accomplish this, from conducting scans on a target's systems, to social engineering or even information accessible from bulletin boards or available documentation. Pfleeger [12] states that "the best defence against reconnaissance is silence. Give out as little information about your site as possible, whether by humans or machines". Reconnaissance is not actually a direct method of attack. Nonetheless, it often plays an important role when it comes to targeted efforts and provides the required information to initiate attacks, because of this reconnaissance is shortly discussed.

2.1.4. CONSEQUENCES OF SECURITY INCIDENTS

Often people and corporations are unaware of the actual consequences that a cyber security attack can have. This section aims to provide a basic and broad understanding of the effects these events can have on individuals, businesses and in some cases even society.

Espionage and data breaches Espionage comes in many shapes and sizes, and with the interconnected infrastructures cyber espionage is hardly a surprise. Clayton [19] reports that "*In almost all cases of cyber espionage, the losses accrue as a steady drip of lost profit via unfairly empowered competitors and resources diverted to cybersecurity related expenses rather than through one sudden catastrophic blow*". Thus, its effects are hard to measure and prove, but are nonetheless there. As an example think only of proprietary information, which can undermine a companies standing with peer firms and even jeopardize its survival over time.

This is not just fiction; the 'Night Dragon' attacks targeted global oil, energy and petrochemical companies. Although the actual goal remains unknown, it seems those behind the attacks were after corporate emails and sensitive documents[20].

It has been reported that the NSA intercepted phone calls and faxes when Airbus and Saudi Arabia's national airline were negotiating a contract. This contract, aimed at modernizing Saudi Arabia's fleet, was worth

\$6 billion dollars, was won by McDonnell Douglas instead. Interestingly, after the intercepted information was first forwarded to Airbus' US competitors and then made public [21].

Every incident has its own consequences and should be looked at seriously. Espionage attacks might be a precursor to a more sinister piece of malware targeting systems, especially when this is with respect to highly complex control systems. The next section will go more into this aspect; an attacker with enough knowledge of a system can cause serious havoc.

Failure of physical systems Another major consequence faced by the industries is the disruption of digital and physical systems. Such disruptions can range from partly and temporary shutdown, full blackout condition or even adversaries that takeover control of equipment without operators being aware. Unfortunately this is no longer mere speculation.

The first publicized instance of physical disruption due to a cyber attack dates from 2000 and involved the water sector [22]. A disgruntled contractor retained access credentials to the control system of the Maroochy (Australia) Water Services facility, which he helped install, after his contract was terminated. His actions caused 800,000 litres of raw sewage to spill out into the grounds of a Hyatt Regency hotel, local parks and rivers [23]. A representative of the Australian Environmental Protection Agency said the following: "Marine life died, the creek water turned black and the stench was unbearable for residents" [24], clearly indicating the severity of the incident.

Early 2013 malicious software got access to a number of offshore platforms after workers unintentionally downloaded it to their workstations. This malware then incapacitated the networks it found itself connected to [25]. In this specific situation the consequences did not pose physical danger, however such a breach of security exposes people and environment alike to serious risks and possible injuries. Generally speaking, the effects malware can have on control systems remains unknown. Scientific literature includes numerous studies on the effects of malware on ICT equipment, but very few field tests have analysed the effects malware has on ICS [26]. Nonetheless, research in this area is on the rise. One such example is the investigation into the effects that four traditional ICT malware(s) - namely Code Red, Nimda, Slammer and Scalpe - have on an ICS network [26]. The most dramatic consequences is the complete clogging of the network by Slammer, causing remote terminal units (RTUs) to lose their functionality.

Malware targeting ICS specifically is on the rise [27], to which the best known example is the highly sophisticated piece of malicious software named "Stuxnet" [10, 28]. Created by the US and Israel [18], its aim was to influence centrifuges in the Natanz nuclear facility in Iran to "self-destruct" by increasing their rotation frequency past their limits. To keep the operators in the dark that the centrifuges were misbehaving, the return signals from the PLCs were altered to report normal operation conditions. Only the increased number of centrifuge failures was out of the ordinary. If anything, the Stuxnet malware demonstrated the viability of a cyber attack against ICS systems [29].

Impact on business operation Within the offshore and dredging industries the accident frequency rate (AFR) is an important statistic. The AFR is a statistical metric that relates to the number of incidents with human injury and/or loss of life over a period of time. Because this statistic is used during the tender phase of a project, companies with a high AFR will have a harder time in acquiring new assignments. Some countries even go so far as to exclude companies from a tender if certain requirements are not met. This connects back to cyber security due to the potential for control system incidents, which can lead to injury or loss of life. This is one way in which a security breach can influence the monetary side of a business and its continuity, which is in addition to the immediate impact experienced during the attack. During the KIACS 2014 conference the petroleum industry was indicated to be the one most affected by such indirect consequences [30].

Worse than delayed or more difficult business operations is going out of business due to a cyber incident. Although this might sound unlikely it is exactly what happened to a company called Code Spaces. After a malicious actor gained access to their infrastructure, all of their data -including backups- were removed from the system. No external backups were available which made it impossible for them to recover from this event. They had to close shop shortly thereafter [31].

2.2. UNDERSTANDING INDUSTRIAL CONTROL SYSTEMS

Industrial Control System (ICS) is an overarching term covering a range of control systems used in industrial environments. These systems range from geographically distributed - such as supervisory control and data acquisition (SCADA) systems - to local configurations such as distributed control systems (DCS). Typically

used in larger industries such as oil, gas, electricity, these computer-based systems monitor and control the industrial processes. This means that they are specifically designed for, and are dependent on, the underlying physical processes. Running an offshore converter station will employ a totally different set of controls than a dredging vessel, for example.

With readers that have a technological background in mind this section provides an elementary understanding on ICS. The first step is to explain the concept that underpins the term control system (Section 2.2.1). This concept however would not function were it not for the basic components that are used at its core. As such section 2.2.2 will discuss these. These components will need to communicate with one another and with the main servers, this makes a structured network architecture an important components of an ICS (Section 2.2.3). This network architecture provides the channel along which to communicate, section 2.2.4 discussed the protocols which enable the devices to understand the information that is being relayed over these channels. While these components and systems are designed to operate automatically, they are not completely autonomous, meaning that human operators and engineers play an important role within ICS (Section 2.2.5). An important realisation is that while ICT and ICS are overlapping domains, they are by no means equal which will be discussed in section 2.2.6.

2.2.1. CONTROL SYSTEMS

The term control system can be applied to any system which influences the behaviour of various other devices or systems. As such, the human body can be seen as a classic example of a control system. Other examples are an operator opening or closing a hydraulic press or an airplane operating on automatic control. For all of these the goal is the same, which is to regulate the behaviour of another device or systems.

The field of control engineering is specifically tasked to design the decision making process (logic) and system (components) that governs the physical process and ensures it behaves as intended. An elementary process which is under control is represented by figure 2.2.

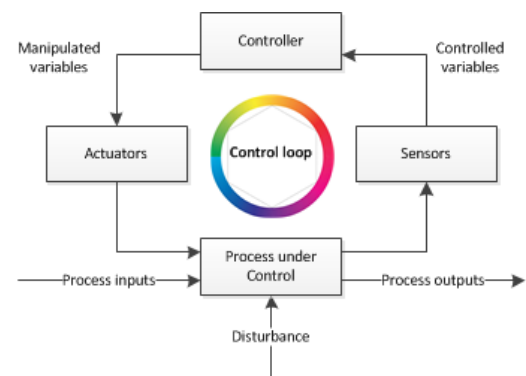


Figure 2.2: An elementary representation of a control system

2.2.2. COMPONENTS

A typical industrial control system consists of a large number of different components to achieve its control purpose. This sections only presents the most commonly encountered components and is thus not an extensive list. The description are based on the info-graphic provided by the ICS-CERT Secure Architecture Design definitions [32].

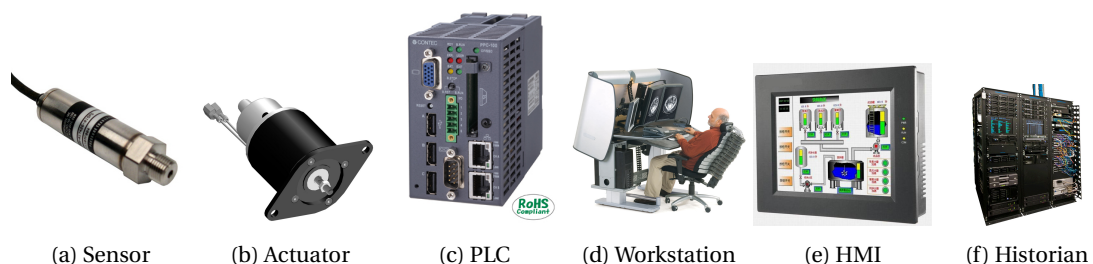


Figure 2.3: Components often used by industrial control systems

Sensor: A device that is designed to respond to a physical stimulus - such as heat, sound, pressure or motion - and transmit a signal as a response. This signal can then be used for measurements or to control a physical process. These devices are designed to have a minimal impact on that which is to be measured, which is often achieved by making the sensor smaller.

Actuator: A mechanical device that is designed to influence the physical environment by moving or controlling something. As such, the actuator is the mechanism through which a control system -human or software-based- can act upon its environment, utilising the actuator to position the process in a desired state. These devices are operated by a source of energy. While this is typically an electric current, this is not always the case. Other possibilities are hydraulic fluid pressure or pneumatic pressure. This energy is then converted into a motion.

Programmable Logic Controller: A programmable logic controller (PLC) is a computer that contains logic which is used to control physical processes based on input values received from sensors and the larger control network. They differ from better known desktop computers in that they are designed for harsh conditions, employ a more stable operating system and have to adhere to strict real-time conditions. The latter means that an output result must be produced within a specified limited time frame in response to input conditions. Missing this time frame can have unintended operations as a consequence that cause the control operation to fail.

Engineering/Operator Workstation: The engineering workstation is usually a high-end very reliable computing platform designed for configuration, maintenance and diagnostics of the control system applications and other control system equipment. The system is usually made up of redundant hard disk drives, high speed network interface, reliable CPUs, performance graphics hardware, and applications that provide configuration and monitoring tools to perform control system application development, compilation and distribution of system modifications.

Human Machine Interface: In computer science and human-computer interaction, the Human-Machine Interface (HMI) refers to the graphical, textual and auditory information the program presents to the user (operator) using computer monitors and audio subsystems, and the control sequences (such as keystrokes with the computer keyboard, movements of the computer mouse, and selections with the touchscreen) the user employs to control the program. Currently the following types of HMI are the most common:

- Graphical user interfaces (GUI) accept input via devices such as computer keyboard and mouse and provide articulated graphical output on the computer monitor.
- Web-based user interfaces accept input and provide output by generating web pages which are transported via the network and viewed by the user using a web browser program.

The operations user must be able to control the system and assess the state of the system. Each control system vendor provides a unique look-and-feel to their basic HMI applications. An older, not gender-neutral version of the term is man-machine interface (MMI). The system may expose several user interfaces to serve different kinds of users. User interface screens may be optimized to provide the appropriate information and control interface to operations users, engineering users and management users.

It can be argued that haptic or kinesthetic feedback control is also a form of HMI, as these controls offer an interface between human and machine. However for the purposes of this work the HMI will be seen as described above.

Historian: A centralized database located on a computer installed in the control system DMZ supporting external corporate user data access for archival and analysis using statistical process control and other techniques.

2.2.3. ARCHITECTURE

To facilitate the communication between components, a network is required. These networks come in a large range of configurations and designs, but they can be generalised to some extent. This generalisation is seen in figure 3.1. From right to left these segments will be shortly explained here.

Physical system: The physical systems encompasses the physical processes which the control system is there to monitor and control. In a hopper dredger for example this can be the physical state of hopper contents or the propulsion system (hull, engine, propeller) maintaining position during heavy weather.

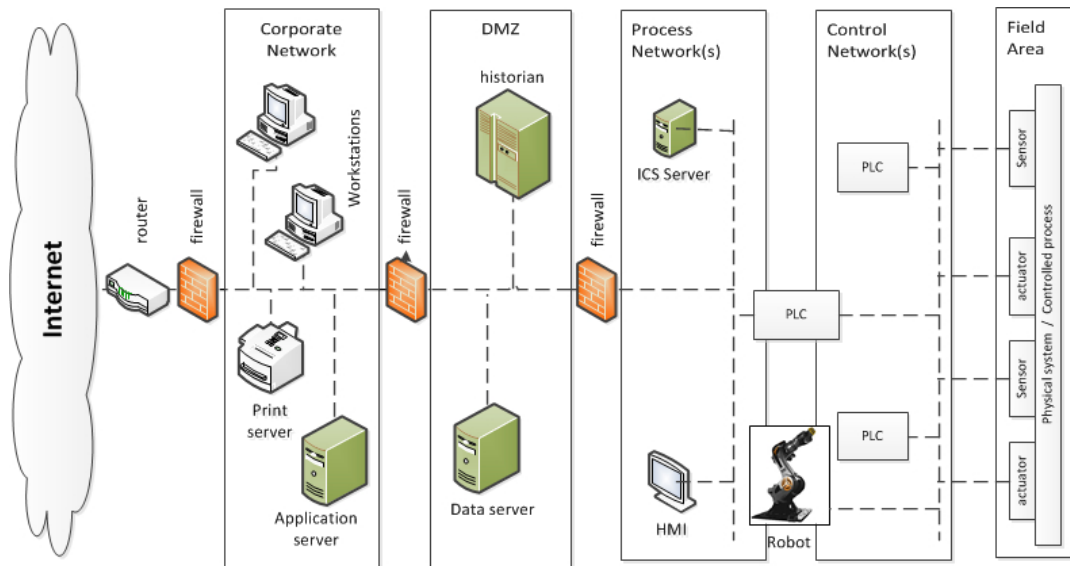


Figure 2.4: Industrial control system network

Field Area: The field area consists of those components that directly interface between the control system and the physical process it aims to control. These are components such as sensors and actuators.

Control Network(s): The control network facilitates communication between controllers (e.g. PLCs) and the equipment responsible for measuring and changing physical equipment. In some instances they take direct control decisions while in others they send information to the process network where a larger overview of the physical state is known.

Process Network(s): The process network is that part of the network which facilitates communication between controllers (e.g. PLCs) and larger control systems, including human machine interfaces (HMI), operator workstations and engineering workstations.

Demilitarised Zone: In computer security, a demilitarized zone (DMZ) or perimeter network is a network area (a subnetwork) that sits between an internal network and an external network. The point of a DMZ is that connections from the internal and the external network to the DMZ are permitted, whereas connections from the DMZ are only permitted to the external network – hosts in the DMZ may not connect to the internal network. This allows the DMZ's hosts to provide services to the external network while protecting the internal network in case intruders compromise a host in the DMZ. For someone on the external network who wants to illegally connect to the internal network, the DMZ is a dead end.

Corporate Network: The corporate network is a networked group of devices (such as computers, printers, routers) which are all owned and run by the same corporation, providing various office, business and engineering functions and are typically accessed by individual users.

2.2.4. COMMUNICATION PROTOCOLS

Communication between networked devices is established through communication standards and protocols. A communication protocol is, in essence, a set of rules for the exchange of data between two nodes. ICS architectures make use of the OPC⁴ standard for industrial automation, which is maintained by the OPC Foundation. Communication between servers and PLCs make use of either open or legacy (proprietary) protocols defining message standards, where each message has an exact and known meaning. These protocols then define topics such as syntax, semantics and how synchronization of the communication happens. Because the specified behaviour is usually independent from its implementation protocols it can be implemented in both hardware as software. Some of these protocols remain industry-specific (e.g., BACnet [33] for

⁴OPC is an acronym for Object Linking and Embedding for Process Control

building automation; DNP3 [34] for power networks), whereas others are used in general deployment (e.g., Modbus [35], Profinet [36], IEC 60870 [37]).

Protocols within ICS are generally binary protocols, making them not readable by humans. This is opposed to text-based network protocols which are human readable (e.g. HTTP, SMTP). These binary protocols are commonly used in networked services because the approach is more compact compared to their text-based counterparts. It is noteworthy to mention that these binary protocols might resemble attack payloads as malware packets often contain binary fragments [38].

2.2.5. USERS

In a typical ICS environment two main (human) users can be characterised. These are the engineers and operators, which can be characterised as follows:

Engineers: The (control) engineers are responsible for the design, implementation and maintenance of the control logic used by the decision making process of the ICS. The control logic is based on control theory which utilises a certain input - obtained from sensor readings - to alter a process and move it towards a desired state.

Operators: The operators are those that are responsible for the day to day operations of an industrial plant. They maintain a watchful eye over the processes and maintain the ICS and equipment when required. When the security system alarm is triggered operators are often the first to respond and are responsible to get the process back to a safe and stable state. Simply put, these are the people behind the buttons.

2.2.6. DIFFERENCES BETWEEN ICT AND ICS SYSTEMS

On the surface traditional IT devices resemble those of ICS equipment, which to some extent is indeed correct. Nonetheless there are some key differences when ICS equipment is concerned. Traditional IT commonly summarizes their security requirements in the triple [4] - also known as *the CIA Triad*, a well-known model for security policy development - as availability, integrity and confidentiality. The definitions below are taken from [39] and are listed in order of importance in typical corporate IT networks.

1. **Confidentiality:** the guarantee that information is not made available or disclosed to unauthorized individuals, entities, or processes.
2. **Integrity:** the ability of safeguarding the accuracy and completeness of assets;
3. **Availability:** the ability of being accessible and usable upon demand by other entities;

For ICS the order of this list is completely different, confidentiality is often regarded as a non issue whereas integrity and availability are of utmost importance. There is a major difference between a sensor reporting 1 bar of pressure as opposed to 1000 bar pressure or that a signal is originating from a real sensor as opposed to a compromised HMI. As such control engineers are more likely familiar with *AIC* given these different priorities compared to computer science.

In addition to the difference in security requirements, there are several other key differences, which include the following:

- **Information vs. assets:** Arguable the most important differences between a traditional IT network and an ICS is the direct interaction that the ICS has with the physical world. Whereas the traditional IT focus lies mostly on some form of data management, ICS have to maintain and control physical process stability.
- **Constrained resources:** ICS components are often designed for one purpose, which hardly changes over time. As such their available resources (e.g. power, processor, memory) are limited, especially for legacy equipment. This makes it infeasible to add extended capabilities to such components. Examples are sophisticated encryption mechanisms or security protocols since these generally require a relative large amount of computing resources.
- **Component Lifetime:** IT components are designed with a limited lifespan of single digit years, usually 5 [13]. Systems deployed in industrial environments on the other hand are intended to operate for much longer, multiple decades in fact. Thus the requirements posed by each are completely different,

and so are the problems encountered. For example: What if the operating system passes its end of life (EOL) date? Will patches and updates available for the duration of the plant? What happens if a software vendor goes out of business?

A summary of differences between ICS and traditional IT environments is given in Table 2.1, which is adapted from [4, 39]. For the interested reader, a detailed description of the differences between ICS and traditional IT networks can be found in [40].

System Characteristics	ICS	Traditional IT
Number of users	low	high
Multi-vendor	limited	common
Lifetime (years)	15 to 20	3 to 5
Outage tolerance	low/none	medium/high
Delay tolerance	low (real-time)	medium/high
Maintenance and Upgrade		
Patching	rare	common
Unsupported soft/hardware	common	rare
Soft-/hardware releases	rare (small changes)	frequent (extensive changes)
Frequency soft/hardware update	very low	medium/high
Security Practices		
Security awareness	low (but rising)	high/very high
Availability of security expertise	low	high/very high
Adoption of security audits	rare	frequent
Real-time security checks	rare/unavailable	common
Security Countermeasures		
Use of Antivirus	rare/unavailable	common
Physical security	difficult in remote sites	high
Use of firewalls and IDSs	rare/unavailable	common

Table 2.1: Differences between traditional IT and ICS systems (based on [4, 39])

2.3. CHALLENGES SECURING INDUSTRIAL CONTROL SYSTEMS

When looking at securing industrial control systems one is protecting the control path, not the data as with traditional ICT systems. The primary concern is to have integrity of process data, not the protection of user accounts or data confidentiality. Unfortunately, with our current knowledge and understanding it is not possible to fully protect control systems while also maintaining their operability [41]. Nonetheless, there is plenty of potential that can improve the current state of cyber security of most of these systems. This section aims to cover some of the challenges faced in securing ICS systems from a cyber security perspective.

2.3.1. TRADITIONAL ICT SOLUTIONS NOT ALWAYS APPLICABLE

On the surface, the IT systems look much like control systems, and in a lot of instances there is plenty of overlap. Nonetheless there are some important differences, as discussed earlier in section 2.2.6. These differences make it difficult to apply IT solutions to an ICS network. Although modification has been successful in some instances, in others the IT solutions will simply not solve the problem at hand. This can be seen in a range of ICS incidents in which traditional IT security approaches would not have prevented the event from occurring [41]. Simply due to the major differences between systems and requirements. The lower part of table 2.1 where security countermeasures and the differences between platform are listed demonstrates just this.

2.3.2. LEGACY SYSTEMS

Control systems are often designed with a life span in mind that far exceeds that of traditional IT equipment and systems. This can be seen by the fact that some control systems still running today were engineered over 20 years ago, when the internet was far from incorporated into every aspect of modern life. Within the offshore and dredging environment this is no different as the average design life for ships and platforms exceeds 25 years. Some platforms in the North Sea, for example, have been there for more than 40 years. This also applies to software which is used much longer than expected, a good example is the Y2K bug [42]. Back

then it was assumed that systems would be isolated and remain unconnected. Instead they became part of the hyperconnected world of the present day and age, still operating under the same old security principles while at the same time the information technology industry has been advancing at an incredible rate.

Unfortunately this problem is not only applicable to the legacy systems still in operation today. The modern and state of design platforms and ships of today all contain systems which will one day run into the same challenges. Not only because no one knows how much technology will transform the world within the next 10 or 20 years, but because this is often not part of the design of the control and security systems.

2.3.3. PATCHING AND UPDATING

Security of both IT and ICS systems is not a one time investment which can then be ticked off the list. Cyber security is a continuous process which requires more than determination of the mean time between hardware failures and keeping spare parts available throughout a system's lifetime. This is especially true within our fast paced and ever changing world. Vendors of both hardware and software publish (firmware) updates and patches to fix uncovered vulnerabilities and problems. The idea is that these should be applied to increase system security and often also improve performance. Due to the large variation of network infrastructure designs and configurations it is impossible for vendors to guarantee there will be no compatibility issues though that can take down a complete network.

Traditionally, in IT environments a (virtual) lab environment is created to ensure system stability after updating the systems. Unfortunately this is simply not a feasible option for most control system environments due to their high complexity, and the costs involved by creating a separate testing environment [43]. Even in those instances where there is access to a testing environment, there is no guarantee that no problems will occur in the real systems. There are even examples where completely unrelated parts of the infrastructure failed after applying a thoroughly tested update. This leads to a lot of operators choosing not to install any updates and accept the vulnerabilities that result from this. These vulnerabilities are then amplified by the increasingly connected world. As Cardenas [44] put it, "patching and frequent updates are not well suited for control systems".

2.3.4. CONTROL PROTOCOLS

Communication between the various components in industrial control systems is done using control protocols. There is a large variety in protocols available, some of which are openly available while others are proprietary. Due to this diversity, sometimes even within the same environment, security solutions that have been engineered for one protocol will likely not work for another. The solution can be adapted to incorporate multiple protocols, but with the number of protocols available and the fact that some are proprietary makes this a challenging if not unlikely endeavour.

This range of protocols also causes a major problem when dealing with research into cyber security defences for control networks, such as intrusion detection systems. These approaches often require access to and understanding of the underlying protocols available for them to be used in practice. Solutions working for only one very specific protocol are unlikely to make it to actual environments.

2.3.5. COMMERCIAL OFF THE SHELF HARDWARE AND SOFTWARE

Commercially off-the-shelf (COTS) hardware and software are readily available products that can be bought and used directly. COTS are designed to be easily used with and added to existing systems, without the need for much customization. Microsoft Office is one such a product for businesses. These products can also be found increasingly within industrial control systems which lowers both implementation and operational costs on the organization. However, due to their prevalence, COTS solutions are a tempting target for attackers, not only because they are used so generally and throughout various industries but also because there is a lot of knowledge on vulnerabilities and exploits for these systems. As such, they can pose a significant risk to companies and contribute to a lower security of ICS networks, especially when coupled with a lack of patch management policies. Additionally the COTS devices themselves can be a direct cause for concern as was demonstrated when an undisclosed ICS vendor explicitly put out a message not to install the latest Microsoft Windows update because it would interfere with Vendor modifications to the operating system and consequently crash control systems [41].

While it can be argued that using customized products will thus provide a better security ecosystem this is not necessarily true. COTS products are often tested, either by the company or by the public, for vulnerabilities and there is more budget to implement security measures. This is not often the case for customized products which can leave them vulnerable to undisclosed vulnerabilities.

2.3.6. MALWARE

Malware running on traditional IT networks can be annoying at best, but can just as likely cause significant problems and impact on business operations. Within ICS environments this is much the same, with the added difference that the consequences can be significant in that the physical processes can be affected. This is even the case when the malware was originally designed for an IT environment. This can happen because of the overlap in hardware and software, especially such as the earlier mentioned COTS devices. That this is not simply theory can be seen in the case where offshore platforms were shut down due to offshore personnel accidentally bringing malware aboard through a malicious download, infected laptop or data carrier. When malware is specifically engineered to target ICS environments the effects can potentially be much worse, as was demonstrated by the Stuxnet attack on the Natanz nuclear facility in Iran [16]. The malware demonstrated undeniably the viability of a digital attack against physical systems [29]. Stuxnet will be discussed in more detail in subsection 4.2.2

Various researchers have both demonstrated and published on the security vulnerabilities present in control environments. Unfortunately this is not the case for effects and range of malware that manages to install itself on control systems, where only limited research is available. In an effort to bolster knowledge in this specific area [26] injected four (ICT) malware samples in a testbed, which resulted in: restarting of the SCADA servers; running of arbitrary code on infected systems; shut down of field control operations. In addition to ICT malware, [26] developed three samples of ICS malware and injected these into the ICS testbed. The results of this demonstrated that the malware was able to control the system as if it was the control system itself.

Keeping malware from gaining access to ICS networks is an important aspect of control system cyber security and can form a major challenge.

3

CYBER SECURITY IN OFFSHORE AND DREDGING

Starting this work little information was available on cyber security and how it relates to the dredging and offshore industries. Awareness on the importance of cyber security, and how it affects the offshore, maritime and connected industries was low to non existent. Operating outside of high-tech computer science and information technology kept the industry seemingly unaffected to the effects of cyber events.

Discussions with experts from the Offshore and Dredging Engineering department at the Delft University of Technology, the cyber security group at the TU Delft, as also the offshore construction firm *Heerema Fabrication Group* and the Central Dredging Association confirmed this image. Which is unfortunate. The world is increasingly becoming interconnected[5], which not only affects the information flowing through organisations but also the physical world. Within the dredging and offshore industries this is especially true, as physical processes are often at the core of business operations.

This increased interconnectivity makes cyber security an important area of research. Not only from a technical perspective but also includes a major psychological component, the people side if you will. Major improvements can often be made by simply making people aware of the importance of cyber security and the effect a single person can have on it. The aim of this chapter is to provide insights into the current state of cyber security in the industry such that steps towards improvements can be made. The main intended audience are industry professionals without much security knowledge and awareness. Those operating in this field are assumed to have a better feeling for the current state of affairs and its importance.

Information used to indicate the state of play has been collected mainly through the following two channels:

1. Desk research: The first part on which this chapter is build is information which was retrieved through various desk research channels. Initially previous research into the area was to be used. Unfortunately this quickly proved to be futile as little to no research into cyber security in dredging and/or offshore is available. To remedy this the range of acceptable resources has been expanded to include alternative sources such as white papers, articles, presentations by industry professionals and even an "Ask me Anything" event. Interestingly this lack of publicly available information was also mentioned during an expert interview[13], whom guesstimated the amount of publicly available information could be as low as 25% of all existing information.

It should be noted that using this alternative information might affect credibility. News articles are simply not the same as peer reviewed research. The author feels however that the aim of this chapter, gaining a better understanding and impression of the current state of affairs, leads itself to stepping outside of the literature restrictions. Especially where the credibility of the source is taken into careful consideration.

2. Interviews: The second part on which this chapter is build are interviews with professionals operating within the industry in which they were asked to share their views and experiences on this subject. However, contacting an adequate number of experts related to the dredging and offshore field turned

out to be difficult for two reasons. First is the observation of a security-by-obscurity attitude¹ leading to a lower response rate on interview requests. Secondly the author should have demonstrated a much more active and aggressive approach in pursuing possible interviewees.

First thing to be discussed in this chapter is the fact that cyber incidents are indeed on the rise in section 3.1. Due to the importance of industrial control systems (ICS) within the dredging and offshore industries the next section, 3.3, discusses the current state of cyber security in ICS specifically. Awareness is an important aspect when improving cyber security, as such section 3.2 discusses the state of cyber security awareness. Often people are sceptical on the importance of cyber security, section 3.4 presents an illustration that threats are real by discussing documented incidents and research. The last section is 3.5 which discusses the results, steps taken to improve the current state of play, and recommendations to take this further.

3.1. CYBER INCIDENTS ARE ON THE RISE

A recent survey amongst security professionals indicates that cyber threats are on the rise [27]. Attacks specifically targeting the oil and gas sector should come as no surprise and are proving to be a rising concern for the industry and government[29, 45] alike. These incidents extend beyond the traditional business network and into the control system infrastructure[19, 27] enabling much of the offshore operations. Unfortunately, these -often critical- systems are a major risk factor[9]. Fortunately security breaches into industrial control systems that have resulted in damage and/or human injury are scarce, although not completely absent[19]. Getting accurate and detailed information on actual incidents is difficult however. Cyber incidents are seldom reported[11] making case studies difficult to find, at the same time incidents are often incorrectly identified and not classified as a cyber incident [41] obfuscating possible sources. This makes actual statistical information difficult to acquire and those that are available thus only present a lower bound estimate.

This difficulty to acquire accurate information is also encountered within the dredging community. There is no literature or ongoing research to be found that investigates the state of cyber security of the dredging industry. Nor is there information (publicly) available on actual cyber attacks/incidents. Making things more difficult, those parts of the community do not seem to commonly share incident information with the public, researchers, or the community itself. The author suggest two reasons for this. First is that it would seem that most adhere to the philosophy of "security-by-obscurity", which means that industrial control system operators rely on secrecy to ensure security. The effectiveness of this approach is debatable and doubtful for interconnected networks [46]. Second is that companies shy away from the negative spotlight and will not expose themselves when they are hacked. Discussions with industry experts have indicated, however, that cyber incidents do occur, both as a result of Malware finding a way onto systems, as caused by engineering mistakes or material failure. Interestingly security researcher Robert Lee stated in an interview:

"From everything I've seen and experienced I would say that there is some level of compromise going on in most large corporations in the world. The forensics and incident response community has looked through its metrics and case studies to determine that well over half of the companies compromised do not know that they are compromised and many will take over a year to detect it."

- Robert M Lee[47].

Clearly cyber incidents are occurring on both IT and ICS networks and simply ignoring this issue will not make the problem disappear[48, 49]. It is important to realise that the impact of security incidents, especially where physical systems are involved, can be dramatic, ranging from simple failures or loss of business continuity to the loss of lives. Section 3.4 offers the reader more insight into these alarming risks.

A note on breaches and incidents It might seem that cyber incidents are the cause of some actor trying to gain access to systems for their own (nefarious) reasons. While this can indeed be true, not all cyber incidents are due to some malicious actor. There are many incidents which are due to the malfunctioning of software or hardware which was not accounted for. This malfunctioning can be anything, ranging from programming and logic errors, failing to test the interconnection of systems or even simply because the hardware fails. Cyber incidents are often incorrectly diagnosed as not being cyber for various reasons, but even when they are correctly attributed they are often not of malicious intent.

¹Security-by-obscurity is a security practice where it is assumed that because system protocols and details are unknown the to the world, it must be secure

3.2. CYBER SECURITY AWARENESS

Arguably one of the most important aspects of a successful cyber security strategy is (employee) awareness. The main argument for this is that most attacks have gained traction through unsuspecting people that have access to the targeted network. Those responsible for resources and business operations should be properly informed on the risks and potential impact on business continuity such that they can set the proper policies and provision adequate resources. In general people have no malicious intent and are simply doing their jobs, if kept uninformed they can hardly be expected to be aware, let alone responsible, for the impact their actions and decisions might have.

The key findings of a 2011 research on cyber security aspects in the maritime sector[9] is that the level of awareness and focus on maritime cyber security is low to non-existent. This observation is applicable at all layers in the industry, including: contractors, port authorities, and governmental bodies, which should be cause for concern. One of the possible explanations for this is argued to be the low number of (publicly) known cyber security incidents within the sector. This seems to agree with the findings in the previous paragraph, there is little to no publicity on this topic, nor is there a platform in place to do so. In case such a platform would be available the question remains if it would be utilised at all [50]. This then makes incidents appear few and limited in their reach[51], and suggests that this sustains the troublesome observation that overall perception within the communities on the risks and the impact of cyber incidents are limited.

For most organisations turning a profit is a prime priority. In the traditional industries of dredging and offshore the importance of computer systems may be misunderstood and some go as far to view them as nothing more than a nuisance with a negative impact on the bottom dollar, as opposed to further improving profit margins. Which applies even more so to cyber security. This trend is also seen in the closely related industry of industry control systems, where the maturity level² is often low to non-existent. The bare minimum security requirements equal a maturity level of 3. Currently this can be considered as being ahead of the pack within the ICS industry[13]. With the increased dependency on IT systems for all key aspects, processes and activities this represents a serious reason for concern. Fortunately there are indications that sounds of questions and critique are heard and rising within some organisations. Especially amongst the lower ranks [50].

A SANS research[53] surveying cyber security awareness in no specific industry uncovered three key findings:

- **Support is Essential:** Preventing security awareness programs from failing will only succeed once they get the same emphasis and support as technical controls. Senior leadership should be educated that security also includes the human element and is more than technical tools alone.
- **Soft skills are lacking:** The survey states that more than 75% of awareness programs surveyed are run by people with highly technical backgrounds but with little experience in softer skills such as communications, change management or human behaviour. This makes them prone to view the world through a strictly technical lens and while failing to account for the human factor.
- **Security awareness is still in its infancy:** Organisations are in the beginning stages when it comes to embodying a secure culture. One of the main challenges will be making people conscious they are targets and their actions can impact the security of the organization. To effectively change behavior this is paramount.

An example demonstrating the effects from this lack of awareness is shown by researchers investigating how little effort/resources it takes to get users to run unknown programs on their systems. The researchers concluded that "users are generally unopposed to running programs of unknown provenance, so long as their incentives exceed their inconvenience"[54]. Employees that are security aware are more likely to not only refuse such an offer, they are also more inclined to inform the security department of fishy behaviour and requests.

Ignorance might be bliss, but simply being unaware of the importance of cyber security and possible threats does not make the challenge disappear. On the contrary, it will more likely worsen the status quo. Creating awareness will not introduce perfect security, no solution will, but it will have a big -if not the biggest- impact on the current state of affairs. Additionally, when awareness of the importance and current situation is on the table, one becomes empowered to improve and to start monitoring, changing, improving and identifying other weaknesses.

²The maturity level rates a company on their dealing with cyber security, usually on a scale of 1 through 5 [52].

3.3. INDUSTRIAL CONTROL SYSTEMS

Industrial control systems are used to control (physical) processes and can be found in many industries ranging from -but not limited to- oil and gas, power generation, to maritime and building automation. A key finding by ENISA in their analysis of Cyber Security Aspects in the Maritime Sector [9] is that there is a direct relation between industrial control systems and cyber security. Not only are these systems found throughout many industries, but are found at the heart of many operations.

Traditionally networks which connected the ICS components were part of a private local area network (LAN) and stayed isolated from external systems. From a cyber security perspective these systems used to be seen as secure simply due to their isolation, which is known in the industry as an airgap. Additionally most information on these systems remained under tight control and little was publicly revealed, which is known as security-by-obscurity.

The present time has caught up with these isolated networks and modern technology has these control system changing in various ways. Three main trends can be observed in more modern designs[4]:

1. **Increase connectivity:** Industrial control systems are following the global trend and are becoming more interconnected. This allows direct communication with the corporate networks which causes increased productivity and decreased costs. This usually means that the control network is connected to the internet through which a connection to the corporate network is established.
2. **Commercial of the shelf solutions:** It is increasingly common to make use of commercially off the shelf devices and solutions (COTS). The well known Windows desktop is one example of this. Using these solutions not only offer a reduction in purchase and maintenance costs due to there abundant availability, but also means that employees are likely already accustomed to their used and thus require much less training to operate the solutions.
3. **TCP/IP adaption:** Modern ICS designs often make use of TCP/IP³, which is used by most commonly known networked devices. The control protocols then build on top of TCP/IP to support the control network. This in effect can lower the costs and make maintenance easier as only one type of wire has to be run through the site, which is then in turn used by all devices that require an connection, be it part of the control system or local workstations.

These trends have obvious benefits, but they also expose the ICS systems to the same threats faced by IT systems making them increasingly vulnerable to cyber attacks and incidents. This is even more so for legacy systems which where never designed to be connected to the internet -which simply did not exist back then- in the first place.

Changing trends are not the only problem faced ICS's. The protocols used by these systems are frustratingly insecure as they are designed for easy debugging and maintenance but not security. Security researchers new to control systems often focus on various exploits without being aware that by design a controller can actually crash by simply scanning them. These protocols are designed and meant to be used in an already secure network, but unfortunately there is no such thing.

If one can draw one conclusion from this it is that ICS cyber security is relatively new and said to be in its infancy. It is currently in the place where IT cyber security was 10 to 15 years ago. Which is as much technical as cultural. Responsiveness from ICS vendor is generally bad and a huge lack of focus on control system forensics exists[41]. This often stems from the fact that generally there is a lack of knowledge and understanding at the executive level where security is hardly a thought at all and is often simply viewed as an a drain on resources that belongs to the IT department[41]. If security would be seen as the risk management that it is, along with the negative impact on a businesses bottom dollar people would be much more moved to secure their systems. Convincing management of the importance of the topic is not the only challenge though, from a research perspective there are a wealth of challenges in securing ICS, some of which have been discussed in section 2.3. Interest from researchers looking into this field is increasing though, and some have indeed started their research already. The field can learn a lot by looking at the steps taken by IT security and should take considerably less time to come to maturity.

Buying into the Hype Corporation XYZ was compromised by an advanced threat which they couldn't defend against and the industry is under assault by advanced actors, sounds that these days are often heard in the grapevine, board rooms, and the media. But what is an *advanced* if anything but relative to the observer?

³Transmission Control Protocol / Internet Protocol

Often it provides a more catching news heading or briefing to stake holders to mention some advanced actors compromising systems, opposed to their compromise by 20 year old techniques that could have been prevented. During a 2014 interview security researcher stated: "Everyone has heard the word 'Stuxnet' more times than they can count, and likely much more than they would like," [47].

Threats to control systems When discussing threats to control systems the focus quickly turns to examples of advanced actors, such as Stuxnet. Stuxnet is a malware that required a nation State employing a team of expert engineers, analysts and test environments to create [10, 16, 28]. The scale of this threat has not prior been seen by the security community and hasn't since. Nevertheless defending exactly this type of threat seems to be the only focus exhibited by some[47]. It is true that there are a lot of lessons that can be learned, but focus should not be lost on incidents that are much more commonplace.

A cyber incident within a control systems does not have to be triggered by any malicious actors, and simply observing negative effects will not implicate the one or the other. Often the only differentiating factor between malicious and non-malicious is the intent of the actor. A simple mistake, or well intended improvement by a control operator can have a serious and malicious looking impact. Even in those cases where the focus is on malicious actor the fear should not be from a random hacker stumbling her way into a control system, but instead should be aimed at those with actual control system knowledge[16].

The main problem at the moment is that the bar on security should be raised to the point where access to critical systems cannot be gained by exploiting ancient exploits or poorly configured networks. Implementing expensive and state of the art solutions will be pointless if the overall security body is failing and the front door to these systems remains open to whomever bothers to look.

Knowledgeable Experts Control system engineering and cyber security are highly complex technical fields which have knowledgeable and passionate experts in their own right. ICS cyber security can be seen as the overlapping, or even combination, of these fields which requires technical understanding and know-how on both fields. This makes it a highly complex field in itself.

Many present themselves as experts in the ICS security field for various reasons. This makes it difficult to find authentic and passionate experts with knowledge on both sides of the coin [47]. World wide there might be as little as a few hundred knowledgeable experts [41]. The situation is represented graphically in figure 3.1, taken from the presentation given by Joe Weiss at Stanford in 2012[41].

Existing educational facilities seem to do little to combat this problem as colleges, universities, and academia focus either into cyber security or control engineering but there is hardly any overlap and a specific path into industrial control system cyber security is yet to be established. Offering courses from opposing fields into their curricula could alleviate the problems at least to some extend.

Failure testing Within the field of dredging and offshore engineering designs are based on many failure and hazard analyses. These are not only requested by clients, but often mandatory by both insurance agencies and governmental bodies. Failure is an intricate part of the design process. Interestingly this is often not the case for the security of both IT and ICS systems.

These systems hardly -if ever- undergo comprehensive testing and evaluation under various scenarios such as virus infections or denial of services. Often they are simply tested to meet functional requirements and tested for this functionality, but the resilience of these systems under abnormal or unexpected circumstance is not part the testing requirements or even the basic design[11]. Nonetheless the consequences of these systems failing under certain circumstances can be much more serious than the failure of a specific process.

A note on industrial control systems It is important to mention that when a trespasser gains access to an industrial control systems, say because it is connected with a default password to the internet, does not automatically translate to the ability to physically destroy equipment at will. In reality this takes a lot of effort and combined knowledge on the attacks part[55]. Although the trespasser can still (even accidentally) cause serious problems, and this has indeed happened[56] as the vulnerability and potential for misuse is there. The reader should be aware that while it remains difficult to cause widespread physical destruction for all but the most sophisticated actors currently, these capabilities will eventually diffuse to the wider public. In addition, unexpected spillover effects from unsophisticated attacks should not be ignored as these could have a more widespread impact than targeted attacks[19].

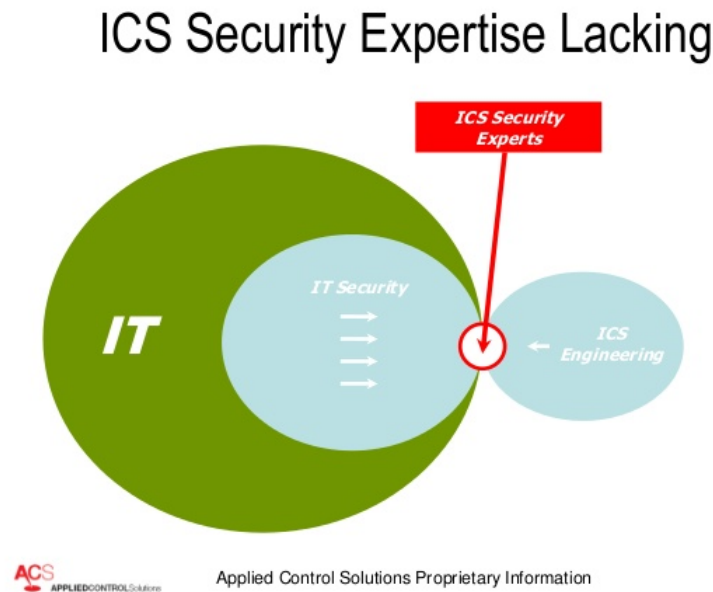


Figure 3.1: Source: Cyber Security of Industrial Control Systems by Joe Weiss[41]

3.4. CYBER SECURITY VULNERABILITIES AND INCIDENTS

Section 3.2 raised the challenge of cyber security awareness. One of the reasons for this was attributed to the low number of documented incidents available. This section aims to provide information on actual incidents and vulnerabilities which are reported. With this it will be shown that the often used argument "we must be secure as nothing is reported" is a fallacy. The section itself is divided in two: The first part involves the offshore and maritime world, whereas the second looks at the ICS industry which is much broader than the offshore and dredging industries.

3.4.1. OFFSHORE AND DREDGING RELATED

This subsection presents four cases of security vulnerabilities and incidents within the dredging and offshore industry.

Research I: "Hey Captain, Where's your Ship?"[57] demonstrates the potential for abuse in the automatic identification systems (AIS) used by ships and is required on any international vessel with a gross tonnage that exceeds 300 tons. AIS is a tracking system for ships used in various applications ranging from vessel traffic services, collision avoidance to maritime security and accident investigation. The researchers were able to create a ghost ("spoofed") vessel which followed a path over time in the AIS tracking systems, which in reality was nothing more than a script running on a computer. The result is demonstrated beautifully in figure 3.2. In addition multiple other problems with the AIS systems are found, this includes:

- Frequency hopping: The ability to disable AIS transponders, for up to 5 weeks. This would enable Pirates to render a ship invisible in Somalia for example.
- CPA Alerting: Faking a Closest Point of Approach (CPA) alert and trigger a collision warning.

If anything, the thing to take home from this research is that the widely used AIS is broken at both the implementation- and the protocol-level.

Research II: In the summer of 2013 assistant professor Todd Humphreys and a team of students successfully manoeuvred an 80 million yacht off its course by hundreds of meters, effectively demonstrating the potential to take control of a ships unmanned systems. The group was able to execute this attack by sending fake GPS signals (spoofing) towards the ships GPS antennas, eventually overpowering the authentic GPS signals and obtaining control of the ship's navigation system. Unlike the jamming of GPS signals this spoofing



Figure 3.2: Hey Captain, Where's your Ship?

triggers no alarms on the navigational equipment as the false signals are indistinguishable from authentic signals to the GPS devices. Humphreys said "The ship actually turned and we could all feel it, but the chart display and the crew saw only a straight line"[58]. This experiment demonstrates a problem that is not only applicable to seagoing platforms but for any in which GPS is used for position determination ranging from automotive to unmanned flying systems such as drones.

Malware on Offshore platforms: Early 2013 Malware got access to a number of offshore platforms after workers, unintentionally, loaded it to their workstations. This Malware then proceeded by taking over the systems and resources it found itself connected to, which eventually incapacitated the networks[25]. All this happened while the workers were out in the middle of the ocean. Fortunately, in this instance the consequences were not serious, however one can easily imagine the disastrous effects such an event might present for people and to the environment. Another, more serious problem arose when a new build drilling rig left its construction yard in 2010 and had its computer systems attacked by Malware. This went so far that even the systems controlling the blow-out preventer were found to be infected. This is the same device which malfunctioned during the Deepwater Horizon oil spill[59]. As the platform was in transit to its first assignment nothing happened, but the potential consequences of this simple Malware infection would have been very serious indeed if the platform should have started drilling while infected.

Shamoon: Shamoon, also known as W32.Distrack[60], is a Malware that targeted Saudi Arabian Oil Company (Saudi Aramco) in August 2012. The infection spread to as much as 30.000 personal computer inside the Aramco network, stealing files before overwriting them by an image of a burning American flag and wiping the MBR (Master Boot Record) [29, 61]. Some have indicated that having physical access to the network was required to insert the virus, which indicates the importance of proper cyber and physical security.

There are no public report that the attack caused an oil spill, explosion or other physical incident, nonetheless production and business continuity must have been effected due to the loss of (production) data and the significant downtime period for all public facing websites owned by the company[61]. Eventually they were able to restore all of the internal services and reported normal business was resumed. The total costs of this exercise can only be speculated. It is important to note that although Saudi Aramco was targeted, the infection spread beyond their network as other oil and gas firms were also infected.

The virus was referred to as "very sophisticated", and "raising tremendous concern about the potential

for the use of that kind of tool" by then CIA director Leon Panetta[61]. Although a virus infecting the systems of a multinational corporation is not an altogether alarming occurrence, incidents targeting companies that hold much of the world's spare oil production capacity is not to be neglected. Even partial disruption of their production facilities would directly effect oil supplies and prices, with a domino like effect on the global economy.

The Phantom Menace: A operation that targeted some ten companies in the oil and gas maritime transportation sector with the intention of stealing user credentials. The breach to security managed to fly under the radar for over six months and remained undetected by any security filter in place, simply by using legit tools and software as opposed to Malware. These tools were sent through a phishing email to specific users whom then continued to open the attachment, unwitting to their purpose. Once opened the attached file, which resembled a pdf, displayed a blank pdf which users quickly discarded thinking a mistake was made. In reality a cascade of scripts and tools were started that compromised the users, stole their credentials and uploaded them to a remote FTP server.

These credentials, it seems, were then used to pull a form of Nigerian scam. Criminals were selling a large batch of comparatively priced oil and presented buyers with Proof of Product papers that the offer was indeed legit. Once the buyers made an advance deposit for \$50,000 to \$100,000 the scam was revealed for what it was: There was no oil and although the Proof of Product documents were real, they had been stolen through the use of the credentials gained from the cyber attack. Interestingly, to date there have been no indictment or even investigations by law enforcement, simply because none of the victims are willing to file a report. Opting to fly under the radar, change credentials and act as nothing happened. Even though there is evidence pointing to a suspect. This case is one of many that indicates that companies should wake up to the fact that there is no absolute security and awareness on being breached[62].

3.4.2. INDUSTRIAL CONTROL SYSTEM RELATED

This subsection presents three cases of security vulnerabilities and incidents related to industrial control systems.

Maroochy shire The Maroochy Shire incident was an intentional and targeted attack executed by a knowledgeable person on the industrial control systems of sewage equipment for the Maroochy Shire Council. The incident caused a spill of 800,000 liters of raw sewage. A representative of the Australian Environmental Protection Agency said the "Marine life died, the creek water turned black and the stench was unbearable for residents," after the event took place [23].

Stuxnet The Stuxnet Malware became infamous when it demonstrated the power to influence the physical world through the industrial control systems of the Natanz Nuclear power plant. It is best known for causing failure of enrichment centrifuges by taking them through their resonance frequency, causing mechanical failure. Stuxnet can be seen as a textbook example on Cyber Warfare and is often credited with being the first cyber weapon by demonstrating the capability of software to impact the physical world. As such its impact for the future is said to be substantial.

This paragraph only gave a short summary on Stuxnet because section 4.2.2 elaborates on Stuxnet in much more detail than would be applicable here.

German Steelworks facility In their annual findings report of 2014, "Die Lage der IT-Sicherheit in Deutschland 2014" [56], the Bundesamt Für Sicherheit in der Informationstechnik (BSI) makes mention of an attack on industrial installations, specifically a steel mill, in Germany. Using social engineering attacks in the form of spear phishing e-mails targeting operators of the industrial installation initial access to the steel mill's office network was gained. From there the actors progressively found their way into production networks, details on the applied techniques are not specified. The impact of this was "the breakdown of individual control components which led to the uncontrolled shut-down of a blast furnace, leaving it in an undefined state and resulting in massive damage". The technical expertise extended beyond that of traditional IT security to include detailed technical knowledge of the ICS and production processes employed by the mill. With this publication the BSI supplied the only other publicly available example of a cyber attack causing physical damage through control systems – The other being Stuxnet. This sharing of incidents is considered to be ex-

tremely valuable to deriving lessons learned and best-practices for defence and incident response. More so with the general resistance to share anything related to cyber incidents.

3.5. DISCUSSION AND RECOMMENDATIONS

This chapter has made a case that cyber incidents are both real and on the rise. With the increase of interconnectivity between valuable resources in the dredging and offshore industries the discussion on the significance of cyber security in the industry is increasingly important. This is especially true when comparing the current general state of cyber security on the maturity scale, where the present conclusion would be that the industry is residing in it's infancy. Cyber security is often seen as a necessary evil that is subservient to the ICT department, where the focus is mainly on prevention.

Lacking awareness and understanding of cyber security amongst industry professionals is one of the reasons behind the current state. Creating awareness by educating and informing people is thus an important aspect in enhancing security, one that can potentially yield large returns on investment. Security is much the same as safety in that respect: A group effort that should become part of a culture and mindset. Achieving this is a different matter though, getting people on board is difficult as they often don't understand why they should care in the first place. To them nothing seems to be amiss. This could clearly be seen during a recent event on cyber security in the dredging industry -aimed to improve and discuss the security challenges in the community- where some simply ignored much of the arguments and examples. Seeing them as nothing more than scaremongering. While this might be true to some extent, oftend aimed to get people to act, the fact remains that the threats are facts. This demonstrates the importance of awareness and the need for open discussions to take place. However, changing a culture takes more than technical understanding and a few presentations. Knowledge on social interactions are equally important as is research into human behaviour. These problems mimic the challenges posed when developing organisational safety cultures during the early offshore oil and gas periods[63, 64]. Lessons from this past should be used to help create a safer and more secure future.

Fortunately there are sounds from the industry that indicate that efforts are under way to further increase cyber security within the offshore/maritime industry. The research by ENISA[9] is seen as one of these. Another the recently proposed cyber guidelines by the international shipping organisations[65]. Shipping organisations and governmental bodies around the world are slowly awakening and putting cyber security on the top of there agenda.

3.5.1. RECOMMENDATIONS

Cyber security in the dredging and offshore industry leaves room for many improvements. Transitioning into a mature situation is challenging but with the rise of incidents one that should not be delayed or ignored. This section provides recommendations to enable a move into this direction. These have been split into three paragraphs: Awareness Education, Maturity Level, Sharing is Caring.

Awareness and Education: Those whom understand the importance of cyber security will be more supportive towards implementation of effective security processes, they might even rally behind this goal and ask for more secure working environments by informing colleagues and management. Tackling the cyber security awareness challenge is a tough one, be it within organizations or industry wide, but nonetheless worth the effort since much ground can be improved this way. As such it is important to keep the pressure on and inform those around us, running awareness events and training sessions.

Another way to increase awareness is through the education systems, if students get educated on cyber security from the start this will benefit the community in the long run. Having informed students entering the market will help spread industry awareness, help produce higher quality products and make our systems more secure. This trend has also been visible with safety, Offshore and Dredging engineering at the Delft University of Technology has safety as part of the curriculum. In this respect there is a lot that educational facilities, such as the TU Delft, can do to further along the cause.

Maturity Level: The majority of the companies have a cyber security policy that is lacking, if one is present at all. It is recommended that the industry promotes those that are self aware in this aspect and work towards self improvement, while stimulating others to follow and do the same. This can be achieved through the use of maturity levels which indicate the current status of an organizations cyber security policies. As with the

accident frequency rate (AFR) a demand on adequate cyber security would drastically improve the current state of affairs as it directly affects the turnover of (sub)contractors.

Sharing is caring: The lack of available information with respect to cyber incidents, be it with malicious intent or without, is used as an argument that organizations are secure and that spending resources on cyber security is nothing more than scaremongering with a negative effect on the bottom dollar. As mentioned previously it is estimated that half of those with breaches to security are not even aware of them. With those that are taking over a year on average to detect malicious intent in the first place. It comes down to the fact that sharing is caring for the industry generally. This same conclusion is also drawn in the recently released report on actors targeting oil tankers, where prosecutors are unable to charge the perpetrator simply because no company is willing to step forward. Would those that are affected by a cyber security incidents come forward and share information it is possible to have an open discussion about the current happenings and improve security of the community as a whole. The positive effects of this approach have been demonstrated by e.g. the banking world, which collaborate within the digital domain to increase security by sharing breaches, experiences and technology amongst those in the community.

In addition to sharing information with the industry there are major benefits to be had by cross-fertilisation from other sectors, such as that of industrial control systems or energy production for example. The offshore industry is not the only one that is heavily dependant on computer systems and inventing the wheel in each and every sector (or organisation) is a complete waste of resources. The author feels that communities should be working together, pooling resources and seeing the benefit in shared knowledge and differing insights.

3.5.2. SECURITY INITIATIVES

Various initiatives have been launched throughout this thesis to follow up on some of the given recommendations. These have started the process of change and improvement while providing future endeavours a platform to continue from.

Heerema Fabrication Group: At the offshore construction company Heerema Fabrication Group (HFG) a presentation has been presented to the board of directors on cyber security, why it is important for business continuity and additional benefits it would offer. This was followed by the starting of a security initiative where HFG ICT security policies were polished and created, while additionally meeting with individuals interested in furthering the cause by participating in a discussion / work group. Finally efforts were made to incorporate a security focus into a major ongoing business restructuring project: "Back 2 Basics".

Dredging Community: To move the dredging industry forward an cyber security event has been hosted during an CEDA-NL⁴ club evening. The focus for this event were three presentations which aimed to bring the discussion to the dredging community and get a momentum going. The main conclusion from this event is that collaboration is paramount and the industry has to work together.

In addition to the event a proposal was made to the CEDA Board to establish a working group on cyber security within the industry. This working group could then focus on increasing awareness and collecting metrics to more accurately reflect the current state of cyber security within the dredging industry. This proposal led the board to request a presentation on cyber security awareness and to elaborate the proposal itself, after which a decision would be made. This has led to the following:

- A request to present a CEDA Webinar on the importance of cyber security, aimed at increasing awareness.
- Collaborate on the organisation of an event for industry professionals to gauge interest and thoughts on the need for a working group.

⁴Central Dredging Association - Dutch section

4

MALICIOUS SOFTWARE

The world is increasingly becoming hyper-connected, bringing a range of benefits to our society but at the same time making us dependant upon their safe and continuous operation. Unfortunately a rise in the number of cyber security incidents can also be seen, as was learned in the previous chapter. At the center of many of these incidents is malicious software, also known as malware. This makes research on and increased understanding of, malware an important topic, and extending beyond those working in the cyber security sector. This is especially true when talking about industrial environments and the potential challenges malware poses to operators.

While malware has a range of applications, the best known is arguable disruption to normal operations. This can cause annoying problems on home systems by encrypting all a users files or even taking down corporate networks. It has long been considered that while malware is known to attack IT systems, there is potential to have an serious impact on ICS systems also. This fear was proven true when the Stuxnet malware came to light in 2011.

This chapter investigates malicious software and presents two case studies of well documented incidents, one being the ICS malware: Stuxnet. Also the current knowledge and understanding of the impact that malware can have on ICS systems is investigated. To increase awareness on the potential impact of malware, outside of the security community, a prototype malware has been designed to be used as a demonstration at presentations and events.

4.1. BACKGROUND

Malicious software, commonly known as malware, are programs which are defined by their malicious intent. They act against the requirements of the user and are used in the disruption of computer operations, gathering of sensitive information and even to provide access to systems for nefarious actors. Thus software causing unintentional harm due to some imperfection does not fall under this definition.

Often credited with the discovery of viruses, a specific type of malware, is Cohen[66]. The truth is though that he gave name to phenomenon that were already known and referenced. Common behaviour displayed by viruses go back to at least 1970[12]. Still accurately describing vulnerabilities and threats to be found today is research by Ware [67] and Anderson [68], both published in the early 1970's. Since those early days of computing many things have changed in the field though, one such example is the sheer number of distinct and unique pieces of malicious code found throughout various networks and systems. Another is that new, and adapted, malicious code is appearing in the wild at an increasing pace[12].

The most intensively studied, and recognized as the earliest virus, is thought to be the *Brain virus*. Its displayed behaviour was to change the label of any disk it attacked to the word "BRAIN", from which the virus received its name. Other than spreading itself Brain exhibits no malicious behaviour though, which is inline with much of the early days during which malware was created more as a demonstration of skill then anything else.

Over the years the goal and use of malware has evolved. Where previously it was mostly teenagers showing curiosity, these days organised crime runs elaborate schemes involving various kinds of malicious software to increase their illicit income. A perfect strategy as government and law enforcement are only recently starting to catch up with the technology used. "Cybercrime is constantly evolving and becoming more mainstream,"

says DCI Terry Wilson of Scotland Yard's e-Crime Unit. malware is being sold for many purposes to various individuals. From simple online nuisances to spammers, fraudsters and now even taking the form of cyber-weapons.

Propagation In essence there are two modes of propagation used by Malicious software. The first is through infection of other software files; malware employing this tactic is commonly called a Virus. Secondly there is the spreading through network tunnelling¹; malware which employs this approach is commonly called a Worm. Thus these two versions can then be classified as follows:

- A *virus*, named after its biological counterpart due the similarity in which the infected program behaves, replicates itself by changing other software and infecting them. This infection is possible because the virus is able to attach itself to other software and then either destroying or coexisting with it. A virus requires user action to get started, which it usually accomplishes by being an innocuous part of other useful software. Due to its insidious behaviour it is impossible to assume that a clean program today will still be clean tomorrow.
- A *Worm* is software that spreads itself through networks by creating copies, behaviour which was first described for non malicious goals by [69]. Its main difference with a virus is that the mode of operation is through a network, where a virus can use any medium.

Utility Malicious software comes in many shapes and sizes, as it does purposes. Three elementary types which can be identified will be discussed next: A Trojan horse, a logic bomb and a backdoor.

- A *trojan horse* takes its name from the Greek tale where during the trojan war the Greeks constructed a huge wooden horse, and hid men inside of it. The Greeks then pretended to sail away, and the Trojans pulled the horse into their city as a victory trophy. At night the Greek force crept out of the horse and opened the gates for the rest of the Greek army that was waiting. This program acts in the same way, seemingly being a useful tool, while at the same time acting out its malicious intent. An example would be Ransomware [70–72] which encrypts all files on an engineering workstation and connected drives, effectively holding data captive until a ransom has been paid.
- A *backdoor* is an hidden, or unlisted, access point into a program to those that have knowledge of it. Its important to be aware though that even those without knowledge might find the backdoor and thus gain entrance, even if unintended. An example would be that someone entering the phrase "Open Sesame" as user name is not required to supply a password to gain access to human machine interface (HMI). Backdoors can be put in place intentionally –for maintenance purposes as example. It can on the other hand also offer an illicit way to wipe out records of a crime committed by basically allowing unauthorized access to a resource.
- A *logic bomb* is a piece of software that is coded to detonate when a specific logic condition has been met. This can be anything from a specific time or date to a command sent by a hacker or command and control centre. What will happen once the logic bomb detonates will depend on its payload entirely.

Virus infection strategies As mentioned, viruses attach themselves to programs and files in their efforts to spread infection. Using this approach there are multiple paths in which a clean program can be compromised: appending, surrounding, integrating and replacing.

- Malware which appends itself to software is able to add itself to either the beginning or end of a block of software as displayed in the following graphic:



- Malware which surrounds itself around software is able to "engulf" a block of software thus surrounding it completely as displayed in the following graphic:

¹Using vulnerable network connections to reach and infect other devices



- Malware using integration is able to completely merge with a piece of software, becoming one. This is displayed in the following graphic:



- Malware using the replacing tactic actually replaces parts of a program with it's own version. Once the program starts it eventually calls the Malicious code, as opposed to the intended block.



A changing scene The malware scene is one that is constantly evolving; Back in 1988 people mostly wrote malware to become famous, while these days it moved more into the space of organised crime[73]. With this also come changes to the nature of malware itself, aiming to stay hidden because the goal is no longer fame and destruction but enslaving victims systems for various illicit gains.

Source code for malware can often be found on (underground) websites by an experienced searcher, which leads the way to a change in the way malware is being developed. Much modern malware is a tex-mex of code and concepts from different authors, making new malware increasingly complex[74]. A good example of this is the Sasser malware of which there are over 15000 versions. Although most samples encountered in the wild are adaptations in one form or another, development of new and innovative malware does not stop. Stuxnet[16], cryptovirology[70] and recently Rombertik[75] are all examples of such development. It is only a matter of time before their inner workings are dissected and made available for the world to observe, learn from, and exploit. Experts warn that the technology used in the most complex cyber-weapons of today is readily available to the hobbyist malware writer tomorrow[16].

4.2. DOCUMENTED MALWARE INCIDENTS

This section provides case examples of malware found in the wild and had a considerable impact. These are the Conficker worm and Stuxnet ICS malware. Conficker, discussed in 4.2.1, is a sophisticated worm which managed to infect millions of computer systems in over two-hundred countries, it contained a specific activation date with unknown payload causing fright and wonder within the security community. Stuxnet, further discussed in 4.2.2, is credited with being the first cyber-physical weapon which demonstrated the true potential of cyber-warfare by causing physical damage to nuclear enrichment equipment. Taken together they demonstrate the potential impact malware can have on our hyperconnected lives.

4.2.1. CONFICKER

In November 2008 the Conficker malware first entered the stage, rapidly and widely spreading across the world in only a short period of time. Conficker is classified as a worm due to its self-installing and self-propagating features, using network connectivity as its main delivery mechanism[76, 77]. Network propagation is not its only spreading mechanisms though. Conficker has the ability to infect other systems through network shares and USB Drives. Not only was Conficker the first worm in years, but it came with the potential to cause equal mayhem as worms in early 2000s[78].

While estimates on the total number of infections vary widely, they are without a doubt in the millions. Infections are found in home and business environments, including major multinationals. One of the key reasons for Confickers ability to infect and spread at such a scale is the lack of protective measures by users, such as installation of software updates and patches[79]. This became more apparent when it is revealed that the vulnerability in MS Windows, exploited by Conficker to infect a system, had a patch available before Conficker even released into the wild.

To gain access to a targeted system, Conficker exploits a Windows server service exploit through a so called remote procedure call (RPC). Because the actual malicious code is started indirectly, the exploited service is told to call a dynamic link library (DLL) which is then loaded into memory. The worm thus runs as part of svchost.exe. After infecting a machine the worm adds a configuration to the registry that activates this DLL once the host starts-up. This ensures that Conficker becomes persistent on the system, even after rebooting the system[78, 80].

After infecting a system, Conficker installs a pseudo-patch that on a first glance seems to repair the vulnerability initially used by the worm to gain entry to the system. This prevents other actors (e.g. hackers or malware) from gaining access to infected systems through this vulnerability. Upon further investigation, this patch is used to maintain the presence of a backdoor which is used for re-infection and updating of the Conficker malware. Updating of Conficker happens when a newer version of the malware calls the RPC on an infected system, which performs a callback and downloads a file. The worm makes use of RSA signatures, cryptographic techniques, to control which code is loaded into the infected machine. All files have to be correctly signed and are rejected if the checks fail[76, 80]. Conficker makes use of more evasive techniques though. Other approaches used are adding of irrelevant unused registry keys, useless in-memory patches to DLLs are installed and all rollback options prior to infection time are wiped from disk. Even in depth code analysis is made more difficult by code obfuscation techniques. These all indicate that Conficker was crafted to a high level of sophistication.

The earliest versions of Conficker recruited all the machines that were infected into a botnet. These enlisted machines created a list of up to 500 domain names every day by running a domain generation algorithm (DGA). At the same time, the Conficker authors used this same algorithm to establish an identical list[76]. A small number of these domains were then registered by the authors and configured a name resolution server for this subset of domains such that they could be resolved to IP addresses by DNS resolvers. After security researchers put barriers in place to combat the conficker botnet the amount of potential domains generated by the DGA was increased to 50.000 potential domains, probably to obstruct those trying to take down the malware. A later variant of Conficker used enhanced peer-to-peer (P2P) technologies to maintain the botnet, removing the need for static domain name generation. The combination of DGA and P2P communication in a botnet is an innovative move which "required new and unprecedentedly intense collaboration not only among the security community, but also with actors controlling decisive elements of the internet"[78]. The Conficker botnet is thought to be used in the executing of wide-scale fraud, spam and general Internet misuses[81].

The Conficker outbreak was evidence of just how effective a worm can take advantage of poorly managed and unpatched systems connected to the internet at a world wide scale. Demonstrating the ability of sophisticated malware to terminate, disable and reconfigure native operating systems and security software. Conficker helped unify a large group of cyber security professionals and academic researchers into the Conficker Working Group (CWG) which played a role at keeping the eventual effects to a minimal where the potential for disaster could have been huge.

4.2.2. STUXNET

The Stuxnet malware became infamous when it demonstrated the power to influence the physical world through the industrial control systems of the Natanz nuclear power plant. After its discovery there have been many security researcher diving into the forensics of the malware, finding many samples outside of Natanz in the process. This section aims to provide the reader with a conceptual understanding of Stuxnet and what it entails for the world at large. Because this work's main topic is a novel intrusion detection system for ICS and Stuxnet directly targeting ICS, this topic will be treated in more depth than the previous. This subsection builds on work done by [10, 16, 28] for the most part.

Stuxnet is a textbook example on Cyber Warfare. It is often credited with being the first cyber weapon by demonstrating the capability of software to impact the physical world. As such the impact it has for the future is substantial indeed. Early analysis of the Stuxnet malware revealed that its target was likely situated in Iran with nearly 60% of infected hosts located there. The target was the Natanz nuclear power plant, with the intention to delay the Iranian nuclear program by causing the enrichment centrifuges to malfunction. Tremendous efforts were taken by the developers of Stuxnet to remain undetected, which supports that the aim was sabotage and not extensive physical damage. Although this could easily have been achieved. The sabotage was achieved by keeping snapshots of the peripheral I/O values and replaying these during the attack phases, this prevented systems and technicians from realizing the system was no longer operating as they expected. While the literature and publications on Stuxnet report mostly on its ability to change the

speeds of centrifuge rotors, there was another attack routine which seems almost forgotten. This second routine is not only an order of magnitude more complex, but to those familiar with ICS security it is recognised as a nightmare. However, it is important to realise that both attacks have the same goal; damaging the centrifuges. Even though different tactics are deployed. The two attacks are:

- Over-pressurisation of centrifuges (The more complex attack.)
- Increase rotor speed, taking them through their resonance speeds (The simpler attack.)

Interestingly, in the development cycle of Stuxnet, the more complex pressurisation attack was the first released. Only later the much simpler and less stealthier rotor speed approach was deployed. This seems to be related to the politics behind the malware, which this work will not discuss extensively. Figure 4.1, based on Figure 2 in [16], displays the different attacks implemented by Stuxnet.

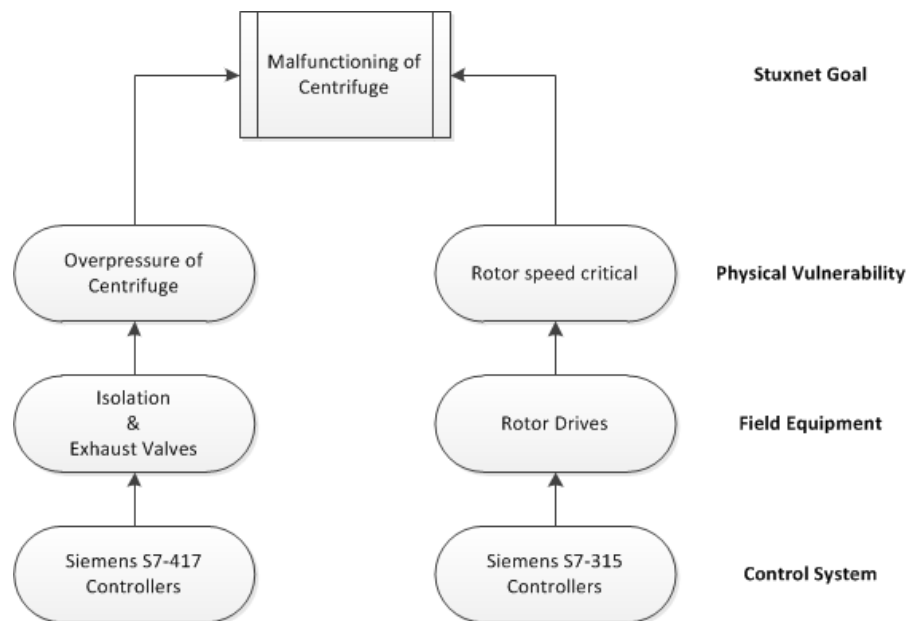


Figure 4.1: Stuxnet attack strategies, based on [16].

Overpressure attack The first known attack was targeting the Cascade Protection System (CPS) which was controlled by Siemens S7-417 controllers. The CPS is there to ensure that the sensitive enrichment system would be able to cope with the low quality of the IR-1 centrifuges used by Iran. These centrifuges have a high frequency of failure but could be built at industrial scale. The control system ensured that once a centrifuges showed signs of failure it would be isolated without further effecting the enrichment process. The malicious logic that Stuxnet installed on the controllers completely decoupled legitimate control logic from all electrical input and output (I/O) signals, while the attack was dormant the malicious code ensured that the legitimate code had access to the received signals to avoid suspicion. In security terms it executed a man in the middle (MiM) attack on the control logic. The concept of this MiM attack is depicted in figure 4.2 where the traffic is diverted and first passes through a malicious actor, who can thus modify the data, before being sent along towards the intended receiver.

Once a specific condition in the process was met all input signals would be recorded for 21 seconds, which would then be replayed to the network during the actual attack sequence. The legitimate control logic would also continue to function during this time but would receive fake input values from the malicious code while in addition any output manipulations would be decoupled and no longer have any effect on the system. The consequences of all this is an increase in the operating pressure in all non-isolated centrifuges without any means to reduce this pressure. Due to the replaying of captured signals control engineers remained none the wiser and were thus kept from interfering in any way. The attack would stop once the malicious code decided that it was enough, it is unknown what the exact conditions for this were but it was likely determined by looking at vibration sensors suggesting a mission abort.

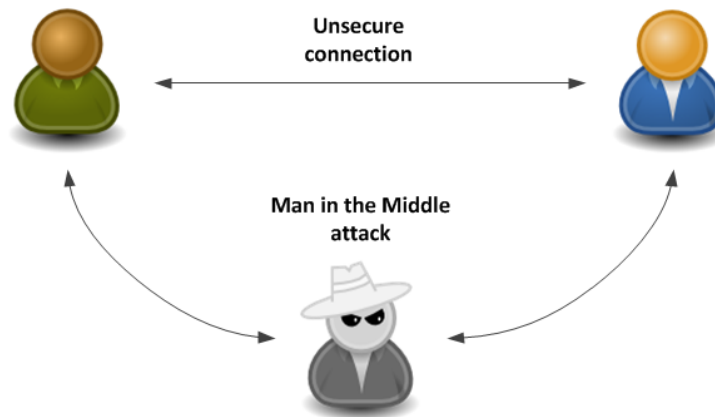


Figure 4.2: Man in the middle attack

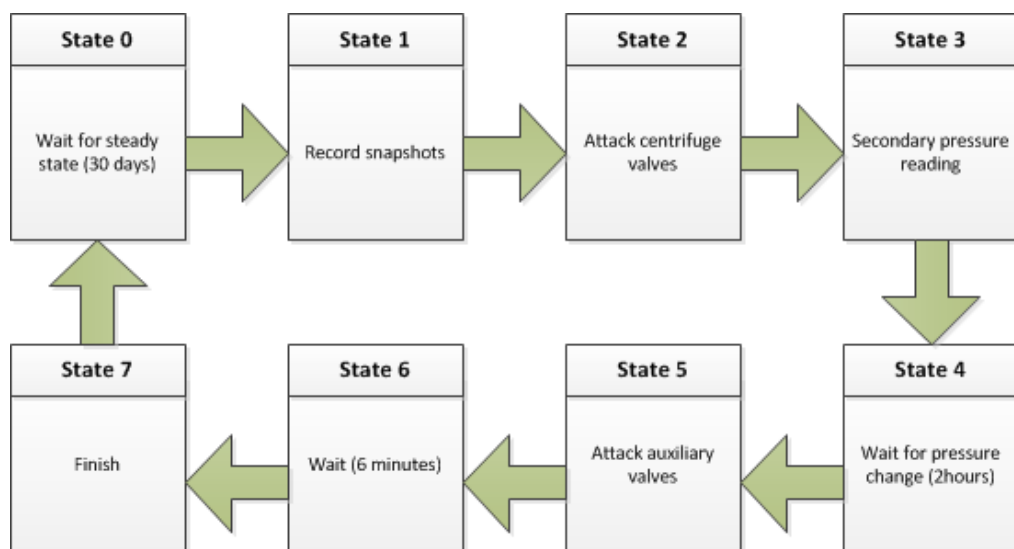


Figure 4.3: Stuxnet overpressure attack

This timely stopping of the attack is another indicator that sabotage and delays were the aim as opposed to destruction. If the intention was to cause catastrophic destruction the malicious code would simply have to keep running and then sit and wait for the system to enter a critical state.

Rotor speed attack The rotor speed attack, which remained undiscovered until 2010, was much less stealthier and also much simpler than its predecessor. While the aim of the attack remained to cause malfunction to the enrichment centrifuges, the attack vector deviated significantly from the first attack as it targets the centrifuge drive system (CDS). Interestingly the code for the earlier attack was still present in the code, but would no longer be executed, which was the reason researchers were able to discover the existence of the earlier version of Stuxnet and the much more complex attack vector.

This newer approach targeted the Siemens S7-315 controllers which contained the control logic for the CDS, which was used to hold the rotor speed constant to the given set-point. The attack operated with a period of about once per month and aimed to change the drive speed of the IR-1 centrifuge. Normal operating speed of the IR-1 centrifuges is 63.000 rpm, which was ramped up to 84.600 rpm for 15 minutes by Stuxnet. On a consecutive attack run, this approach was changed and all centrifuges were slowed to around 120 rpm only to speed them up again afterwards. The approach took a total of fifty minutes and took advantage of the supercritical design of the IR-1, meaning that normal operation of the centrifuge was above its natural frequency. Taking the already fragile devices through their harmonic frequency range would cause the rotor to vibrate (if only briefly) and cause potential damage to the already fragile devices, or even break it completely.

Malicious code running on the controller simply suspended legitimate control code and after running its attack sequences executed a BLOCK END command. This tells the process to jump back to the start of the code, in this case thus escaping the legitimate code altogether and thus constantly reiterating the attack while suspending any subsequent code. The attackers did not take care to have legitimate code executed with fake input data as they did in the CPS attack. It would seem that this is unlikely, which can be attributed to the fact that a change in rotor speed would not be detected by operators or the control systems. Reason for this is that the control application retrieves its data from the controllers memory banks, which need to be actively updated by the control logic. The control logic that got suspended by the malicious code. As such the rotor value turns static while matching normal operation conditions.

Experienced control engineers would nevertheless be able to eventually detect the blocking of control logic for up to an hour, especially when 164 centrifuges (or multiples thereof) would be brought from 63.000 rpm down to 120 rpm and then ramped up again. The sounds of which should have been very noticeable. Additionally a Stuxnet infected system probes controllers every five seconds for data specifically injected by Stuxnet, a proper forensics lab would not miss the traffic such an approach would produce.

Spreading the infection The infection of Stuxnet can be seen to spread on two levels, first the operational level and second the control system level.

On the operational level the attackers choose to evade security measures (firewalls, intrusion detection, virus scanners, etc) by aiming for easier targets that had legitimate access to the facilities: Contractors. In most facilities that utilise industrial control systems there will be a dependency on contractors to some extent, which while often very good at their engineering tasks are bad at cyber security. Their cyber security policies were in this case certainly not at the same level as with the Natanz facilities. Installing Malicious software on contractor mobile devices (e.g. laptops) and external drives (e.g. USB sticks) were sufficient to get Stuxnet inside the facilities as sooner or later contractors would unwittingly carry the infected devices through any security measures present and connect them to even the most critical control systems. It is important to note that this approach is very likely to be explored as an approach to infiltration by any follow up attacker that seeks to gain entry to a secured network.

On the control system level, Stuxnet achieved its malicious intent without exploiting any zero-day vulnerabilities or other fancy tricks. The worst vulnerabilities in this cases were the features offered by the control system components. This also means that vendors would be unlikely to release any overnight fixes to enable customers to patch their systems.

Aftermath To control engineers seemingly unexplainable problems with the control system is one of their most frustrating experiences, which is usually referred to as *chasing a demon in the machine*. Adding the possibility that this can be caused by a malicious actor makes this even more frustrating. On the plus side though, one should note that a sophisticated malware such as Stuxnet will not be a lone operative, but rather a group with sufficient funding, experts covering all fields and access to a testing facility. Unfortunately one

should not assume this is a deterrent for copy cats, the code and approach are out there and it is only a matter of time for others to incorporate this into their designs.

If anything, the Stuxnet malware demonstrated the viability of a digital attack against physical systems[29]. It is unlikely that Stuxnet will be remembered as having a major impact against the Iranian nuclear program, but it will likely go down into the history books as the first field demonstration in cyber physical weapon technology. These cyber physical attacks are different from everyday cyber attacks in that they involve three layers with each having their specific and unique vulnerabilities[16]. Figure 4.4 represents these layers and indicates their malicious use.

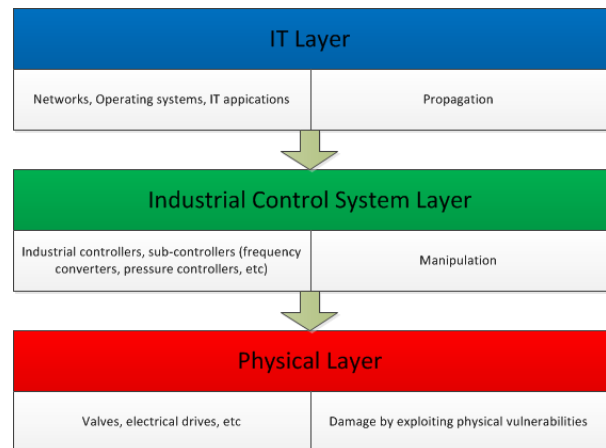


Figure 4.4: Three layers of a sophisticated cyber-physical attack, based on [16].

4.3. IMPACT ON INDUSTRIAL CONTROL SYSTEMS

Interestingly traditional ICT malware comes in both targeted as non-targeted versions, while of ICS malware currently only targeted samples are known in the field. This is not to say this will not change in the future. With the publicly available information on malware, e.g Stuxnet, it is only a matter of time before knowledge disseminates and fuses with more traditional malware (e.g. Conficker).

It is clear that control systems are an essential part of many industries, two of which are offshore and dredging. More importantly they are at the heart of many critical infrastructure. As such an understanding of the impact that malware can have on such systems is paramount.

Research into malware which is running on control systems is a relatively new field of research with only limited published work. Currently there is little publicly available knowledge, besides speculation, on the potential impact that malware can have on ICS. This not only applies to the academic world but also in the field, where control challenges and problems are directly caused by various cyber incidents. Even when not identified as such initially. However the last few years have seen increased research into the security of ICS systems and with it the effects of malware on such systems.

Experimental work conducted by Fovino[26] is an example of such research. Their experiments investigated the consequences of malware, build for traditional IT systems, gaining access to a control network. For this they developed a test-bed that recreated the ICS environment as can be found in a typical power plant, but keeping it in a protected environment. This then mimicked the physical environment of a power plant along with its security policies, maintenance policies, firewall rules and access policies while allowing for the incorporation of tools that enabled the conducting of experiments in a systematic and scientific manner. The strings of malware used to infect the system were: Code Red, Nimbda, Slammer and Scalper. The research concludes that no dramatic consequences were observed during the experiments themselves. However they note that in different situations the effects could potentially be dangerous.

In addition to malware which is created for traditional IT systems, there is malware that is specifically designed to interfere with control systems. The most famous example of which is Stuxnet, which was discussed previously in 4.2.2. In an attempt to help the academic community along in this field [26] created malware that specifically targeted control systems and evaluated it on their experimental environment. The intent behind the attack was to cause a denial of service on Modbus² slaves on the control network because a disruption in master-slave communications could lead to a potentially disastrous situation in a power plant. Upgrading the malware by coupling it to a worm-mechanism gave it the ability to simultaneously infect multiple computers in the network. This would further increase the bandwidth consumption and speed up network degradation. After conducting experiments it is demonstrated that the ICS malware was able to completely circumvent traditional security mechanisms by adopting *ad hoc* infection and attack strategies, enabling the

²Modbus is one of the communication protocol used by PLC's in control networks.

malware to disrupt or even seize control of vital components.

Although available research and case studies is limited, with the available knowledge it is possible to get insight into the effects malware might have on physical systems. As was discussed in 2.2.6 ICS security is often defined using AIC: availability, integrity and confidentiality. The following will discuss the effects malware can have on each of these elements.

AVAILABILITY

Due to their real time requirements control systems are typically sensitive to disturbances of availability. Losing connections can have large impacts on a system, especially if the design did not take this into account. Disruption of service not only affects the control system but will likely have consequences on a physical level.

There are strings of ICT malware which are known to flood the network with various sorts of traffic which can trigger an (D)DOS³ on control components. Research discussed earlier have also shown that malware designed for ICS specifically are quite capable of creating a (D)DOS on components.

The effects caused by this loss of availability will depend on many things: Design of device logic (is there a fail-safe to fall back on when no information is received), backup LAN connections to reroute traffic over another channel, physical process itself. Disruption of service might be annoying but in many cases the effect will not be severe.

However it is important to note that triggering a denial of service at the same time that the alarm system should kick in can have disastrous effects. For example when a fire alarm is triggered on-board a vessel and the safety system is not informed the systems onboard to deal with the fire will not be enabled. Additionally the firefighting team on duty will not be informed of the event. The correct timing of such an attack could thus have dire consequences.

INTEGRITY

Control systems are often of a distributed nature where many components communicate with each other to determine which actions to undertake. Altering the information flowing between nodes can have a serious impact on operability and safety of a system.

The suspicion that malware might be able to influence operations by targeting the integrity of a control system has been present for a long time. This could cause significant problems, not just to the physical system itself, but also to those operating in close proximity to the equipment. Security and safety are thus closely related.

After the discovery of Stuxnet it became clear that this suspicion has been correct. As discussed previously in 4.2.2, Stuxnet compromised the integrity of the system by executing a man in the middle attack on the logic running on PLCs which controlled enrichment centrifuge behaviour causing input and output signals from the control logic to be decoupled. While Stuxnet operated on an Nuclear enrichment facility there is no reason to suggest that the same would be impossible on other industrial installations. Malware that compromises the integrity not only has the potential to influence the physic system and take control, but can do so while keeping operators in the dark.

CONFIDENTIALITY

The confidentiality indicates to what level information that is present on a (control) network is inaccessible from prying eyes and is often considered the least important of the AIC triad. This does not mean however that it can be neglected. Having detailed information on a specific system can be used for many reasons, which has been discussed previously. Two examples would be corporate espionage and reconnaissance for a larger attack, having detailed information on the design and layout of a facility can help attackers enormously in making an attack more successful and hidden from operators.

Malware which main focus is on stealing information from these more traditional systems have been known for years and even quite common today[12, 82, 83]. It is important to note that even though many systems used in ICS are highly specialised compared to ICT systems, they nonetheless are increasingly built upon common ICT platforms, which are vulnerable to traditional espionage malware.

Malware is not limited to traditional systems however, Stuxnet has shown that cyber attacks can easily transition into other domains[84], for all its offensive abilities, was equipped with espionage capabilities. To let Stuxnet be effective at its job the designer would require access to very detailed information on the targeted system. Another example is the Duqu malware which, displaying many similarities with Stuxnet,

³(Distributed) Denial of Service

was designed as an information gathering platform. It has even been indicated that due to their similarities it is conceivable that Duqu was designed to steal information in preparation for a Stuxnet-like attack[85]. This demonstrates the potential for malware to steal valuable ICS operational data which can later be used to execute targeted attacks. Discussing the risk for loss of confidentiality in ICS with security experts indicated the same: Much like any other systems, malware can steal information from a ICS. Once a string of malware has gained a foothold on a system, the path used to get in can often be used in reverse to get information out again.

4.4. OBTAINING MALWARE

One of the larger risks for cyber security is that of an malicious event caused by insiders. While cyber security experts are well aware of the ease with which systems can get infected with malicious software and are generally well informed on the ability and negative impact that malware can have on equipment and business continuity. This does not apply to layman however, whom still adhere to security practices not necessarily recommended by security professionals [86]. This can provide an malicious insider with multiple vulnerable surfaces to attack, the internal network directly or by using an unsuspecting co-worker.

The main challenge for the insider will be to obtain a string of malware which can be used to complete their goal. There are two possible avenues that can be taken here:

- Prototyping the malware
- Purchasing the malware

This section will start by discussing the prototyping of malware, to give insight into the processes an insider would have to go through to develop a custom piece of malware. Whereas the section 4.5, will investigate the possibility to purchase malware.

The goal for this prototype malware, and the approach used to create it, are discussed in 4.4. The flow of information needed to achieve this goal will be presented and discussed in 4.4.1. Various modules, called exports, are designed which when taken together will form the malware. This design choice, along with export descriptions, are treated in 4.4.2. The implementation stages of the design follow in 4.4.3, which will wrap up the creation of malware part.

GOAL AND CONTROL LOGIC

Due to the prevalence of control systems in the dredging and offshore industries the goal of the prototype example will loosely be based on Stuxnet (see 4.2.2). Keeping things simple and out of murky legal territory[87] a simple program will be targeted which builds a short story. This story will be build with the help of the user answering some questions posed by the software. After the program received response to all questions it will load the logic required to build the story, which it retrieves from a file called *logic.dll*. Such a file, called a dynamic link library, enables the program to select which logic to run. This offers more flexibility than a statically build program that requires an update to change its internal logic. Most, if not all, modern software work with dynamic link libraries, including Step 7 which was compromised by Stuxnet by replacing such a .dll file.

The aim for this malware is to mimic this behaviour and replace the *logic.dll* file with a malicious version which changes the happy story to an unhappy one authored by malicious actors. This approach is represented graphically in figure 4.5 by the two flow diagrams. Displayed in the upper flow is a clean (green) system, whereas the lower flow represents a compromised (red) situation.

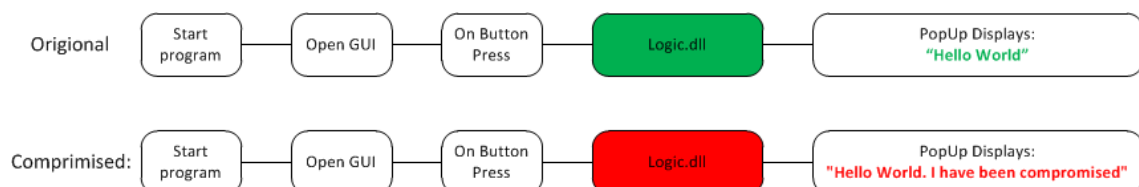


Figure 4.5: ICS malware demo: Clean versus Infected file

Prior to infection the malware should check that the system is a valid match, thus containing the story software. Infection should happen through a malicious USB stick as this will ensure air-gapped system can

be reached, much like Stuxnet. An extra, more advanced, goal is to ensure persistence on the infected system. Thus after a reboot load itself into memory again and ensure the story software stays infected after changes and updates. The goals can thus be summarised as:

1. Identify valid system for infection
2. Infection through malicious USB device
3. After infection replace the logic.dll with malicious version
4. Maintain persistence after the system reboots (extra)

4.4.1. CONTROL FLOW

Achieving these goals will require the malware to have logical decision elements embedded in its design. This logic is build on the control flow and steps taken, starting at the malicious USB up to actual infection and persistence. Figure 4.6 represents these various steps and thus sketches the control flow that will be implemented. This subsection will quickly run through the various stages.

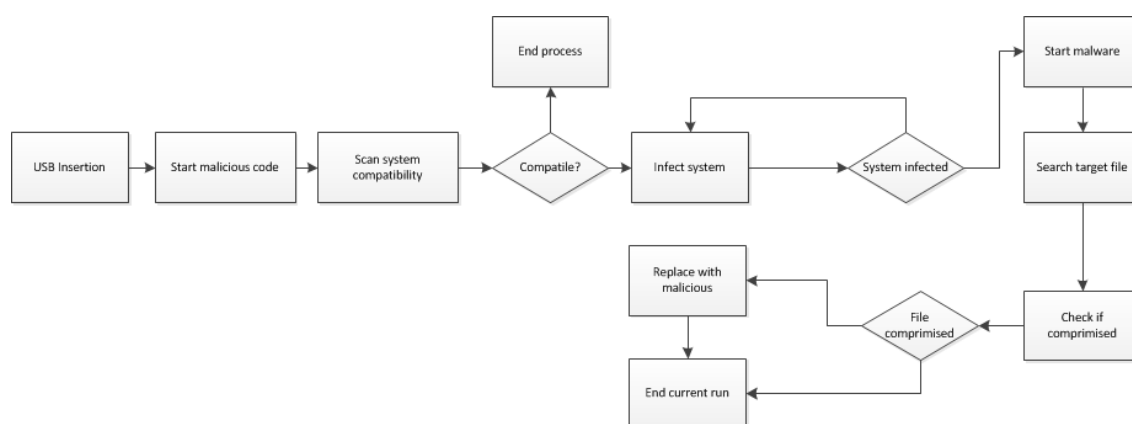


Figure 4.6: Control flow

System infection The first step in the infection process will be to create a connection between the bad USB device and a computer system. Obviously this step will be executed by the presenter at demonstration time. After a connection to the computer system is made the malware is called which will start scanning the system to meet requirements. A valid system will return true for all of the following:

- Operating System is the Windows XP platform
- Current user is in the administrator group
- Targeted program is found on the system.

If these requirements are met the malware will further load itself into memory with a flag that indicates if persistence is to be enabled or not. Once loaded either the routine for persistence insurance or attack execution will be started, pending the persistence flag.

Ensure persistence Once started this routine will simply check for a copy of the malware to reside in the windows "Startup" folder. If a copy is found it will continue with loading the attack routine. Should no copy be found the malware first places a copy of itself in this folder and then starts the attack routine. This ensures that once the system is rebooted the malware is loaded back into memory again. While far from a sophisticated method it suffices for demonstration purposes.

Execution of attack This is the main routine of the malware and contains the actual logic to perform the attack. The routine runs on a looped timer and will go through various steps:

1. Ensures the current date is between a set initial and expiration date, if this is not the case the logic is suspended
2. Locates the directory of the targeted program. If none is found it starts a self-destruct routine
3. Locates the targeted .dll file and creates a copy if none is present
4. Replaces the targeted .dll file with a malicious version
5. Suspension of the logic for specified time

4.4.2. EXPORTS

The control flow indicates a logical separation into different modules. Following the naming convention used by [10] in their Stuxnet research, these have been called Exports. Table 4.1 gives the exports that will make up the demonstration malware. The remainder of this subsection explains the intent behind each of the exports.

Export	Function
Core	Main installation
Recon	Responsible for system information
Exploit	Responsible for the attack logic
Payload factory	Returns the malicious file
Logic	Malicious file
Harakiri	Removes the infection

Table 4.1: The selection of Dynamic Link Library (DLL) exports

Core module Once the malware is started the Core will be called. This should be seen as the heart and brains of the software which coordinates with various other modules to create functionality and determines what to do depending on their output. The first thing the Core will do once running is load the configuration file to get an "understanding" of its environment and actions to be taken.

Recon module The intention for the reconnaissance module is to provide knowledge of the surroundings back to the Core, this enables it to make decisions as to what to do next and in what manner to do this. The information retrieved by this module is basic system information (e.g. operating system and version), privileges of the users running the malware, current running processes, if there is a network and internet connection present and if the system has been infected prior to the current infection.

Exploit The exploit module is responsible for execution of the attack on the target system. This means it identifies the file which should be replaced, makes a copy of the original, loads the malicious file by calling the Payload factor and replaces the original file with the malicious version.

Payload Factory This module provides the malware with the capability to craft payloads. This could be to exploit network connected systems, but in this work it aims to supply the malware with a malicious .dll payload.

Harakiri module This module provides the software with the capability to remove itself completely from the system it is currently running on. This is achieved in two steps: The first is to remove any malicious file created and replace it with the original where one was replaced. The second and final step is then for the malware to self destruct and remove itself from the system.

4.4.3. IMPLEMENTATION

With the goals, control flow and components listed the last remaining step is implementation. The implementation process is divided into different stages, each building on the stage before it. The following is a description of each stage:

- **Stage I: Foundation** The first stage aims to create the minimum viable product, providing the foundation on which the rest will build. This foundation will determine the validity of the system. If the system is invalid the code stops running. If the system is valid the next applicable routine will be selected and started.
Additionally, this is also the stage in which a feel for success will be gauged and early issues can be detected.
- **Stage II: Payload** The second stage aims to take care of the malicious logic that will search for the targeted file, create a copy of this file if required and replace the file with a malicious version. This then implements the exploit and payload factory modules.
- **Stage III: Infection** The third stage will ensure that the malware can actually be reach and infect various systems by using a malicious USB stick.
- **Stage IV: Persistence** The fourth stage is to add further (extra) refinement. This in the form of persistence after system reboots and self-destruction.

Throughout the research into malware the concept of malware as a service (MaaS) was repeatedly touched upon. The MaaS concept also directly relates to the potential to obtain malware through financial means. After some further investigation this proved to be the perfect way forward because it demonstrates that even without skill an actor can obtain malware to inflict harm to others. Malware as a service is thus discussed next.

4.5. MALWARE AS A SERVICE

Creating and executing a succesfull cyber attack is often seen as a difficult and/or expensive process, which is not available but to a select few. The previous chapter demonstrated that creating even a simplistic malware for the windows operating system is not that simple.

This view is not necessarily true however as the sizeable criminal underground markets are selling Malware as a Service (MaaS), much like it's legal counterpart Software as a Service (SaaS). Such markets provide access to build it yourself kits which enable anyone who purchases these kits to create there private malware. Often these kits come enabled with a hosted management service such that the malware is easily deployed and managed over the internet. Execution of an succesfull cyber attack is thus made relatively easy and takes away the need for experience. To build on the previous section, obtaining malware, this section will discuss a well known service known as Zeus.

ZEUS

One of the best known examples of malware as a service is probably Zeus, which has been first detected in the wild since at least 2007. Zeus is alternatively known by the identifier Zbot, the specific trojan used to infect machines. The main aim of the Zeus malware is to mine compromised machines for financial information and login credentials.

Zeus can be purchased from underground markets for as little as 700 USD and comes in the form of a tool-kit. This tool-kit can be used by the buyer to build his own string of bot, or bots, which can then be used to infect machines using various mechanisms. One of which would be crime as a service. The tool-kit is easy in use, allowing even those users with minimal technical know how to configure and create a functional botnet and even comes with an manual in both Russian as English. While many different users have used Zeus to create bots, none of them will have the same signature. This means that virus scanners are unlikely to detect a new infection is taking place.

The Zeus tool-kit contains three components:

Configuration file: The configuration file is used by the user to define the details of the Zeus malware. This includes the location of the command and control server, timing information for updates and when to send out stolen credentials, and other functionality to be used by the bot. As each user will

have a different configuration file, this will also be the main cause for the difference between various malware strings. Creation of the configuration file couldn't be simpler, it requires no more than editing an ASCII document with a text editor of choice.

Builder: The builder is a graphical user interface which enables the user to perform two actions. After editing the configuration file the builder can be used to convert this into a (encrypted) .bin file. This .bin file is then to be placed at the user specified location, which is considered to be the command and control (c&c) server. Second is the creation of an executable which is the actual Zeus malware to be used in the infection of machines. Once an infection takes place the malware will contact the c&c server where in will find the configuration .bin file. This will inform the malware on further instructions.

Server: The toolkit comes with an easy to install server module which is used to control and inspect infected machines, better known as bots. The server has an easy to use graphical user interface which makes administration of the botnet an easy task, providing the user with comprehensive details on each of their bots. The server also acts as a drop points for the stolen credentials, screenshots, and output generated by the different malware features. Much like the control panel, configuring the server is a simple procedure. The user only has to drop the web server files onto a machine, look for the installation page and provide some basic settings and press go.

After setting up the infrastructure and infecting a machine Zeus offers a list of functionality, the following are considered to be part of it's main functionality:

- Gathering system information.
- Stealing credentials for e.g. protected storage, FTP servers, e-mail and online banking
- Contacting the command and control server for additional tasks to perform.

However Zeus has been build such that it is easily extended with extra features, many of which can simply be bought on the same underground markets as the toolkit itself. If a feature is not available yet a criminal can also put in a request for a specific feature and get it custom made. This ease of use and extensibility have caused Zeus to become a major player and example within the crime as a service market.

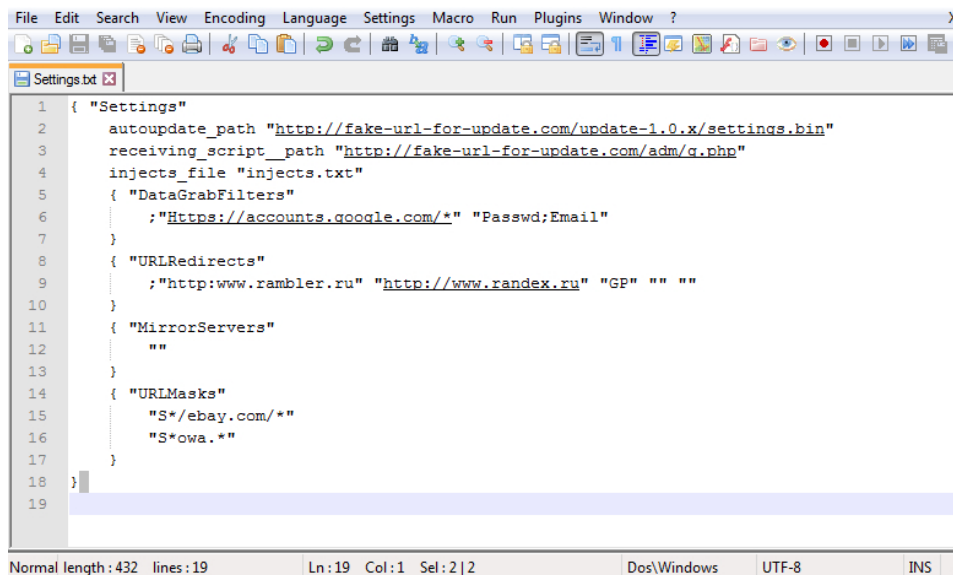
BUILDING THE ZEUS BOT

Before deploying ZeuS the malware has to be configured to fit the requirements and details of the actor, which includes for example the location of the command and control server. After configuration the bot is ready to be build, this will create a malicious .exe file which can be spread to unsuspecting victims. The final step is to configure the server such that the bot(s) can be controlled and updated from a remote location.

Configuration file The settings.txt file, as shown in figure 4.7 is used to set global configuration parameters. This includes the location where a bot can find updates and new scripts which is should executed on the infected machine. The file also highlights a number of other options, such as the "URL Masks" section. This section enables the bot to perform certain actions when a user visits a specified website on the infected machine. Figure 4.7 contains the URL masks for ebay.com and owa for example. The options that accompany the URL enable any of the following actions:

- N Do not write data in reports
- S Make screenshot with mouse clicks on the page area
- C Preserve all cookies associated with the site and block access to it
- B Block access to the site

Figure 4.7 also lists a variable "injects_file", which points to the injects.txt file and is one of the impressive features of the ZeuS family bots. This file contains specific injects that make it possible to interact with the specified website once it is accessed on the infected machine. The injects work through the bot and thus directly on the machine, which means that any security measures (such as two-factor authentication, SSL/TLS encryption) that are present on those websites are evaded. One example here would be that every time an infected machine visits a certain bank website the account details are uploaded to the command and control server. This creates a overview of various users and their account balances. This information can be used for multiple purposes, or simply to determine which victim is most worth while to target specifically. This is simply one example, the owner of the botnet can create many variations of inject files to steal information. True to the concept of malware as a service it is also possible to download readily made definitions that can be used immediately.



```

1 { "Settings"
2   autoupdate_path "http://fake-url-for-update.com/update-1.0.x/settings.bin"
3   receiving_script_path "http://fake-url-for-update.com/adm/q.php"
4   injects_file "injects.txt"
5   { "DataGrabFilters"
6     ;"Https://accounts.google.com/*" "Passwd;Email"
7   }
8   { "URLRedirects"
9     ;"http:www.rambler.ru" "http://www.randex.ru" "GP" "" ""
10  }
11  { "MirrorServers"
12    ""
13  }
14  { "URLMasks"
15    "S*/ebay.com/" ""
16    "S*owa.*"
17  }
18 }
19

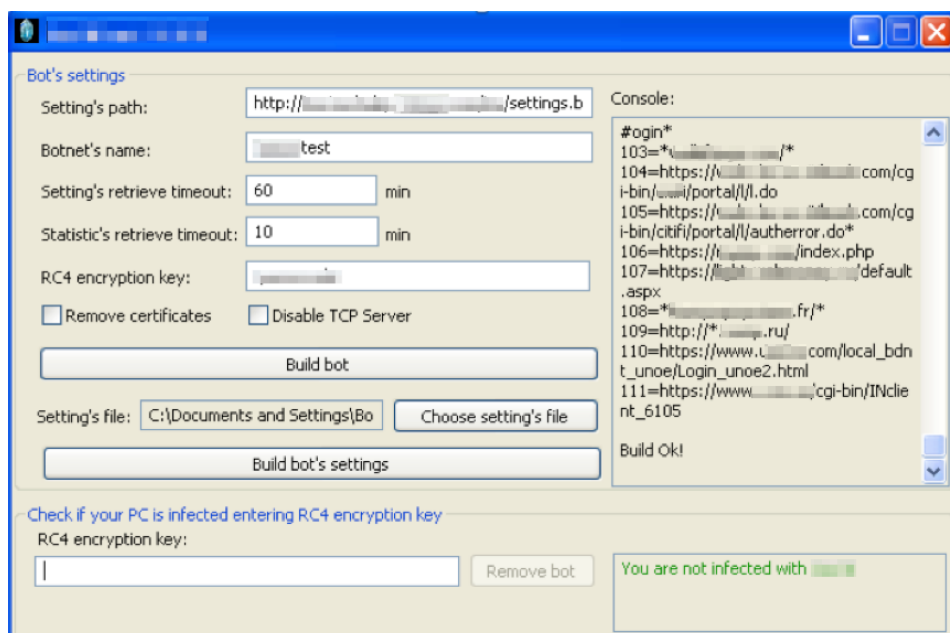
```

Normal length: 432 lines: 19 Ln: 19 Col: 1 Sel: 2 | 2 Dos\Windows UTF-8 INS

Figure 4.7: Zeus Settings.txt

Building Now the configuration details of the bot are taken care of the next step is to use the ZeuS builder application, see figure 4.8, to create the executable malware file. Building the malware requires two pieces of information:

- The URL to the settings.txt file, usually this is somewhere on the command and control server.
- A symmetric-key encryption key to embed in the payload. This key ensures that the payload can communicate with the server in a secure way.

Figure 4.8: The ZeuS builder interface used to compile the malware, source: <https://www.fireeye.com>

The malware file that is produced by the builder is likely to be discovered by anti-virus software on the victim machines. To evade detection it is possible to run the file through a obfuscater which will disguise a file to pass through anti-virus software undetected. These obfuscators sometimes come with a MaaS product, can be downloaded online for free, or in themselves be purchased as a service. The latter often providing a guarantee to evade anti-virus successfully.

Server configuration Configuration and building of the malware executable are ready. The final step before infecting victims is to setup a command and control server. There are many ways to accomplish this, but the easiest is to sign up with a cloud server for a low-cost unix server. This can be accomplished in less than 10 minutes.

The ZeuS kit contains command and control (CnC) files which should be uploaded to the server. Accessing the server through a browser will now make a web interface appear that guides the user while installing the command and control requirements. This interface is shown in figure 4.9. The only steps the user has to undertake is to fill out the required fields and press "install". Once completed the user has access to a working malware executable and command and control server. The bot is now ready for the infection process.

Figure 4.9: Web user interface for the command and control server, source: <https://www.fireeye.com>

The steps taken to build the malware and server are straight forward, clearly documented when a kit is purchased and can be completed by almost everyone. This is to show that even without the required knowledge to build one's personal malware, it can simply be bought on the black market.

DEMONSTRATING EFFECTIVENESS

To demonstrate the effectiveness of ZeuS three things are required: A victim machine to be infected with malware, a machine which is used as command and control (C&C) server, and the ZeuS builder kit. For the victim machine a windows 7 virtual machine is created, whereas the command and control server will be a virtual machine running Ubuntu. Following the steps provided above creates a file called bot.exe, which contains the malware. This file is sent to the victim machine, where it is opened by an unsuspecting user. Running this file causes it to install the malware and remove itself from the screen, leaving the unsuspecting user in the dark about what is going on.

Once the malware is operational it calls home towards the control server and registers itself such that

a communication channel is provided for future contact and changes to configuration. Once the bot has been registered by the C&C server it is possible to access logs captured by the bot which contain detailed system information. However it is also possible to obtain other information, such as user credentials when specified websites are visited, or screen shots displaying the screen from the victim machine. The latter is demonstrated in figure 4.10.

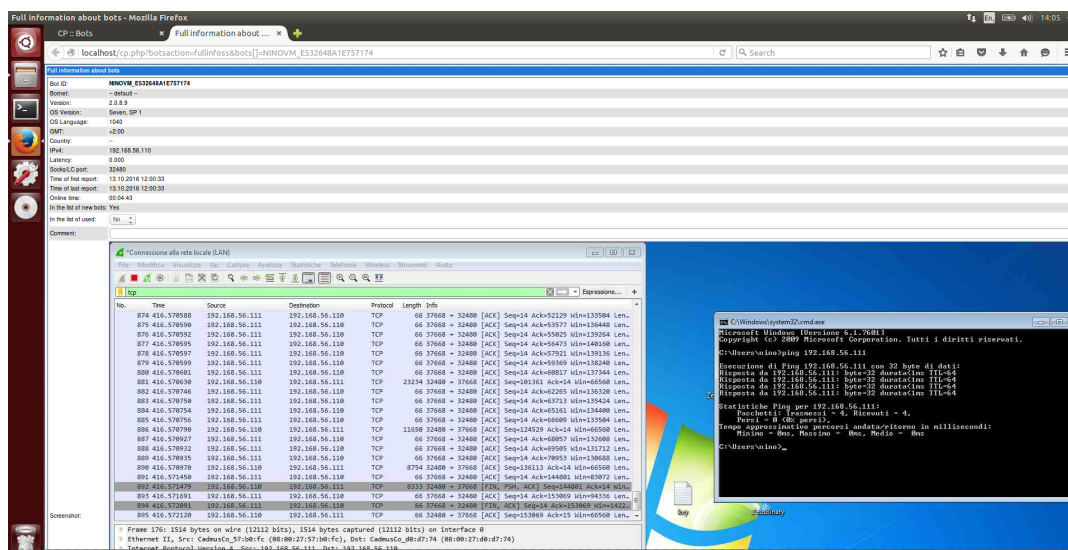


Figure 4.10: ZeuS interface panel demonstrating a compromised machine.

Although this demonstration is aimed at capturing credentials which can be leveraged to obtain banking information, it would be trivial to include a payload that offers higher control over the victims machine.

4.6. CONCLUSION

The effects that malware can have on industrial control systems ranges from annoying glitches to complete physical failure of a process. However, to achieve the worst possibly effects on a facility detailed information on the systems in use is required. As opposed to an traditional ICT environment, gaining full access to the ICS infrastructure does not mean that malware is also able to successfully exploit it and cause physical destruction. Nonetheless malware infections on ICS infrastructure are known to cause serious challenges for operators, cause downtime on business operations for owners, and have resulted in shutting down systems for periods of unplanned maintenance.

Actors with deep pockets have been shown to be capable of developing highly complex malware that is able to ex-filtrate detailed information on industrial control systems. This information can then be used to develop, or change, their malware in such a way that it will cause physical failure of field devices while simultaneously keeping operators in the dark from deviating behaviour. Potentially this can cause major problems where natural impact, injury to humans and even loss of life are possible. Currently this capability seems only to be accessible to actors with deep pockets. However, as history has demonstrated, it will only be a matter of time until the technology diffuses and gets incorporated into other (malicious) projects.

Industrial control systems have also been found to be infected and vulnerable to more traditional malware, which will often try to infect any system it gets into contact with. Although the payload will not influence the logical controller directly, it often causes problems by flooding the network and preventing communication between components. Imagining the impact when malware infects safety systems and prevents them from operating correctly in case of an emergency. Potentially this can, and some say, have led to injury and even death. This in itself has lead to multiple facilities having to shut down for unplanned maintenance as systems stopped working. The cost implications with taking such measures, directly leading to loss of production, should not be underestimated. This could even affect the reputation of a facility.

4.6.1. RECOMMENDATIONS

There are many steps that can move this field forward and provide more resistance against malware infections impacting ICS environments. Three of these that could have a large impact are listed below.

Education and awareness Cyber security is often seen as a specialist field and is not part of the educational system for other fields such as control engineering. If control engineering students would be introduced to the topic of cyber security and how cyber incidents (incidental or intentional) can impact production and the state of a plant many problems might be noticed sooner and solved before major problems arise. This because security would become embedded into their daily operation, peripheral vision and hopefully second nature. Much like has been done with safety within the offshore domain.

At the same time this would help increase awareness throughout the field as students with this knowledge enter the field and further inform professionals less knowledgeable in this domain. This increase in awareness then trickles through to those in control of projects and plants, and thus potentially having a cascading effect eventually reaching those in control of investments and monitoring the bottom dollar.

Secure by design During the design of control systems security can be included to be part of the design, as opposed to a bolt on solutions after an incident happens. Much problems might be solved if control designers would start from the premise that the control of their systems falls under malicious intent and thus embed security into the fabric of the design. One simple examples is utilising a mechanical pressure relive valve which has an operating value that lies below the pressure at which the system will fail. Even in a worst case scenario such a valve would ensure critical pressures build-up is impossible.

Security with design often begins with awareness and proper management support though, which again builds on the previous paragraph of education and awareness.

Research Research which investigates the interaction between the cyber and physical domains is limited, especially when malware is concerned. Knowledge on the potential effects of malware infections on ICS networks help system designers and operators to harden their system and prevent malfunction by using the correct techniques and tools. This would be especially potent where accidental infection is concerned as these are not specifically targeting a facility.

Additionally research into resilience engineering of control systems and their physical components can prevent cyber incidents from causing major problems. Having a system that is physically unable to fail regardless of the control commands received or set-points used is hard to exploit as attackers would be held in check by the laws of nature.

5

A NOVEL INTRUSION DETECTION SYSTEM

In this hyperconnected day and age, intrusion detection systems (IDS) are a required addition to the security infrastructure of many modern organisations and can be found in the likes of both office and industrial environments. The primary purpose of such systems is, as their name might suggest, to monitor a system or network for potential malicious activities such as intrusion attempts, policy violations and malware threats. Typically, an IDS records information related to events observed and determines a corresponding threat level, if any, on which follow-up actions can be based. If the threat level given to an event exceeds a specified limit, an alarm will be raised and security personnel is notified of the potentially malicious event. Additionally an IDS might be equipped to respond to and attempt to contain any detected threats. There are multiple techniques available which can be leveraged to execute precisely such a measure, for example automatically changing the security environment (e.g. firewall) or removing malicious content found in a threat[88].

Research investigating new approaches or improvements to IDS in ICS environments generally focus on network traffic captured in some way, often by having a network switch mirror data streams to an IDS server. This server then performs a detection strategy on the data received, such as signature- or anomaly detection, and determines the level of threat an packet presents. When dealing with ICS data this means that knowledge of the specific protocol(s) used is required, which can be problematic as there are many and most are proprietary at that. As such, most research taking control protocols aboard focus the open Modbus protocol. An additional hurdle when monitoring the network itself was demonstrated when Stuxnet recorded and then spoofed control information, keeping control engineers (and presumably IDS) in the dark by taking over the logic on controllers.

An potential approach to evade these issues is to make use of the states in which a system can operate. In the domain of control engineering these states are well defined and so are their failure domains. A requirement by the safety rules governing these facilities. This physical state of the system might be utilised in the design of a new IDS that aims to detect threats that aim to change the physical state of the system and thus focusses on changes to system stability and behaviour, as opposed to (potentially fake) network traffic. This chapter then presents a novel IDS concept that is able to detect when a system moves from a secure state into a critical state and operates independently of network traffic and control protocols. Through an implementation orthogonal of the existing ICS system - neither sharing the communication path, nor sitting in between system components like a regular IDS or firewall would - the proposed system achieves a number of distinct advantages that will aid in the adoption and customization of said approach:

1. Extensibility and Multi-Vector Detection.
2. Unmodified signal path.
3. Ability to detect compromised ICS infrastructure.
4. Upgrade without compliancy issues.

The proposed concept will be further elaborated upon in section 5.4. First however a background on intrusion detection systems is provided in section 5.1. An investigation into available related work targeting intrusion detection of industrial control systems is discussed in section 5.2. Next an adversary model is introduced in section 5.3, which is used as start for the IDS concept.

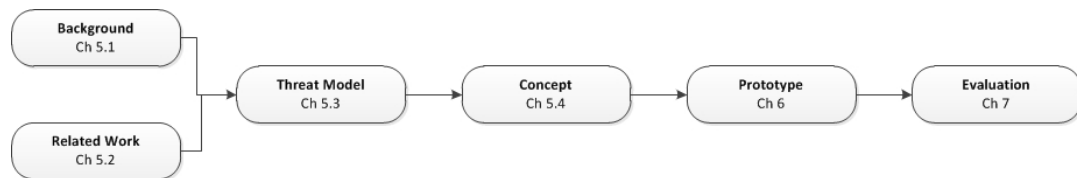


Figure 5.1: Reading guide for a novel intrusion detection system

5.1. INTRODUCTION TO INTRUSION DETECTION

In the past, cyber security would focus on perimeter defence mechanisms which aimed at keeping intruders on the outside. This approach has seen a shift to a defence in depth approach, where the assumption is that eventually any perimeter defence can (and will) be breached. As such the industry started to look into the detection of breaches on the inside of the perimeter with the aim of preventing further damage and thus reducing risk. Looking at the definition of risk this is easily seen:

$$Risk = chance \cdot effect$$

Assuming a breach is inevitable (chance = 100%), the way to reduce the risk is by changing the effect or impact of said breach. The way to detect these intrusions is to have some sort of monitoring solution, or an intrusion detection system (IDS). Such systems do not prevent attacks, but they keep monitoring the state of a system and alert operators when suspicious events are detected. This knowledge however can be used to trigger mitigation actions aimed to reduce losses. Depending on which phase of the attack was detected, it is even possible to completely prevent losses by stopping intruders from establishing a full attack chain[89].

There are modern IDS solution that have extended their abilities and do more than issue warnings to operators when suspicious events are flagged. These systems are called intrusion detection and prevent systems (IDPS) and aim to automatically mitigate detected threats, or at least contain them while an operator has time to act. Essentially such systems alleviate some of the administrator's responsibilities and enables them to focus only on the important and high risk threats.

Besides the main functionality of an IDS, they often have additional benefits which reach beyond (current) protection. One example is that of intelligence provider to aid forensic research if a breach to security occurs that adept enough to remain hidden. After such events, this intelligence can help to prevent future incidents using the same technology from occurring again (killing the attack chain in an early phase[89]). Sharing truly is caring in this respect. Another example is that intelligence gathered can be used to enable further improvements to security policies as it offers statistics on current threats, targeted resources and enables defenders to improve their threat profile. Indicating where current security policies and mechanisms might need further improvement and help monitor and evaluate the results after implementing them.

Intrusion detection systems have become an important tool in the arsenal of anyone defending their network and/or systems which are best seen as complete platforms that offer multiple benefits. Be it direct or indirect.

5.1.1. TYPE CLASSIFICATION

IDS systems can be divided into two main types which are depended on the information flows they monitor. These are network based and host based systems. A more detailed description of each follows next.

Host intrusion detection systems (HIDS) run on individual systems, also known as hosts, which may or may not be connected to a network. A HIDS monitors flows of data which are directly related to the host it operates on. Examples of monitored data are inbound and outbound packets if the device is network connected or the (critical) system files required to operate the system. HIDS are often part of security software on critical systems that are not expected to change their configurations often. Figure 5.2a gives an example network that contains a HIDS.

Network intrusion detection systems (NIDS) operate on the network level. Such systems monitor traffic between networked devices and aim to detect suspicious, anomalous behaviour in this traffic. Once abnormal behaviour is detect this is reported to a control panel which triggers an alarm and or action if the event is deemed serious. In addition to monitoring traffic a NIDS can often scan system files to ensure data and

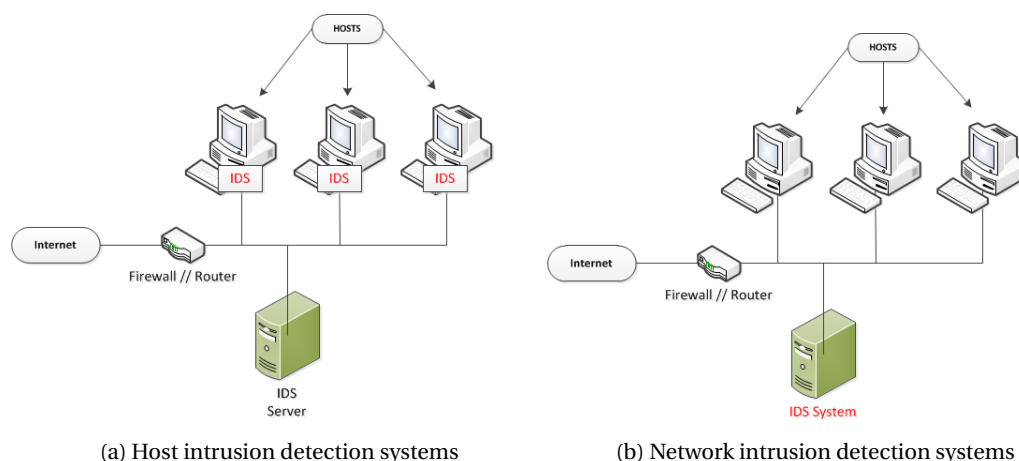


Figure 5.2

file integrity. Operating on the network level the NIDS can be utilised in an active way to improve security by scanning components, such as firewalls or servers, for the presence of potential exploits. Figure 5.2b gives an example network that contains a NIDS.

5.1.2. COMMON DETECTION TECHNIQUES

Intrusion detection technologies usually fall within one of two specific methods, or are overlapping. These two are known as *signature* and *anomaly* detection strategies. Both of these approaches are discussed next.

Signature detection searches for specific pattern in observed data. These pattern can range from system files to specific packets flowing over a network, which are known to be part of malicious intent or activities. Such patterns can include behaviour which seems to violate security policies in place. Once a new threat has been found and identified, signatures for this specific threat and it's behaviour are created, added to a signature database containing signatures known to be malicious, and distributed to other parties. Once this behaviour is then found within an IDS monitored system/network an alarm will be raised. This strategy, shown in figure 5.3a, is essentially the same as how anti-virus software operates.

Signatures themselves are based on many identifiers. Take the case of network packets, these are usually the headers of the packages. If the actual contents of the packets is rebuild it even possible to inspect the actual payload, a method called *deep packet inspection*. This approach however is very resource intensive and can have large privacy implications.

The actions to be taken when specific signatures are detected can be very different depending on the specific signature. Obviously, the security team will want to be alerted when an intrusion occurs, but this is not the only scenario out there. If a network engineer sees strange network traffic (s)he might want to be alerted if it happens again, or after a certain number of repeats, such that possible bottlenecks are detected before actual problems occur. This could be accomplished by signature detection.

Anomaly detection can be best described as the search for behaviour with seems to be strange in comparison to a given or learned baseline. This approach was presented, along with the first IDS, by Denning in 1986 [90]. The anomaly detection system is first trained for a specific time to learn what identifies as normal (baseline) behaviour. These can be metrics such as general bandwidth usage, protocols in use, nodes and hosts which generally connect and communicate with each other. Upon encountering behaviour which deviates from the baseline the IDS an flag will be raised for further action. This strategy is conceptualised in figure 5.3b.

Anomaly detection has the potential to discover threats which were not known previously, as opposed to signature detection which needs supplied signatures to know where to look out for. In modern anomaly detection systems the emerging field of machine learning, or AI, plays a major role and offers a lot of potential. A major difficulty that persists however is determining the thin line when behaviour should be flagged as "significantly" deviating from the baseline and when this is causes (too many) false positives.

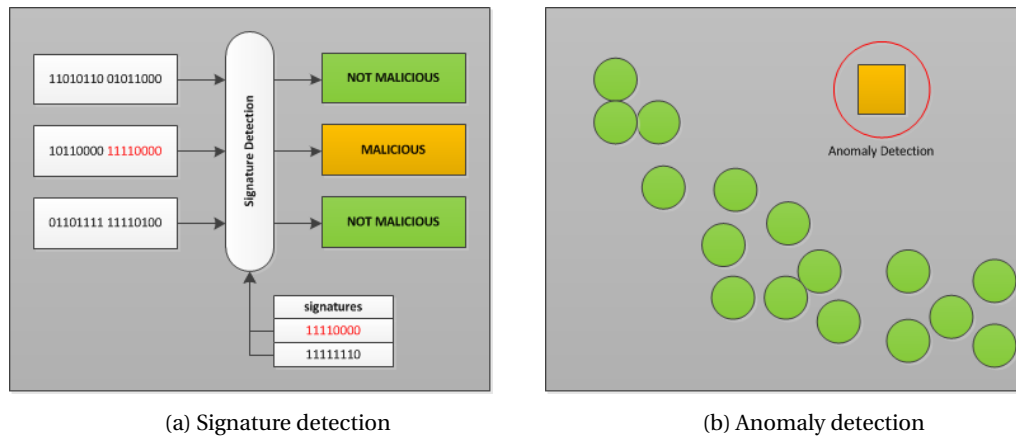


Figure 5.3: Common detection strategies

5.1.3. LIMITATIONS

Intrusion detection systems are not one shot security solutions. As all security solutions they too come with their limitations. To provide a better understanding of such limitations the most common ones are now discussed.

False positives: Detection rates are usually not perfect, this means that some events which are reported to be malicious might not be malicious in nature at all. Although a nuisance this does not have to be a major problem. If however the number of false positives far outrank those of true positives it is easy for real attacks to be missed simply because operators get bombarded with false information. As a result they become "tired" and start to assume alarms are probably faulty. This is especially true if the monitored data streams contain a lot of noise.

False negatives: Opposed to false positive some events are reported as non malicious while in reality they were malicious and should have been flagged. These false negatives do not necessarily translate to a reason for concern, as long as the rate stays within acceptable limits to be set by the operators. If however the rate of false negatives is too high such that significant events are missed this can turn into a serious issue. However there is one risk that is important to be aware of. When malicious events have a chance of being missed by the monitor, this also means that attackers have the chance to get passed security and staying completely under the radar. This aspect is important to take into consideration when deploying such solutions, having an IDS only improves the chance of detection to a certain extent.

Resource limitations: Monitoring systems have to capture, process and store huge volumes of data, preferably in (near) real-time. This can cause the resources made available to the IDS to be insufficient to handle the load, especially when security is operating on a tight budget. There is a direct trade-off between resources made available and the performance and detection rate of the deployed systems.

Encryption: Depending on the streams of data which are being monitored the IDS might encounter encrypted traffic. These encrypted channels cannot be analysed using normal detection strategies and require specialised solutions. This raises a security concern as an encrypted intrusion attempt or breach of policy might be allowed to remain undiscovered until other, possibly more significant, signs are detected.

Liability: Some IDS systems have a prevention component which allows the IDS to control its environment and thus act to prevent a security breach from happening automatically. Such a system is called an intrusion detection prevention system and thus requires additional, often very specific, privileges. If an attacker manages to gain access to these systems, the monitoring platform itself becomes part of the attack vector.

Human operator: Monitoring systems easily seem the shiny new toys within a security strategy, nonetheless, it requires knowledgeable workers to operate. Without the relevant training or experience there are many

mistakes which will lead to the solution to operate on a suboptimal level. This then leads to system with increased rates of false/positives and false/negatives. There are case studies where a party decides to add an IDS solution to their security strategy and after installing a new high tech solution they revert back to thinking they are secure. All the while not reading the logs, ignoring blinking lights and refrain from keeping the system updated. As one security professional put it: "*Smart people defend networks, products don't defend networks*" - Joe McCray

5.2. RELATED WORK

Historically, the origin of intrusion detection systems (IDS) evolved out of a set of tools mostly intended to help administrators review audit trails such as user access logs. In 1987 Dorothy E. Denning published a paper titled "An intrusion-detection model" [90], describing what to this day remains the basis for many monitoring systems. Denning's method was based on statistics for anomaly detection and monitored both user and network level data while operating "independent of any particular system, application environment, system vulnerability, or type of intrusion". The IDS was named IDIES for Intrusion Detection Expert System and ran in production on Sun workstations at SRI International. By 1993, IDIES was eventually upgraded with the NIDES for the Next-generation Intrusion Detection Expert System. Since these early days a large amount of research effort has been invested into intrusion detection systems aimed at monitoring our systems and networks.

Today there is a large body of IDS related research available. Almost all of these focus exclusively on IT based environments and the offered solutions are not directly suitable for ICS environments. This applies even when underlying protocols and infrastructure used are the same [39, 43, 44, 91, 92]. This does not mean though that their lessons should be disregarded as they can offer valuable insights. One such noteworthy example is the common intrusion detection framework (CIDF) discussed by [93] and created by DARPA in 1998. The CIDF work defined an framework for a general IDS based on four modules which together are used to build an IDS architecture.

Public research investigating the challenge of monitoring solutions for ICS environments are not completely absent though. However, almost all of these make use of IT based approaches such as network traffic analysis and packet inspection [94, 95]. These proposed solutions are thus focussing on the protocols utilized and ignore the physical domain entirely. In itself these solutions are not bad, but they forgo the goal of protecting a physical system and lack feedback on what is actually happening to the physical. As such a large and presumable the most important resource is missed.

Some researchers have suggested to utilize this resource and incorporate knowledge of the physical system into the workings of the IDS itself. By understanding the network traffic it is possible to simulate the system to some extent, which could then be used to take the physical state into account [26]. Research investigating a direct tap into the physical state by going directly to the field devices (sensors and actuators) seems to be missing.

Change detection As part of a larger research endeavour into the security of ICS systems, Cardenas came to the same conclusion, that only limited research into ICS security is available and that what is published are generally tweaks of solutions aimed at an IT environment [43, 44, 96]. As such, incorporating knowledge of the physical system might very well trigger a paradigm shift in the sector. This realization leads to the proposal of a linear mathematical model which is used to analyse the actual system and determine if an attack is ongoing. Their main aim being to "*protect the operational goals from a malicious party attacking our cyber infrastructure*", which is a two part challenge: 1) detection of attacks on cyber-physical infrastructures, 2) survival of attacks on cyber-physical infrastructures [92, 97].

For the detection problem Cardenas suggest that when having knowledge on how the output sequences should react to the control input sequence it is probably possible to detect an attack by comparing this expected output with the actual received output signal. The effectiveness of this idea will depend on the quality of the estimated output signal. Further investigating this idea they created a model of a physical system and formulated an anomaly detection algorithm. The formulation of this detection system will now be discussed further.

The real time detection requirement leads to the use of sequential detection theory, which deals with problems where the measurement time is not at a fixed moment. More specifically, the method used by Cardenas is that of change detection, also known as quickest detection, which falls within the domain of optimal stopping problems. Although the security industry usually makes use of a method called sequential detection, the researchers selected this method because it is more intuitive with the detection method. Change

detection works under the assumption that a set of measurements starts out under the normal hypothesis, H_0 . The thought is that this hypothesis will then be rejected in favour of the attack hypothesis H_1 at a certain measurement. To avoid making any assumptions on the probability of an actual attacker their work does not assume a parametric distribution but only puts mild constraints on the measured sequence. The input sequence for the detection strategy is then determined by:

$$z_i(k) := ||\tilde{y}_i(k) - \hat{y}_i(k)|| - b_i$$

Change detection is performed using the following

$$S_i(k) = (S_i(k-1) + z_i(k))^+, S_i(0) = 0$$

Where the decision of an ongoing attack (or lack there off) is made by:

$$d_{N,i} = d_\tau(S_i(k)) = \begin{cases} H_1 & \text{if } S_i(k) > \tau_i \\ H_0 & \text{otherwise} \end{cases}$$

In which,

- \tilde{y} Received (and possibly compromised) signal
- \hat{y} Expected signal
- z Expected value of the random process
- b_i Small positive constant
- S Nonparametric CUSUM value
- d_n Decision rule

This gives two parameters used to tweak the experimental outcome. First there is b_i , which is a value that directly relates to fluctuations in sensory output and thus uncertainty. This parameters can be estimated empirically. The second is τ , for which the researchers found that a large enough value for τ would return no false positives by the strategy. However the time required to detect an ongoing attack would suffer significantly. This in itself is not necessarily a problem however, at least if the attack is detected and mitigated prior the occurring of any adverse affects. As such the final conclusion for the proposed IDS is that while very effective, this comes down to a balancing act between false positives, detection time, and reaction time and requirements of the system itself.

Virtual image Having researched the effects that malware has on industrial control systems, both for the case of IT malware as samples specifically targetting ICS, [26] suggests to leverage available knowledge on the physical system to enhance its protection. Following this suggestion [98, 99] does exactly that by creating an new intrusion detection system which maintains an internal representations of the physical state of the controlled systems in a virtual image. To define and build this virtual image a new formalized language has been specifically defined, which has been named Industrial State Modeling language (ISML) [98]. The virtual image is meant to operate parallel to the monitored system, providing real time insights and analysis. At start-up the IDS will load the systems model from an XML file which comes with predefined settings and values. During operation, the IDS received a copy of the network traffic which is scans for ICS protocols. The ICS packets are then processed and the commands they contain send on towards the virtual image where they are used to update the internal representation of the control system. Each update also triggers a monitoring module that compares the (virtual) system state to a list of predefined critical states. If a match is found the IDS will raise an alarm. Implementation of a prototype and conducting of experiments have demonstrated that the proposed solution is successful, proving the approach has merit.

By adding a multidimensional metric that provides a parametric measure of the distance between a given state and the set of critical states further extends the IDS, giving it the ability to estimate future instability in the system[100]. Conducting experiments using a prototype implementation of the extended IDS demonstrated the improved functionality and that the approach indeed has merit.

5.3. THREAT ANALYSIS

An important process in the design of a new security solutions is to have an understanding of the threats which are being faced. This applies both when designing a new control system for an vessel as it is when creating novel security tools. The threat analysis does just that. Essentially it is a review of factors which are

a threat to a system, available controls to mitigate threats and possible risk associated with a threat. It is an essential step in determining the state of security of a system. This section then aims to provide a high level threat analysis of control systems to gain an understanding of ongoing weaknesses and make a case to build a novel monitoring tool.

The threat analysis will consist of the following elements: Assessment scope to determine which part of the network should be considered, actors and their capabilities to build an understanding of the adversaries, available security controls to map out existing tools and finally vulnerabilities to the system and the risks this entails.

5.3.1. ASSESSMENT SCOPE

At the start of a threat analysis it is important to have knowledge on the assets which are at stake. These assets can be in any shape and or form, be it digital, intellectual or physical. This research looks into the security of control systems and at the center of the discussion will thus be the control network itself. To determine which parts of the network should be included in the scope a look is given at the various segments of a control network, which were previously discussed in section 2.2. This section presented figure 3.1 for the benefit of the discussion, which has been reprinted below as figure 5.4, which identified the following network segments: Corporate network, Demilitarized zone, Process Network(s), Control Network(s), Field Area. Each of these will now be discussed to determine if they should, or should not, be included into the scope of the threat modelling process. By extension this will provide an early indication of potential resources for a new monitoring system. Please note that the reader is referred to section 2.2 for a detailed discussion on control systems.

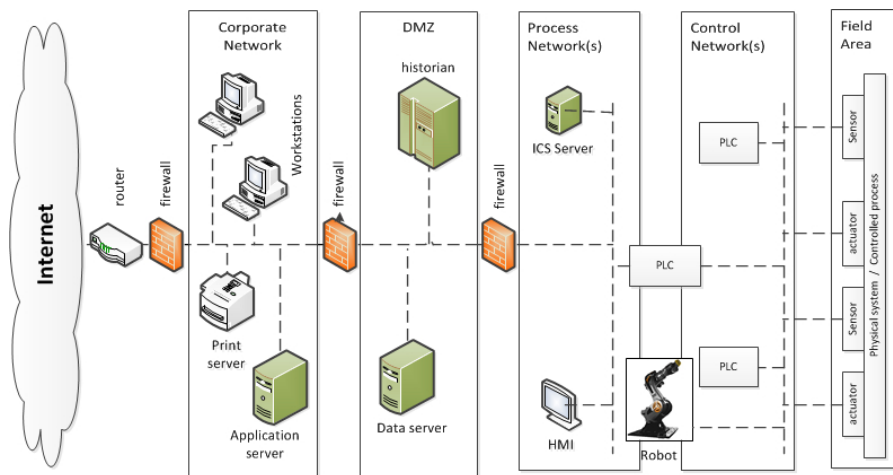


Figure 5.4: Overview of a typical industrial control network

Corporate network When looking into the security of an industrial control systems the corporate network is often assumed to be compromised by security experts. This means that from a security perspective it will be viewed as part of the internet itself: unsecured, unreliable and out of their control. This assumption has the benefit that the threat model and security measures will not required immediate revising the moment it is discovered that the corporate network is compromised. From the onset it was viewed as untrusted. As such the corporate network will not be included into the assessment scope.

Demilitarized zone The DMZ acts as a buffer between untrusted networks and the (secured) internal network. The devices residing in this zone are only permitted limited connectivity to specified devices within the internal network. Although this restricted traffic towards the inner network provides increased security opposed to open access, at the same time it provides a risk. This risk stems from the trusted devices inside the DMZ which are deemed secure and have direct access to internal systems. Compromise of such systems (e.g. due to a badly configured gateway) are thus a perfect vector into the internal network. Due to the servers within the DMZ and potential risks they pose the DMZ will be part of the assessment scope.

Process Network(s) The process network facilitates communication for ICS systems and contains ICS equipment representing the overall state of the process under control. From there it connects to the control network and has influence on the physical processes. These centralized servers clearly means the presence of a single point of failure. This centralization also means that compromise of equipment is made easier as only a limited number of machines should be infected. Incidents in this area can have wide-spread consequences, both on operability of the control network as impact on the physical devices. Because of this the process network will be part of the scope.

Control Network(s) The control network contains the controllers, sensors and actuators which maintain stability of local processes. It also facilitates communication between components, collects information streams from the field area and pushes (selected) data upward into the chain. At the same time the control network is responsible for the distribution of control commands to the actuators, ensuring relevant actions are performed to control the processes. The control network is distributed in nature which makes it harder to secure against incidents and easier for malicious actors to remain undetected. Fortunately this distributed nature also makes it harder to cause wide-spread malicious behaviour. Due to the potential to directly influence the field devices and thus the physical processes the control network will be included in the scope.

Field Area The field area enables the connection between the control system and the physical processes through sensors and actuators. Not only are commands carried out on this level, but it also informs devices residing higher in the chain. Due to the direct influence that these devices exert on the physical world they are an important part of the threat model. Directly compromising them might not always be possible and directly depends on the field device used (mechanical / computerised), but the field area will be able to provide valuable information to help with defending the network. This is especially true when the physical systems is to be taken into account for intrusion detection.

Assessment Scope After looking at each segment of the control network the following areas have been determined to be part of the assessment scope: Demilitarized zone, Process Network, Control Network, Field Area.

5.3.2. ACTORS AND CAPABILITIES

To improve the threat model it is paramount to get insight into the potential threat actors targeting control networks. These actors, along with their capabilities, provide insight into the possible consequences that can be expected if security is breached and the security measures that can be taken to mitigate these. Additionally, insight into threat actors provides a basis for the budgetary requirements of cyber security. The threat actors are separated into two main categories, which are based on their intent. These are those with malicious intent and those without malicious intent:

Malicious intent: There are many actors which have malicious intent, criminals seeking financial gains, competitors trying to gain an upper hand in negotiations or a disgruntled employee seeking to get even. These actors could both be outsiders and insiders. The intent here is to disadvantage those targeted and better themselves in some form.

Non-malicious intent: Those without malicious intent are generally employees or subcontractors and are more often then not oblivious to the consequences of their actions, or they simply don't care. These are the people that bring home routers and connect them to control systems, where the simple motivation is to download the latest software versions from vendors websites, or their ability to freely surf the world wide web during their shift. Their intent is not to cause any harm.

It is important to indicate that the absence of malicious intent does translate to benign consequences. Often the opposite is true, a simple mistake made by a subcontractor could take down a complete control network and put a halt to a companies business operations. Their actions could also provide direct access to those that do have malicious intent in mind, or be swayed to unwittingly help these actors. It is thus important to take both these actors into account. The capabilities that both parties have are overlapping and sometimes cascading, as such these are grouped together for the remainder of this subsection. Those with non-malicious intent often already have access due to the professional roles they play as operators and subcontractors. Malicious actors can gain access through phishing emails, USB attacks or even bypassing subcontractor security systems. Thus the threat model assumes a foothold has been established within the control network.

To cause actual physical damage to a system actors will have to find a way to influence and control the ICS themselves. This translates to full access to the network, the engineering workstations and even the PLCs. This full control has been demonstrated by the Stuxnet malware, traveling by USB and engineering workstations, changing firmware on the PLCs. The capabilities of the actors are as such assumed to be full access, which will be discussed now.

Network traffic Having access to the network enables the sending and receiving of network traffic. This means multiple things from a security perspective. It is possible to eavesdrop on network traffic to gain a better understanding of the system, potentially creating a network map which might aid further exploitation endeavours. It also implies that a man in the middle (MiM) attack could be established to gain credentials and possibly funnel data out of the system. This access also means that any normal network command can be executed which could be anything from an innocent ping message to the active scanning and interfering with network traffic. As indicated prior ICS networks are very sensitive due to their real-time requirements and any interference on the network, even a simple ping message, have been known to cause hiccups or even take down a complete system.

Set-points Having access to the control network enables the changing of configurations and set-points which dictate how the physical processes are controlled. Changing such a value could prevent a safety system from responding when an incident requires it to, or force it to respond while nothing insecure is happening.

Controller logic The logic running on controllers is often installed through engineering workstations connected to the network or by connecting an engineering laptop to the controllers directly. In both instances it is possible that the logic running on this controller is changed in a way other than intended. This could be by a programming mistake made by the engineer or purposefully by a piece of malware which tries to create malicious controllers. The presence of faulty controller logic in a control system can cause any range of problems, from delays during commissioning of an offshore platform to the physical destruction or equipment and injury to human life and environment. Often this malicious logic is hard to detect, not only because the operator can be kept in the dark but also because often they will not surface until the specific action is performed/required.

Human machine interaction As was raised before, malicious controller logic has the potential to send fake information up the chain towards the HMI. If they are fed false information they essentially go blind as to what is actually happening to a system and are unable to perform their jobs: Keep an eye on process stability and interfere if required.

Summary The impact of malicious actors with full access is that even the controller components cannot be trusted to function as their design suggest they should. This includes all digital devices within the control network, from the monitoring systems in place to keep track of the process to the failing of automated safety systems to ensure safety of the process, equipment and operators in case of failure.

5.3.3. EXISTING SECURITY CONTROLS

There exists security controls which can be used to improve security and defend a control network against breaches. Such security controls will be discussed next to gain insight into available technologies, their capabilities and to identify possible gaps they leave open. This reveals what needs to be improved or even missing completely. The discussion will list the most common controls that can be used to secure a control system from cyber attacks.

Firewalls Firewalls are network security platforms which establish a barrier between multiple network areas and control the flow of information between them. Managing this information flow is typically done through a white- or blacklist. The whitelist allows only traffic which has been specifically allowed to go through, while the blacklist allows all traffic to pass through unless it is specifically marked not to be allowed. Standard firewalls verify packets based simply on their headers, however the most advanced solutions offer deep packet inspection techniques. These can perform inspection of the actual payload of the traffic flowing through and understand its contents. Theoretically this would enable certain control packets -a MODBUS shutdown

command for example- to be dropped, while other commands are allowed to pass through unhindered. Installation of a firewall can be done at any location within the network, most commonly however this is done at specific choke points such as into and out of the DMZ. Within control systems firewalls are also often placed in front of field devices. Even the most advanced firewalls are unlikely to stop all malicious traffic from passing through because some traffic has to be considered safe for the control system to maintain operability. Setting a new set-point for example is a normal operational procedure in plants and would thus require free passage. Nonetheless this change can trigger an incident and force the system to move into a critical state. The same applies to updates to controller software, which can sneak malicious versions of the software through. More strikingly, a firewall does nothing to stop infection through USB or engineering laptops which simply evade the security measure altogether.

Intrusion detection systems IDS are security mechanisms which essentially monitor network activity for malicious behaviour which they then report to a designated location. Some IDS are capable to alarm an operator in case of a significant threat or even prevent the flagged behaviour from continuing. These systems are commonly installed near locations where large streams of data are handled, such as the network switches. Detection of malicious behaviour can occur through the use of pre-known signatures or some sort of behavioural analysis, often through machine learning. Intrusion detection systems differ from firewalls in that they evaluate suspected intrusions as opposed to try and prevent them from occurring. There are multiple ways in which IDS can be evaded. The easiest is probably to evade the defence mechanism completely by infecting portable devices which are later connected to the targeted devices. For example the engineering work stations can be used to change controller logic with a malicious version, with no data this will be impossible for the IDS to detect. Another method would be to monitor the network traffic for common anomalies (ICS traffic is often very predictable) and piggy back through the network when a spike is happening. An IDS which uses signatures for detection is easily evaded by using custom made signatures which are guaranteed to be unique. Both anomaly and signature detection can be evaded by executing commands which are not seen as malicious in the first place: Setting a new set-point or sending control packets.

Network segmentation When each and every device is connected to the same network the risk of a single point of failure increases considerably. This can be prevented by dividing the network into multiple segments which only contain a certain set of nodes. When required interaction with other segments can be achieved through a firewall and router which can monitor the communication channel for suspicious traffic. This makes it much harder for cyber incidents and malicious actors to spread horizontally throughout the network when they establish a foothold as they are contained within the sector. At the same time the total network structure is hidden from within a segment, which further enhances the security of a network. This approach also adds other benefits such as a reduction of traffic congestion which directly improves network performance. There are ways through which network segmentation can be overcome however. Dynamic nodes, such as USBs and laptops, that have the ability to move through multiple segments are a potential risk as they can directly defeat the defined boundaries. Misconfiguration of a switch can offer an attacker direct access to all the various subnets and in some cases even claim to be a switch aiming to divert traffic over the attacker's device, often without alerting monitoring systems. There are devices which will require access to multiple subnets, and if where one node holds access to all subnets a single point of failure resides again. It is also important to note that segmentation will not stop malicious behaviour within subnets themselves. While an infection might not easily spread to other subnets, if the compromised subnet is one that contains the ICS security controllers significant problems can still result.

First principles security One control that is often overlooked is that of engineering the physical systems using fail safes based on first principles. Reliance on control systems to ensure system stability in the case of an critical incident should be kept to a minimum and critical components should be supplied with a physical fail-safe device that is not networked. One simple example is to have an overpressure valve in a high pressure pipeline system which will open when the pressure rises just shy of the pipes maximum. This should be an traditional valve which cannot be controlled through anything but mechanical ways, this ensures that tampering with its set-points is not possible without physical access. In a highly dynamic and changing environment the settings for such a device should be set that all possible variations are within its bounds, this to prevent the the operator from continuously having to change the setting: Which in itself can lead to accidental incidents. This approach will increase engineering complexity in some situations and call for creative

solutions but heavily increase security of a physical systems, especially when they are part of critical infrastructures. While safety engineering is an important part of control system design, this is not necessarily the case for security engineering. Additionally with the increasing hyper connectivity the demand for remote operation will only increase and clients might overrule such safety systems in their hunt for ever increasing efficiency. One major problem with this approach however is that many vendors are simply not supplying these old, dumb devices. In many cases they come with a processing unit by default. Circumventing a physically secured system can only be done by having intricate knowledge of the design and first principles which might enable an expert to find security vulnerabilities in the system itself which have been missed, or ignored, by engineers while designing the systems. One of the things that can be learned from the security industry is that there is no such thing as perfect security. Some things are simply not impossible to mitigate. A pressure relieve valve might be installed for example, but as Stuxnet demonstrated effectively the centrifuges operating the flow system will always have a natural frequency which can be exploited to cause resonance damage.

Training of workforce Even the strongest chain is no stronger than its weakest link, and in the field of security this link is often the person(s) operating a system. This is not to say they are to blame, often they are unaware of the security repercussions of their actions and how much negative impact their negligence can have on the whole. A fact happily exploited by many adversaries. This can be mitigated to some extent by training those with access to systems and devices (operators, engineers, janitors) and making them both aware of- and responsible for- their actions. Training removes the "I didn't know" excuse. Although training of people will increase the security rating of any system, we are all human. The psychology field of social engineering is specifically designed to exploit the human mind and trick people by utilizing their emotional responses against them. Kevin Mitnick, probably the world's best known social engineer, said that he has been successful on all social engineering ventures he was hired to execute so far.

5.3.4. VULNERABILITIES

For any network operation having knowledge on possible exploitable vulnerabilities is paramount. While the previous paragraphs will have improved security there are still paths that can be used to cause mischief. This section then discusses the major vulnerabilities indicating a lack of focus on certain elements.

The majority of the security controls discussed focus on the process network and higher of the control network, which leaves the field devices that are interaction with the physical process vulnerable to exploitation. It will be argued that with good security in place on the higher levels this is not a problem, however consider the weaknesses that have been discussed earlier.

Both the firewall and IDS control have difficulty to detect traffic which resembles legitimate commands from the control system and operators but in actuality is malicious. Examples include the changing of set-points or sending control commands to controllers and field devices. This opens up the possibility to send malicious control commands which actively or passively move the system towards a critical state without any alarm bells going off until an incident occurs. Even when these solutions are fully successful in blocking all malicious traffic a simple USB device or engineering laptop would be enough to evade them and compromise field devices.

The fact that current security controls can simply be evaded calls for increased security of field devices. Their vulnerability opens up a host of problems which are hard to solve currently. First is that their settings can be altered without approval, a power plant can be shut down for example if the required commands could be issued. More worrisome is the ability for knowledgeable actors to change the actual logic on these devices, which can be done by piggy backing on an actual software update performed by engineers. The change of code could be detected by utilizing cryptographically secure hashes, however this is hardly ever done in practice. Once a field device has been compromised it is easy for malware to maintain persistence and remain hidden. A compromised field device has direct access to part of the process and since it can be programmed to ignore control commands is not actually owned by the operators who are powerless to prevent malicious behaviour. Except by manually shutting it down.

This is where the importance of physical security is important. Currently systems are not designed to be safe from critical failure using first principles. This means that once a system has been compromised, be it by a knowledgeable attacker or an accidental malware infection, it can be placed into a critical state by interacting with the control system and field devices. When designed using first principles security however critical failure can often be prevented. Graphically this can be seen in figure 5.5, where by analysing the controlled process a certain disturbance might be found which is able to force the process or controller into a critical state.

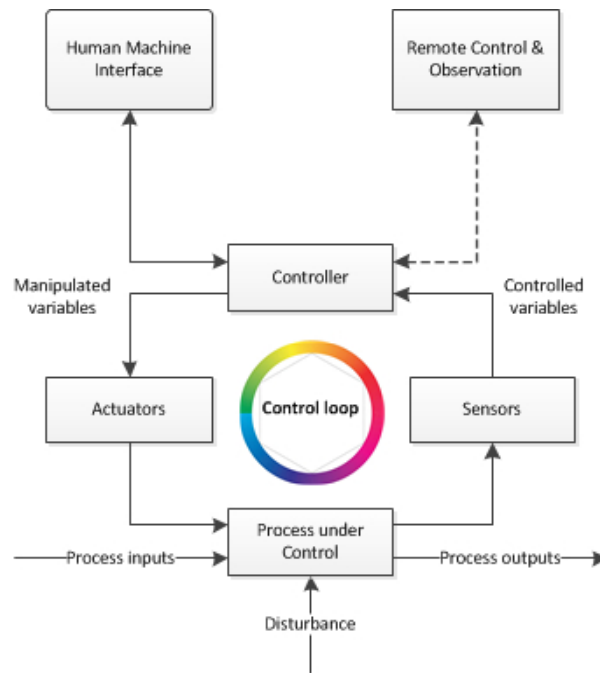


Figure 5.5: Controlled process with disturbance

5.3.5. RISKS

There are various risks associated with a breach to security, which is part of the threat analysis and are thus also listed. Some risks are merely a nuisance while others could prove to be more disastrous. This subsection discusses these risks.

Espionage Having specific information about process parameters can tell experts a whole lot about the details of a process. Not only could this information be used to circumvent security and gain further access, it could also reveal intellectual design decisions used to stay ahead of competitors. It could give insight into the current state of operations and potential ongoing problems that might lead potential clients / buyers to rethink doing business. Espionage as such will not have a direct impact on the daily operations on the short term, but when looking at stability over a longer time span it could well tip balance out of ones favour.

Falsifying process parameters Being able to interfere with the information that is delivered by a controller enables an actor to falsify the information which is provided towards operators and technicians. This essentially places them in the dark as to what is going on. A major risk here is that an unaware operator or engineer might move the system towards an unfavourable state because the information (s)he is relying on it not correct. The same goes for maintenance, which these days often relies on operational hours. If these are artificially increased the component will be replaced quicker then required, thus increasing operational costs, alternatively the hours could be lowered which would increase the risks of a failure. In both cases a cyber incident would not be suspected, every component has a chance to fail and mean times to failure are based on statistics, not exact science.

Delaying business operation Directly building on the previous is the risk that a system can be delayed, sometimes significantly, simply by playing with the operational parameters of a plant. Control systems are designed to be stable in a certain configuration, but often little consideration is put into "what if" scenarios. Creating an artificial ghost in a system can cause downtime while nothing is wrong, forcing maintenance engineers to search for problems that might not even be there, they might even replace components which are perfectly fine. An example would be sensors reporting states that are outside of the physical operational domain (negative volumes for example), which might indicate faulty sensors and thus warrant replacement. The benefits to this delay can be found by competitors seeking to delay production, negatively impact reputations or cause financial loss.

Critical states The largest risk faced is that of the process moving from a stable state into a critical state, possibly even without the operators being aware of anything happening. This has the potential to not only cause damages to the equipment itself, but harm people and planet in the process. One hypothetical problem could be the interference with a automatic hopper overflow mechanism on a dredger which is filling its hopper with heavy soils. If the pipeman¹ fails to detect the issue at hand and stop the process in time, or worse is not able to because his controls are disconnected from the controller logic, the ship has a serious risk of sinking. This however provides a good example of first principles security: Ensure a full hopper cannot possibly sink the vessel.

5.4. FIRST PRINCIPLES MONITORING SYSTEM

As stated in the introduction to this work a major focus in mechanical/structural engineering is on failure. Designs are often based on years of failing things, which is unfortunately hardly the case for security. The first principles monitoring system, or IDS, aims to make use of this failure knowledge and the requirement for control systems to perform a failure analysis which returns expected and potential modes of system failure. Physical knowledge of the system state can then be used to detect behaviour that is out of the ordinary of even moving towards failure.

Such an IDS has the potential to be much more than an intrusion detection system, it has the ability to act as a secondary security system that alarms operators when problems arise. This might sound redundant to some, however in certain industries this added value is more than worth it while in others it is considered standard practice. Take Airplanes for example where three completely separate (and differing) control systems are used and a majority vote is used to select the correct course of action.

This section will discuss a novel first principles monitoring system. However, before directly continuing with this design its objectives and the assumptions will first be discussed. These will be followed by a look at the decomposition of an control network to gather possible design possibilities, to be wrapped up with the proposed design.

5.4.1. DESIGN OBJECTIVES AND ASSUMPTIONS

The first step in looking into a new intrusion detection system is to identify what its objectives are. This then enables the validation of a prototype by checking it if functions as expected, requires tweaking or fails miserably to meet expectations. The objectives have been split into two separate parts: The primary objectives are aimed to detect malicious or anomalous behaviour of the control system. The secondary objectives are there to expand upon the intrusion detection and possibly instigate a move towards a security platform. Also discussed are the assumptions which lie at the base of the design.

PRIMARY OBJECTIVES

The primary objective of any intrusion detection system is to detect when malicious activity is present within a system. The aim for the proposed solution is to do as such using the physical world manipulated by the control system. To accomplish this the following three primary objectives have been set:

1. **Ensure value consistency:** Ensures that the output generated by field devices is the same as the data which is utilized by the control system and reported to the operator. When a compromised controller tries to game the system by changing output this should be reported.
2. **System specification verification:** Verifies that the controlled process operates within the system specifications and boundaries determined by the operators and asset owner. This includes knowledge on physical boundaries and raising an alarm when any are (about to be) crossed.
3. **Critical state detection:** Verifies that the controlled process only operates within (predetermined) safe states and triggers an alarm when the systems transitions into a known critical state.

Fundamentally these objectives thus encapsulate the possible course of action a compromised controller has to negatively impact the system, as was also demonstrated by the Stuxnet malware. These objectives do not include the verification of the logic residing on the controllers and indeed aim to detect malicious intent without having to mainly because Cardenas has very successfully demonstrated the success of such an objective previously [92].

¹The specialist operating the dredging equipment aboard a dredger.

SECONDARY OBJECTIVES

Having a technologically innovative monitoring system might be alluring, however businesses still require a reason to start incorporate such a solution into their daily operations. As such the secondary objectives aim to provide other benefits and thus offer an incentive to business owners.

Safety improvements: Since the aim is monitoring by observing the physical processes the system can likely be utilized as a second safety system. This enhances safety performance has the added benefit that an installation will be more appealing to potential clients.

Forensic data on incidents: When (cyber) incidents occur it is often hard to termine what actually happened because systems are not designed to facilitate forensics. A monitoring system which offers forensics has provides added benefits to prevent incidents from happening again, independent of malicious intent or not.

Scalability: Due to their technical requirement modern intrusion detection solutions require a moderate amount of resources, for example to perform deep packet inspection or to simulate parts of the physical system. Minimizing resource requirements would enable scalability of the developed solution and ensure the proposed concept can be used by large ICS deployments.

Upgrades without compliance issues: Patch management and performing upgrades are non trivial actions within the ICS domain. By having a system that can be upgraded without resulting in compliance or balance issues will increase the appeal of the solution. At the same time the concept should be able to run independent of the ICS such that upgrades to the monitored system do not required a complete reconfiguration of the IDS solution.

Extendibility: A monitoring system which is extendible with other detection modules designed by passionate security professionals has the potential to turn into a security platform. The ability for an platform to easily extend itself with more advanced technology, while not requiring core replacements makes it a more robust solution for the future. Additionally this promotion of collaboration is expected to have a positive effect on the monitored networks.

ASSUMPTIONS

Throughout the design process some assumptions were made with respect to the control system being monitored. These were made mostly because the specific information required was not available or unspecified. However it is important to be aware that malicious actors having knowledge of the these will try and violate them to undermine design functionality [96]. As such they should be kept to a minimal and where possible hardened such that violation is discouraged or prevented. The following paragraphs will now discuss the assumptions used to propose a first principles monitoring concept.

Availability system information Proper operation of the monitoring system will depend on knowledge which is available concerning the control system. It is assumed to all engineering, operational and safety information is shared and known.

Data path The field device data entering the monitoring system does not pass through devices which could be compromised. Where this to be the case it is impossible to guarantee integrity of the data.

Field devices The most important assumption is that about the field devices. It is assumed that the field devices which feed the monitoring system are analog, which makes it virtually impossible to compromise them through digital means. However realistically it is increasingly difficult to purchase such devices as even the most basic temperature sensors will come with a full suit of digital tools and options. When an facility is equipped with digital sensors the assumption is that the most influential² of them can either be replaced with analog versions or that analog versions can be added to the system. Otherwise they are considered to have their security hardened and are exceedingly hard to compromise, which can be achieved through verification

²Not every sensors will be able to cause a physical break down of the system when compromised

of the cryptographic hash of their firmware, disabling remote settings / update changes and following best practices.

Violation of assumptions Attackers with knowledge of these assumptions are most likely to try and violate the field devices assumptions. The simple reason for this is that the field devices are the connectors between the physical process and the monitoring system. When it is possible to influence those operators will be unable to trust the monitoring system and revert back to potentially being kept in the dark.

5.4.2. CONTROL SYSTEM DECOMPOSITION

The objectives which should be fulfilled by the new IDS have been identified, leading towards the next step in the design process: Decomposition of the control system and identification of components which can be used in the detection process as assets. Figure 5.6 represents a process which is operated by a control systems and represents the main components which are commonly found. Based on the threat model some of these components are not as trusted as they were initially set out to be. These are represented in red, whereas those that are trusted are represented by green. From this it can be seen that those components which are assumed

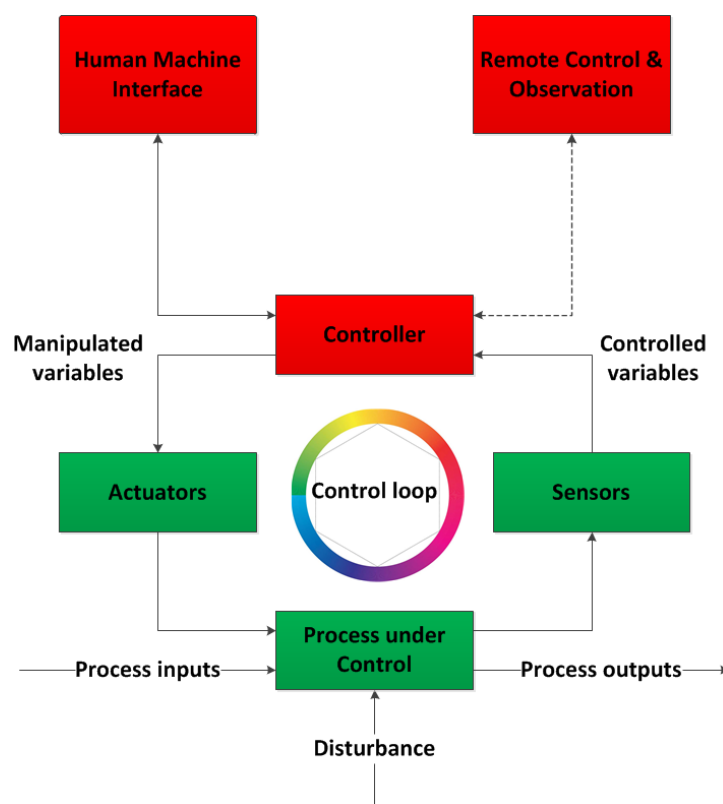


Figure 5.6: Level of trust for control system components.

trusted are the actuators, sensors and physical process itself. The trustworthiness is based on an assumption as has been discussed previously, however it is important to stay aware of this. The decomposed control system with its trusted components then gives insight into the requirements which are needed to meet the primary objectives:

1. To detect any deviation between field devices and the control system there will have to be input gathered from both sides.
2. Deviation from system specifications is only possible if the specifications are known. Which was presumed to be the case.
3. The system moving into a critical state requires a definition of what constitutes a critical state. These then have to be supplied to the IDS through external means. In addition to the system moving into a

critical state, machine learning might be used to create what are "default" states and the paths that lead there. This requires machine learning to be implemented, but requires no additional external input.

4. A controller can usually be divided in two components: an estimation algorithm to track the state of the physical system given y , and the control algorithm which selects a control command u given the current estimate. || Use this (from cardenas)

The trusted elements, along with the requirements, will provide a solid base towards meeting the objectives of a physical monitoring system.

5.4.3. CONCEPTUAL DESIGN

The related work revealed that there is only limited ICS intrusion detection research available which take the physical processes themselves into consideration and use the state as its main stream of detection formation. The available research however has shown promising results utilising this concept. They however appear to acquire the required process information from the network, which makes them dependant on the specific control protocol(s) used by the specific system. This immediately presents a difficulty in a field with a broad choice of often proprietary protocols. More so when the view of the field is extended to take all automation field into consideration (e.g. building automation).

Basing information input on the network side also presents an possible problem, which is that one has to assume the information received to be correct. The threat model presented in section 5.3 indicates that this is not necessarily a valid assumption however. By design ICS networks don't offer much in the way to ensure that the traffic adheres to the CIA triangle of "Confidentiality, Integrity and Availability", data is easily faked. Malicious actors could simply be sketching a picture that all is well, when in reality, it is not. This leaves a security gap where exploiting the physical process itself is concerned.

Current solutions require the simulation of a physical model. Not only does this require the presence of such a model, it can also put a serious constraint on available resources. Especially due to the real time requirement of ICS. By extension this reduces the ease of scalability to larger systems.

This work aims to work around these problems and starts from the assumption that the controllers can not be trusted and might indeed be under control of malicious actors which aim to keep operators in the dark of their activity. Not only is control traffic on the network hard to understand due to control protocols and the actual data itself is not integer, little gathering opportunities remain available for input towards the IDS. As is also clearly shown in the overview of figure 3.1.

Figure 5.6 zooms in on the connections between a PLC and the rest of the system; On the one side the field devices while on the other the control network. This essentially makes the PLC an information broker between the field devices and the larger control network, modifying, merging or even holding back signals. Where only the field devices can be trusted³. The proposed concept then is to have the IDS retrieve data from the last line where information can possibly be gathered: The last line, the field devices and processes themselves. Not only does this fill the gap where the threat model has indicated that the network and controllers cannot be trusted, but it utilizes information from the physical process.

The design of the proposed principles intrusion detection system is seen in figure 5.8. As is clearly seen the concept takes samples from the i/o signals which are being transferred between field device(s) and controller(s). These are not diverted but mirrored towards the IDS as input, this prevents any possible interference with the original signal path. The figure also demonstrates the potential to add monitoring points at other locations in the system, such as on the networking side of the controllers, but this could just as simple be the addition of extra sensors or the monitoring of traffic to the historian or operator control panel. This choice will depend on the specific situation (e.g. client requirements) and possible detection strategies, which will be discussed further down.

After a sample has been taken by the monitor it will be send forward for processing by the IDS. This process will be done by so called "detection strategies", which can subscribe to a certain set of monitors and only get fed the requested data. Not only does this method motivate a modular approach, it also prevents every strategy from having to analyse all the traffic. This flexibility enables improved performance and targeted detection. This work presents multiple detection strategies, however the thought is that offering a modular approach enables the security industry to pitch in and further improve upon the concept. At the same time the system has the potential to be expanded into a larger platform that includes safety, monitoring of wear and tear, and other modules.

³As was discussed in the section on assumptions.

After the detection strategies processed a sample they deliver an advisory to a decision unit within the IDS which proceeds to either do nothing or raise an alarm. The basic logic used by this decision unit is that an malicious advisory results in an alarm. However the operator might well want to add more complex logic to this, depending on the used detection strategy, sensors being monitored, and other factors. The goal is to provide flexibility to work with different modules. The complete process which is employed by the IDS is presented in a flowchart by figure 5.8. The detection strategies will be discussed next.

DETECTION STRATEGIES

At the heart of the proposed IDS are it's detection algorithms, strategies used to determine if nefarious activities are present within the control system. As such they form an important part of the conceptual solution. This subsection presents the strategies used by the first principles concept.

Consistency comparison This compares the output value from field devices to the value which is known by the control system. Assuming that the PLC and/or control system is not tampered with these values should be equal. Any deviations are thus a basis for alarm and further investigation, indicating a malfunction or deliberate action. Deciding if the values are indeed consistent is determined by:

$$y = y_{field} - y_{ICS}$$

The decision of an malicious sample is then made by:

$$m = m(y) \begin{cases} \text{Negative} & \text{if } m(y) > 0 \\ \text{Positive} & \text{otherwise} \end{cases}$$

Graphically this is represented in figure 5.9.

Value analysis In addition to a basic consistency check, instrument readings are compared against the device, component and system level specifications, describing the minimum/maximum operating context for specific components or the rating alarm and trip setting (RATS) list for the entire design. Think for example of the flow rate in a pipeline, which the control system only monitors for a lower bound value - no flow for example. There is also an physical upper limit though, which could be the maximum capacity the pumps can sustain. Any deviations from this are flagged for immediate investigation, especially if combined with a consistency comparison alarm as this indicates a potential compromise of a PLCs integrity.

Signature analysis Industrial control systems run very structured processes, onto which a form of signature analysis is applied. The value under consideration and the context in which it appeared is matched against a list of logic rules and reference traces, raising an alarm when deviations are encountered. This rules can take any form, as long as they can be specified in a machine-readable and -interpretable format. Three possible signatures are (a) timing analysis of ICS control packets (since PLCs will show a different answer time and deviations from their otherwise exact response patterns in case they are executing different software branches than usual), (b) request-responsive-sequence analysis of packets (PLCs communicate in set intervals status messages to which other devices then react), and (c) power-trace analysis of PLCs (as PLCs execute other code branches than usual their relative power consumption over time will change). Figure 5.10 shows a schematic representation of this method.

Envelope escalation As in control engineering the various states of the system, the failure domains and safe operating conditions are well defined, it seems possible to utilise knowledge of the system's secure and insecure states to follow the actions and reactions in a multi-level multi-dimensional model, thereby creating a safe-state envelope. Envelopes have been used for dependability engineering of communication networks to provide hard performance guarantees during challenge events[101, 102], however the method may directly be applied to control engineering as well. Each independent subcomponent of an ICS is described by one or more envelopes, which are defined by a set of metrics assessing a particular component from various angles. Each envelope hence captures an N-dimensional state space, which is annotated based on the system specifications which operating context is safe or not.

Figure 5.11 shows this concept in a two-dimension plot for 2 independent metrics, with green indicating a safe operating context, red an unsafe system condition and yellow an operation outside norm values

which may be temporarily acceptable or after a legitimate operator override. As can be seen in the figure, the IDS system monitors the development of the system's status based on the envelope specification and tracks whether it can still be considered safe. As commands are issued and controls are actuated that would take it into an unsafe condition, an alarm is generated. This tracking and detection can be done in two ways: First, many control processes are well defined, i.e., it is possible to determine before-hand how say a pumping process is expected to change given a particular change in input variables. Second, in case such information is not directly modelable, it is usually generated and available after the testing period during a system's commissioning, during which normal and various boundary cases of a system are being tested.

Both cases will let the IDS directly flag a command as abnormal to the operator, and in case an actor conducts a previously unknown attack a comparison with historical commands and system responses will help the system maintainer to at least identify, where and how things went out of the ordinary with concrete pointers on how to roll back.

MAIN BENEFITS

The main benefits expected to be gained by using this novel approach are as follows:

1. **Extensibility and Multi-Vector Detection.** The system may be extended to include more sensors to accommodate evolving attacks and new vectors, including sensors not connected to the ICS system such as a microphone listening to the acoustics of machinery, sensors reading power usage and output of equipment, or even radio frequencies readings. Stuxnet for example caused damages by changing the spinning speed of centrifuges, causing mechanical damage. While a microphone could have easily recognized such changes, it is exemplary for additional types of sensors that are (normally) not providing input to an ICS system, but would help within the context of the proposed IDS to detect abnormal events. Making it significantly harder to launch attacks that would go unnoticed by other off-the-ICS sensors.
2. **Unmodified signal path.** Existing network-based IDS are located in between devices (here the controller and the PLCs) to scan and block malicious traffic. As scans however change the latency of communication messages, may in some cases change packet order and since the blocking of select packets within a larger control packet train may cause significant side effects, control engineers voice concerns about placing such devices inside a production environment. As the proposal system does not interfere with IDS communication and control messages are interpreted off the bus, timings, packet order and the integrity of packet trains remain unmodified.
3. **Ability to detect compromised ICS infrastructure.** As the Stuxnet malware compromised both the control system as well as the PLCs reading and responding to sensor values, it was able to send back falsified sensor readings and remain unnoticed while bringing the ICS outside of the safe operating context. An IDS system reading both the state of actuators and sensor readings at the last line (which are analog voltages) and comparing it with the readings reported by the regular control infrastructure has the ability to detect malfunctioning or purposely compromised equipment. While this would seem like a duplication of the control infrastructure, the additional expenditures for such an approach are actually minimal: they simply require a single microcontroller per group of sensors and actuators digitizing analog voltages and reporting them cryptographically signed via Ethernet to the IDS.
4. **Upgrade without compliancy issues.** As no changes to the existing ICS infrastructure is necessary, the approach would allow for an effective upgradability of existing legacy systems. Note that since nothing is placed into the signal path that may intercept or alter its behavior, no compliancy issues or the necessity of re-certify the system would arise which would make a roll-out within certain critical infrastructures very expensive or time consuming.

POTENTIAL DRAWBACKS

As with arguable any security solution the approach will not offer absolute security, which is simply not possible. This paragraph discusses the main draw backs that cling to the proposed concept.

- **Shift of security assumption.** The assumption that the network is safe is shifted to the field devices. Is is likely a matter of time before these get targeted. The NSA for example has already demonstrated their intention and ability to modify server equipment with malicious chips en-route to customers.

- **Installation process.** Setting up a system where each field device, or a subset, is monitored requires extra wires and work to implement. It is expected that this will add more costs to the implementation than adding an IDS to the main switch with mirroring techniques.
- **Computational resources.** Due to the sheer amount of field devices present in larger facilities the required computational resources to keep the IDS working on an acceptable pace might be significant.

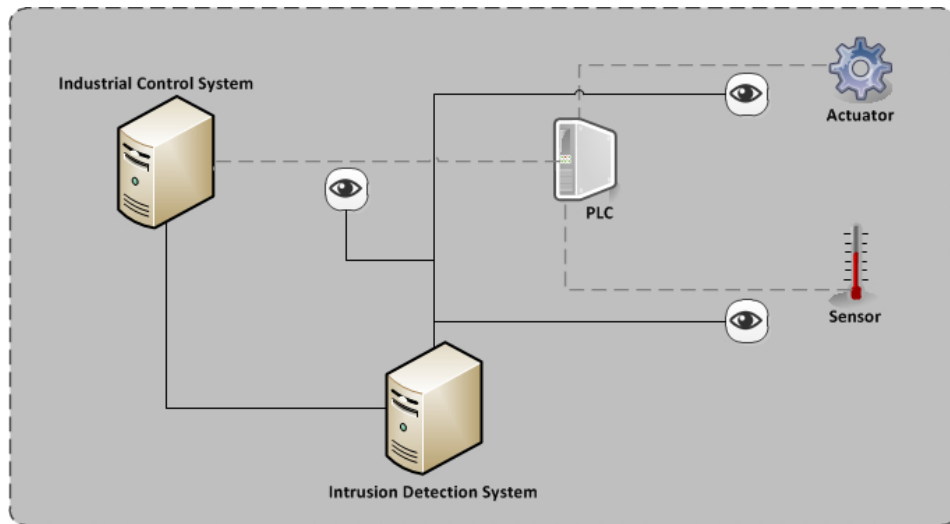


Figure 5.7: Conceptual design of an intrusion detection system based on process information

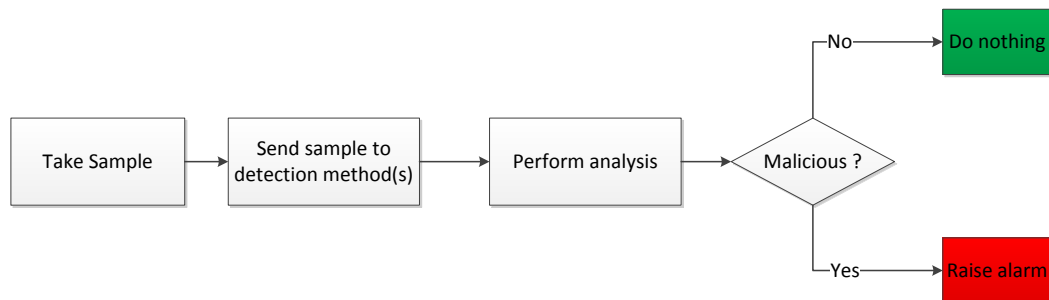


Figure 5.8: Process steps of the concept.

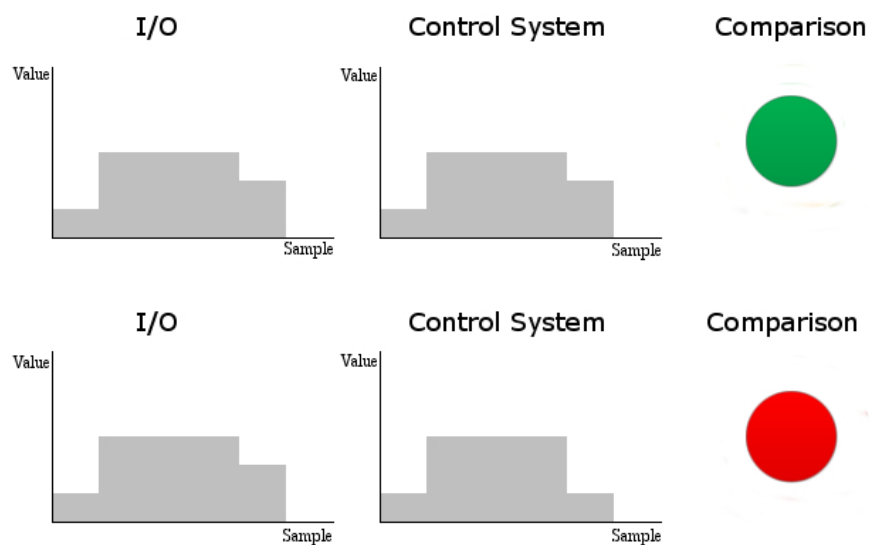
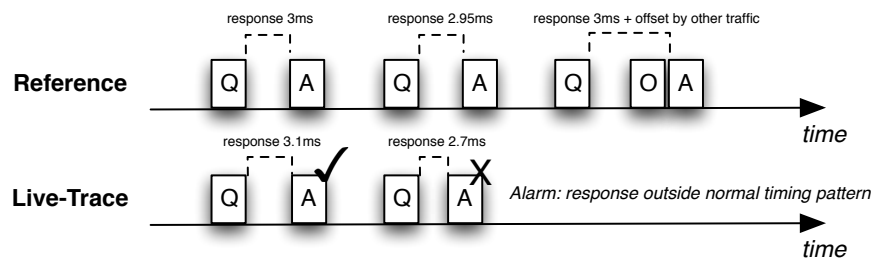


Figure 5.9: Consistency comparison

Timing Analysis



Power Analysis

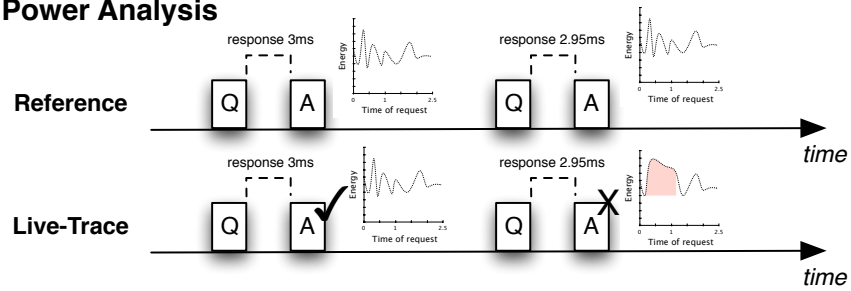


Figure 5.10: Proposed signatures: Timing analysis and power analysis

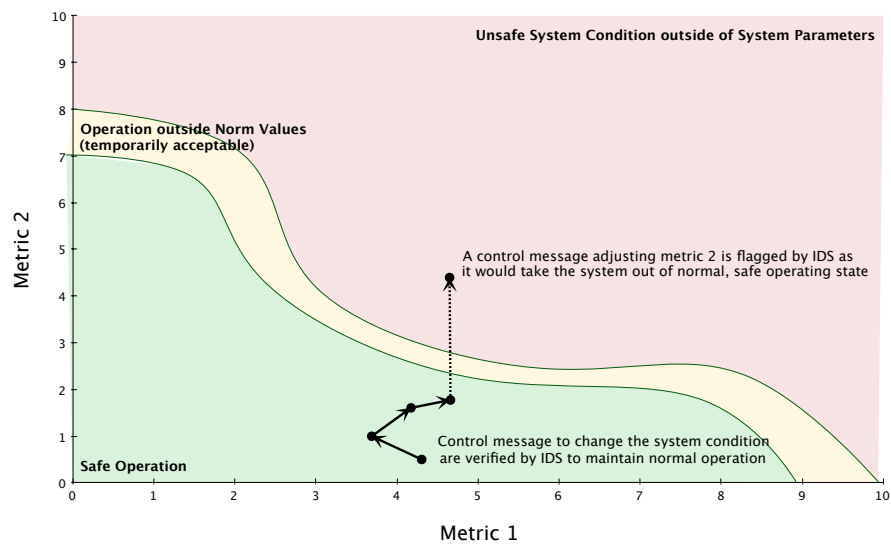


Figure 5.11: Envelope escalation

6

BUILDING A PROTOTYPE

A concept for a novel intrusion detection system based on the physical process has been introduced in the previous chapter. Proposing a novel IDS concept may lead to critical discussion and insights, but without experimental data a critical assessment of the concept will be difficult. Experimental data cannot be collected until a prototype is build such that validation experiments can be carried out. The results of which yields data which can then be used for the critical assessment of the proposed concept. Such a prototype has been build and is the topic of this section.

The main purpose behind such a prototype is to enable feasibility and verification experiments with respect to the proposed concept. Having an operational prototype will provide new and deeper insights into the concept which enables evaluation data for the concept and makes it possible to determine to which extend it has merit, needs further improvement, or should be scrapped.

Prior to starting implementing the prototype, or even finishing it's design, considerations are made with respect to the requirements, basis of the design and how various components will communicate, which are followed up by the design of the prototype in section 6.1. Here the internal behaviour and structure will be further elaborated. Now that the design has been finalized the actual implementation will be discussed. Section 6.2 discusses the UML diagrams for the main software components. Although the experiments -and their results- will be discussed in the next chapter, a feasibility study will be presented in section 6.3 to get insight into the functionality of the prototype and also serve as a Go/No-Go moment for the IDS concept.

6.1. PROTOTYPE DESIGN

Before starting on the implementation of the prototype it is beneficial to first come up with a design. This will make implementation easier and eliminates design mistakes early on. Because a design can be started from scratch but also from a frame work the building blocks to use in the prototype design will first be discussed. After laying this groundwork the design structure and it's intended behaviour will be discussed.

BUILDING BLOCKS

The first step in the design process is to get an idea for the basic building blocks which are to be used, if any. Such blocks will be discussed here.

Framework The framework of the program indicates the philosophy and strategy used to create the prototype. Although it is possible to simply start implementation without thought on any foundation, it is very likely that this becomes messy quickly and lead to tape upon tape to keep the contraption together. The alternative is to use a strong foundation which aims to support the basic design requirements for the prototype: modular and scalable. In their work on intrusion detection systems [93] presented the the modular common intrusion detection framework, or CIDE. This concept has been shortly discussed in the related work and it's architecture is represented by figure 6.1. Utilizing this framework as the foundation for the prototype offers multiple benefits:

- Implementation: clear and modular structure to the program making modifications to existing code simpler.

- **Scalability:** At the same time this basis enables a design which is expected to be highly scalability because each module, or so called block, could be run on a separate thread, processor or even system.
- **Extendibility:** The modular approach presented by the framework extendibility due to it's modular design. Adding extra input signals requires only the addition of an extra analysis block, while adding a new detection strategy can be done by adding another analysis block.

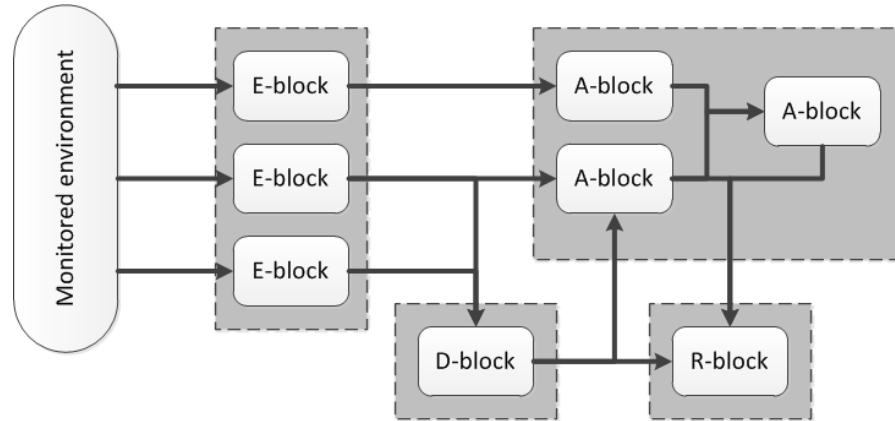


Figure 6.1: General CIDF architecture for IDS systems.

Communication channels The inter-block communication presented by the CIDF -the arrows in figure 6.1 indicate the possible flow of information through the framework. Facilitation of this communication can be done either through point to point connections or through a message bus. Figure 6.2 shows the former, while figure 6.3 represents the latter approach. The requirement for the IDS concept is to be scalable and modular. This would make point to point communication a tangle of lines rather quickly when the application scales, increasing complexity. This can be seen in figure 6.2b. Extending the system with increasing analysis blocks or detection strategies would get complicated really fast. Adding extra blocks to a message bus however will not increase complexity, as can be seen in figure 6.3b. This improves scalability of the design while at the same time maintaining its flexibility and modularity. The major drawback being that the maximum possible throughput of the message bus could pose a bottleneck. For the prototype it is expected that this will not pose a problem however, while for bigger systems the IDS can and possibly should be split into a decentralised approach.

An additional benefit of working with the message bus is that it is much easier to modify and extend the platform with future blocks. Adding specified analysis blocks for example becomes as easy as connecting the new block to the message bus, other connections are not required.

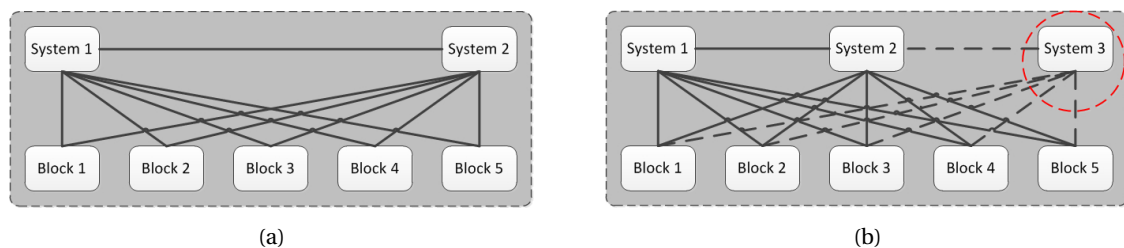


Figure 6.2: Blocks that use point to point connectivity

STRUCTURE

With the basic building blocks in place the structure of the prototype can be build. The CIDF specifies four main classes, namely the event-block, analysis-block, database-block, and response-block. These blocks communicate with one another by making use of the message bus. Combining these components together now yields the architecture for the IDS prototype software, which is shown in figure 6.4.

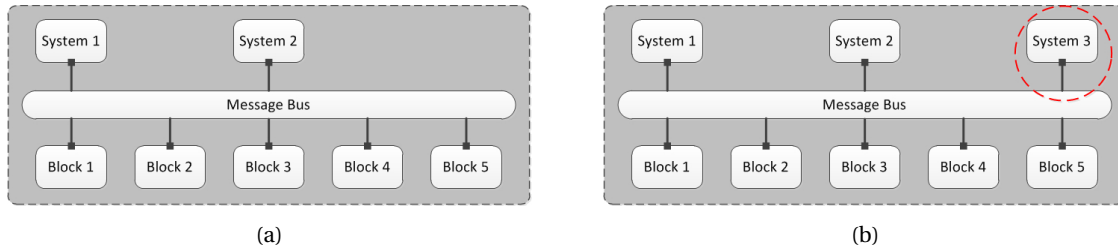


Figure 6.3: Blocks communicating through a message bus

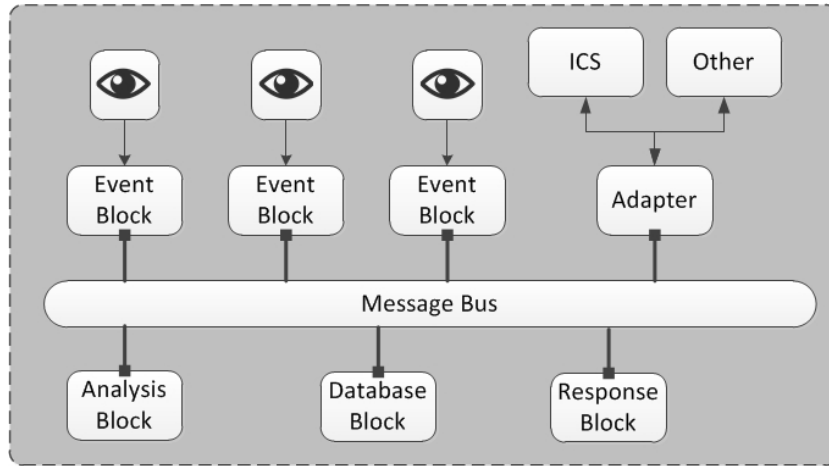


Figure 6.4: Prototype design of the IDS concept.

This structure puts the message bus at the center of the design, which operators based on a list based publisher/subscriber basis. This means a list of subscribers is maintained by the message bus. Each message that is being send then contains a publisher status and is consequently send to each subscriber enlisted to that publisher. As such the message bus is responsible for the delivery of messages to the correct subscriber. The alternative would be to have each subscriber receive every message and let the subscriber determine what to do from there. In the list based system the message bus can be optimised to deal with this overhead and thus reducing the subscriber overhead when many message are being send. Which is expected from the IDS system due to the large number of field devices in a control system.

Each component in this design will be running on it's on thread, making this a multi-threaded design. This has the benefit that once the message bus delivers a message it can directly continue with the next message as opposed to waiting until the prototype finishes processing that one sample before continuing. The same applies to all other blocks, which when multiple analysis blocks are being used has the benefit that slower detection strategies can be run parallel to faster ones and even be divided into various ICS subsystem. Another benefit is that this decentralised approach makes it possible to optimise the resources being used and even spread operation over multiple systems.

6.2. PROTOTYPE IMPLEMENTATION

With the prototype design discussed in the previous chapter the next step is the implementation of the prototype. This will be the topic of this section. Implementation of the prototype is based on the prototype design shown in figure 6.4. This design overview only incorporates those classes of the prototype which are most essential to the prototype operation. The remainder of this section will discuss the specifics of the core classes of the design. These are, as also represented in the diagram, the following: Message bus, Event block, Analysis block, Response block.

6.2.1. MESSAGE BUS

The message bus is the main communication channel between other classes and is responsible for the delivery of messages send on the channel to the correct subscribers. As was discussed previously the implemen-

tation follows that of a list based published subscribe mechanism. To make this happen the class maintains a list of subscribed classes to specific publishers in the "subscriptions" attribute. When a new message is published on the bus the class ensures that it is delivered to the subscribers listed within the subscriptions list. In principle all messages will be treated by a first come first served basis, however to accommodate future priority traffic (e.g. for highly critical ICS subsystems and components) a priority lane is provisioned for. Table below represents the class diagram for the message bus class. Further details are provided in Appendix B.1.

MessageBus	
-	subscriptions:Map<String, HashSet<MessageListener>
+	subscribe(MessageListener subscriber):void
+	subscribe(MessageListener subscriber, String publisher):void
+	unsubscribe(MessageListener subscriber):void
+	unsubscribe(MessageListener subscriber, String publisher):void
+	publishMessage(Message message):void
+	publishMessage(Message message, boolean priority):void
-	messageDeliveryHelper(Message message, boolean priority, String publisher):void

6.2.2. EVENT BLOCK

The event block provides the connection between the field devices being monitored and the intrusion detection systems. As such these instances are seen as the eyes and ears of the IDS. Once new information is received, mainly due to taking an sample, the event block will wrap that information in a message container and publish it to the message bus for further processing. In the case of production this class can also be used for mapping purposes between field values and units understandable by the detection strategy. Further details on implementation are provided in Appendix B.2.

EventBlock	
-	ID:long
-	monitoredDevice:FieldDevice
-	name:String
-	priority:boolean
-	publisher: String
-	sampleRate:int
-	sampleValue:double
+	connectFieldDevice(FieldDevice fDevice):void
+	setPriority(boolean priority):void
+	setSampleTime(int sampleTime):void

6.2.3. ANALYSIS BLOCK

The analysis block is the heart of the IDS. It is subscribed to event messages and is responsible for detecting potential malicious behaviour. This is done by adding specific detection strategies, such as those specified in previously, to the block. Once an event message has been received it is passed down to the added strategies, the result of which are added to an analysis report that is then wrapped in a message and published on the message bus for further response. Further implementation details are provided in Appendix B.3.

AnalysisBlock
<ul style="list-style-type: none"> - mBus:MessageBus - mQueue:BlockingQueue - values:Map<String, Map<String, Double>> - timestamps: Map<String, Map<String, Timestamp>> - sourceElements:ArrayList<String> - strategies:ArrayList<DetectionStrategy>
<ul style="list-style-type: none"> + addDetectionStrategy(DetectionStrategy strategy):void - addSourceElement(String sourceID):void + getSampleValues():Map<String, Double> + getSourceValues():Map<String, Double> + getTimestamps():Map<String, Map<String, Timestamp>> + getValues():Map<String, Map<String, Double>> + messageHandler(M message):void Timestamp requestedTimestamp, long tsDeviation):boolean

6.2.4. RESPONSE BLOCK

The response block is tasked with handling the reports made by the analysis blocks. It determines the response when malicious behaviour is detected, which ranges from waiting for more information, reporting to the operators or in high risk cases even place the control system into a secure state. The latter will be out of scope for the current prototype version however. Table below represents the class diagram for the message bus class with more details provided in Appendix B.4.

ResponseBlock
<ul style="list-style-type: none"> - mBus:MessageBus - mQueue:BlockingQueue<Message> - logs:ArrayList<Log>
<ul style="list-style-type: none"> - updateLogs(AnalysisResultMessage message):void + void messageHandler(M message):void + getLogs:Arrayist<Log>

6.3. FEASIBILITY STUDY

The feasibility study is intended to determine early on if the proposed IDS has merit and further development would be warranted, or that the proposed solution would have to undergo significant changes. To this end a proof-of-concept experiment has been developed, implemented and evaluated. This has the additional benefit of returning insights into the testing process for future evaluation.

This experiment is based on the seawater cooling system of an actual high voltage direct current (HVDC) converter platform, currently being installed in the North Sea. Such a platform converts power from nearby offshore wind-farms to higher voltages, thereby significantly reducing transportation losses, and then facilitates Offshore-to-land transportation. As the experimental environment is subjected to intentional tampering of the control systems, it is for security reasons not running at this stage on the production system but in a simulation environment, which follows the design specifications and blueprints of the platform. Figure 6.5 contains a concept overview of such a HVDC platform and its connection into offshore wind-power distribution.

An important -and mandatory- phase during the design of an offshore platform is the risk assessment, where the risk levels of (sub)systems and the possible consequences are assessed. Targeting the cooling water system could prove disastrous on a platform wide level. Not only does this system provide cooling water for the converters but it forms a backup for the fire-fighting systems. Although prolonged loss of the cooling system is not necessarily cause for panic, critical temperatures can easily be reached if operators are kept in the dark on what is happening. The remainder of this section will further elaborate on the source model, followed by experiments aimed at evaluating each detection strategy.

Important to note is that this feasibility study contains three experiments, while four conceptual detection strategies were mentioned. Due to the nature of the physical model, namely a digital representation, it unfortunately not possible to gather insights into this strategy. This is because this strategy requires actual real world inputs, or complete modulation of those properties which would make the model exceedingly more

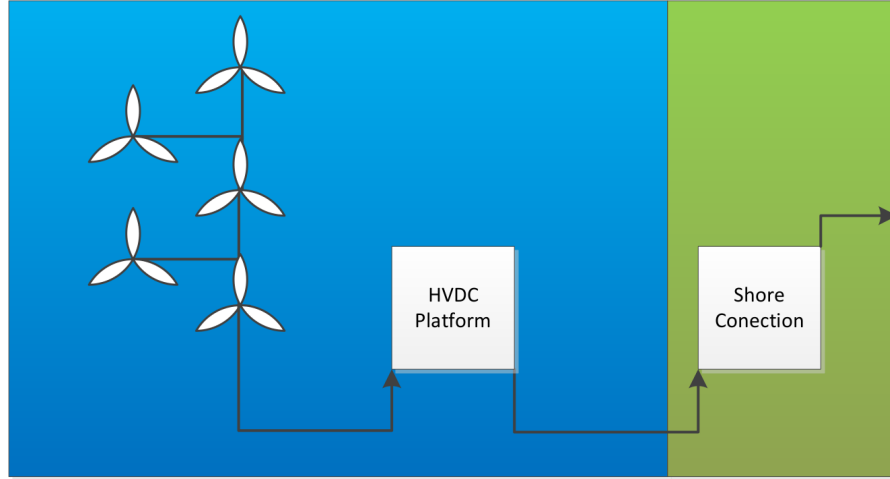


Figure 6.5: The HVDC concept

difficult. Also, this is a method for which there already is some basic understanding with respect to: Its functionality, the false/positive rates, and the methods weaknesses. This taken together warrants not spending time on it within the feasibility and proof of concept phases and make it fall out of scope of this project.

HVDC COOLING SYSTEM

The feasibility model is based on the sea water (SW) cooling system from an offshore HVDC platform. While the cooling circuit itself has multiple purposes (e.g. hydrochloride generation), this model only takes its main goal into consideration: Cooling the fresh water cooling system, which in turn cools the electrical equipment aboard the platform that receives alternating current from wind farms, and converts it into direct current before transporting it onshore via subsea cable. Failure to cool such systems could potentially prove disastrous for the operational continuity of the platform itself and potentially its safety.

Presented here is a simplified version of the actual SW cooling system and the overall IDS prototype, yet sufficient to get early feedback with regard to the applicability of the physical IDS concept. The cooling system consists of two temperature sensors, one flow sensor, four SW pumps and four heat exchangers. Figure 6.5 provides a diagram of the implemented system, its components and connections. Two elements of control are part of this model. The first being a sequence controller for the pumps, which rotates them over a 24 hour time frame. The second responds to the difference between the seawater temperature entering the system (T_{in}) and that being returned to the North Sea (T_{out}). For this controller the following surprisingly simple control design specification rules apply:

- If $T_{in} - T_{in} < 5$ degrees: a pump is turned off.
- If $T_{in} - T_{in} > 10$ degrees: an additional pump is switched on.

Additionally there will be at least one pump running at all times, and at most two. For the simulation context, the temperatures of the incoming seawater are based on average North Sea water temperature overlayed by a sine function to add fluctuating cyclic noise. For the evaluation the cooling demand by the transformers is set to a constant at the average design specification.

Evaluation For the evaluation, we set the cooling demand by the transformers to be constant at the average design specs, and assume that an attacker following a Stuxnet-like (see 4.2.2) approach has compromised the ICS control system and is able to let the PLCs report back false sensor readings and incorrectly switch pumps to reduce the cooling capacity of the system to overheat vital components. The temperature and flow sensors as well as the pump switches are also additionally connected to the IDS, which will compare this information against information it passively reads out from the source model.

EXPERIMENT I: VALUE COMPARISON

The first experiment aims to demonstrate the value comparison strategy as described in section 5.4.3. This was accomplished by letting the system components report deviating values to the ICS controller than what

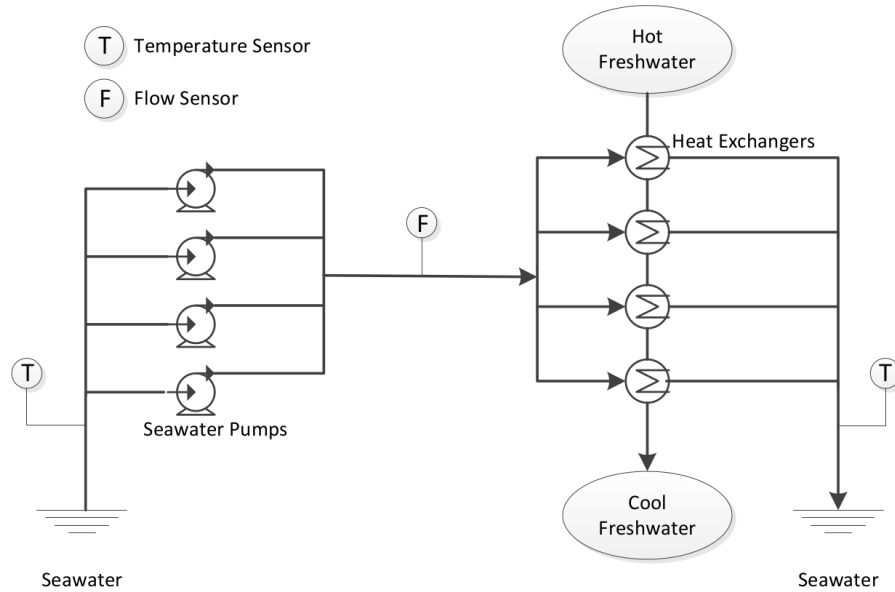


Figure 6.6: Source: Basic model.

was reported by its instruments and is also read out by the IDS itself. The deviations are inserted at a random time and with a random fluctuation, and it needs to be marked by the IDS as malicious when a deviation between the values is detected. This was chosen as it exemplifies the worst case scenario, as only single incorrect values will be more difficult to spot than a long term deviation in sensor readings.

For this test, the system was run for a total of 16605 time-step sensor reports, of which 1672 updates were falsified by a simulated attacker. In its base configuration the prototype was flagging 63% of all tampered values as malicious. The reason that about a third of falsified sensor values were missed can be attributed to the fact that the IDS samples sensor values at a different and lower rate than the ICS, hence some random values had gone unnoticed. While it should be pointed out that in practice this signifies the minimum detection accuracy (as for a suitable attack not individual sensor values but entire sequences need to be consistently modified to mask an ongoing problem), the system should be updated to sample independently of the ICS and whenever sensor readings are reported by the PLCs. The deviation however also highlights that in practice a tuning between false warnings and the IDS's alertness needs to be made, as in principle an IDS can be set as alert to immediately detection any change at a cost of false positives which might on the long term undermine the operator's trust in such system. Good heuristics to automatically determine a suitable error margin while generating close to 0% false positives is hence taken aboard as ongoing initiative for future work.

EXPERIMENT II: SIGNATURE ANALYSIS

The second experiment aims to demonstrate the Value analysis strategy as described in section 5.4.3. In the evaluation, an attacker may let the ICS switch pumps off and as a result the flow past the sensor will read zero flow but a compromised PLC will indicate normal operation. The platform's design specifications list minimum and maximum system conditions in the RATS list, which are already loaded into the IDS. A "Low-Low alarm" listed on the RATS list for the control system indicates such a boundary, which upon triggering should be immediately investigated. During this experiment a total of 5941 time-steps were run, of which 405 were malicious. A tampering of a flow sensor following boundary conditions was able to be correctly identified.

EXPERIMENT III: ENVELOPE ESCALATION

The third experiment aims to demonstrate the feasibility of the envelope escalation strategy as described in section 5.4.3. This is accomplished by putting the SW cooling system in an known insecure state, outside of the safety envelope. Again, sensor values are changed with random input. In this experiment a total of 27150 events were processed, of which 8471 were modified in a malicious way. Similar to the results in the first experiment, we see a detection ratio of 2/3, which can be improved by increasing the sampling speed and sharpness of the alerts.

DISCUSSION

During the feasibility study multiple feedback points and questions arose which can be used to further improve both the IDS proof of concept as the validation model and procedure. The discussion here then is aimed at getting to these lessons learned, establish their root cause and utilizing these to drive improvements.

Enabling the proof of concept quickly indicated that the sample rate which it uses to take samples from field devices can be very important, more so with a discrete and time-step variant source systems. Say for example a sample is taken every (real) minutes, but the attacks manages (by luck, or design) to stay between these samplings. To what extend can this still be detected? By which strategies? At the same time, even when an anomaly is detected the accuracy setting is very important. If any deviation of an exact value is to be taken the amount of false/positives will increase. If this accuracy is too wide though the number of false/negatives will increase. This would require calibration by an expert, however this might be a place to make investigate the use of machine learning.

The creation of the model quickly indicated that time would become an issue as the IDS is working with sample rates, as opposed to packets delivered in a queue. This also brings up the point of discrete time and how to speed up real time to keep simulation from taking forever. Additionally the program architecture of the model itself was based on actual components, thus having a pipe class, a pump class and so on. This presented a lot of unnecessary problems in relation to connecting the components and how they should interact with each other. This led to the idea to combine the components as classes concept with a more mathematical approach for the validation.

Some of the mentioned challenges could have been prevented and overcome before the feasibility experiments were running by simply spending more time preparing and designing the feasibility phase. Which is probably the best lesson for the future: Spending the time on preparations and contemplating on the approach to be used is gained back later during the implementation and execution phases. Then again, there is no better teacher then falling and standing up again.

7

EVALUATION

The concept for a novel IDS system have been presented and a prototype has been build. This chapter provides an evaluation of this prototype. What is currently missing is an evaluation of the new system. Although a simple feasibility study has been conducted while building the prototype this only offered enough information to get insight into the big picture, identify problems early on and identify potential evaluation challenges. This chapter aims to change that by providing metrics on the operational results of the IDS concept, which it does by evaluating the prototype.

Evaluation of the prototype requires an strategy and an model. The strategy (Section 7.1) describes the overall approach which will be used to determine the prototypes performance, where as the model (Section 7.2) is the physical system which is monitored by the prototype. Section 7.3 discusses the specific experiments which are performed to obtain raw performance data. This data is then processed and evaluated in section 7.4, where suggestions for improvements will also be provided. These improvements have been implemented and exposed to the same experiments as the initial prototype, the results of which are discussed in section 7.5. Section 7.6 discusses the performance of the prototype, to what extended the prototype is able to detect malicious behaviour on a control system and suggest potential avenues of future work.

7.1. STRATEGY

The evaluation of the prototype performance requires a strategy. This strategy defines the metrics, methods and testing environment which is to be used for the evaluation process. Cardenas [96] published a framework for the evaluation of an IDS, however the aim behind this framework is to help operators select which IDS out of a selection has better performance for their specific use case. It is aimed at comparison. Since the prototype is still in development it has not reached a stage yet to fully compete with other solutions. Otherwise there exists no security model for the evaluation of an intrusion detection systems[96]. As such another approach is required.

The remainder of this section will discuss the used approach, along with the following required elements: *a lab environment*, *evaluation metrics*, and *the experiments*. The strategy and all elements are represented by figure 7.1, which depicts an overview of the complete strategy.

Traditionally the challenge to evaluate the performance of an IDS is solved by using tools from other fields (such as medical, or machine learning). The most common evaluation tool used is that of *binary classification*, which classifies samples into one of four possible states. These states are then used to calculate the performance metrics used to evaluate the performance of the prototype. Binary classification will be further discussed in section 7.1.3.

The next step in the evaluation strategy is to create a physical source system (lab environment) which can then monitored by the IDS prototype. To evaluate the IDS, this source system will have to be subjected to various cyber incidents and attacks. Compromising this source system are three experimental situations that exploit ICS related vulnerabilities, which have been exposed by the threat analysis in section 5.3. The main intent behind the experiments is to undermine the physical stability of the source system. Because the IDS can work with multiple samples rates, each experiment will iterate over a range of sample rates for every seed. The exact number of simulations will be determined by stabilization of the average system outcome, to be discussed later. After a simulation completes its run the results are processed by a binary classifica-

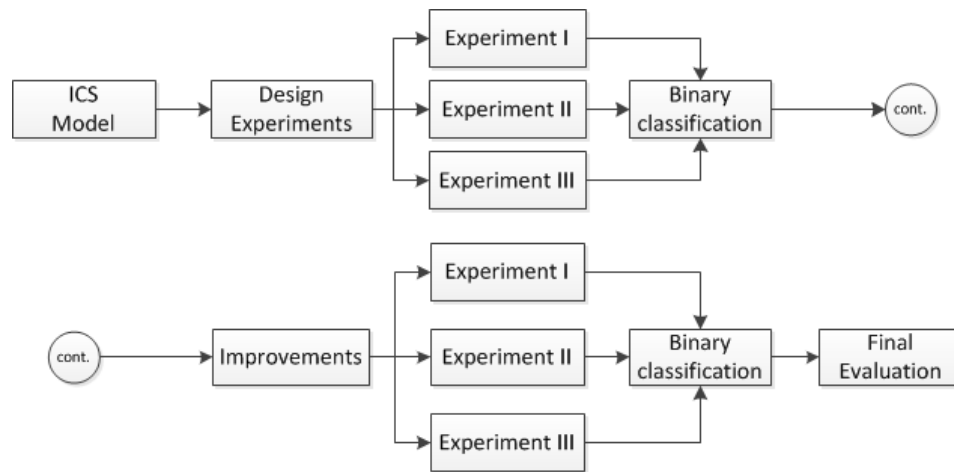


Figure 7.1: Evaluation strategy

tion module which will produce the evaluation metrics. Every iteration will be linked to a randomized seed number such that the exact details of the attack can be modified, yet returns repeatable results.

These metrics are then used to create a graphical representation of the results by plotting the specific metric versus the sample rate. This graph then represents the performance of the IDS detection strategy and its relation to a varying sample-rate. Following these experiments and their evaluation are modification suggestions to improve performance. This modified prototype is then itself re-evaluated utilizing the same experiments. Last the results will be compared and discussed. This strategy is graphically represented by figure 7.1, the remainder of this section will discuss each stage in more detail.

7.1.1. LAB ENVIRONMENT

The evaluation process requires a control systems that can be monitored IDS prototype and is subsequently exposed to both incident and attack scenarios. There are three possible approaches here:

- **Real world system:** For evaluation purposes a real world system would be the best testing environment, for example the HVDC platform as discussed in the feasibility study. However challenging, testing on a operational system takes away any doubt that a solution might not work in the real world. At the same time it would be likely raise issues that a laboratory or digital model might miss.
- **Laboratory environment:** The second possibility is that of a laboratory environment which simulates a real system. The availability of a realistic lab is vital to support the design and testing of new solutions countering various attacks against control system[103]. Such an lab would support the application of active penetration testing, which is especially dangerous when conducted directly on operational production systems. This availability thus has a major influence on developing security solutions and plays an important role in the validation of theoretical models[43, 104]. Unfortunately there are only a few publicly available lab environments because operating such systems is an expensive endeavour. This is especially true when compared to IT where one can often fall back on virtual machines.
- **Digital model:** Because the IDS detects samples based on the physical behaviour of the system, this makes it possible to create a simulation of a physical system and use that simulation as the source system as the experimental environment. Interaction with the physical state of the simulation would be done through a logical controller that as part of the simulation and mimics the functionality of a PLC.

For evaluation purposes the best testing environment would be a real system, for example the HVDC platform as discussed in the feasibility study. Such an approach however is simply impossible, running experiments in an operational environment is not only a huge security liability, but would also require system downtime and delays to normal business operation. The second option is to go with a laboratory environment. In the early stages of this work a collaboration with an external research facility was being discussed, which offered a lot of potential. This facility had access to the required hardware/software and could provide basic training

requirements to get the evaluation up and running. Unfortunately this did not result in the sought after collaboration, largely due to communication challenges. This twist of events has lead to the search for an independent evaluation solution which is found in the digital model. Using this approach the dependence on external parties was removed. At the same time this approach opens up a point for future discussion on using a physical experimental environments when the prototype has proven successful. The simulated source model used for evaluation will be discussed in section

7.1.2. EXPERIMENTS

The ICS threat model indicated that the controllers are a vulnerable part of the control system. They can be exploited by changing their logic or by preventing communication. This can lead to a range of unexpected and unwanted responses by the physical system. When such an event happens they are in most cases not correctly attributed to cyber incidents which makes detection of a malicious controller more difficult.

The experiments are designed to take advantage of these controller vulnerabilities and trigger anomalous process behaviour. This behaviour should then be detected by the IDS prototype which is designed to detect anomalous physical behaviour. The controller vulnerabilities can be separated into various categories which have led up to a specific experiment. These experiments are discussed further down. The aim behind the validation experiments is to provide insights into the effectiveness of the prototype and demonstrate that the proposed concept is a valid method to detect malicious behaviour in a control environment. The experiments are conceptually explained below, while a detailed discussion is provided in section 7.3 on page 87.

- **Cyber incident** This experiment is based on the possibility for operators and engineers to accidentally introduce faulty logic into the control system. Although their intentions are good, the consequences of such a mistake can be severe and move the process into a critical state. Such an event is called a cyber incident and is classified by its non malicious intention. The goal here is thus to introduce an mistake into the control logic that moves the system out of predefined normal operating conditions, which should then be detected by the *value analysis* strategy.
- **Manipulation attack** This experiment exploits the possibility that an actor manipulates PLC control logic. The goal is to move the system into a critical state and cause physical damage, while remaining undetected. The attack will do this by obfuscating the operators vision through falsifying data. The actual attack will -unwittingly- be executed by the operator or ICS as they initiate a specific action while responding to the spoofed state of the process. In actuality this action then causes the process to move into a critical state. The *consistency analysis* strategy is the detection strategy indicating obfuscation of operator data.
- **Active attack** This attack exploits the same possibilities as the manipulation attack and also aims to move the system into a critical state. The difference here is that the actors do not try to remain undetected. Their goal is to cause significant physical damage without any care for being found out. This means that the code can be extensively changed, field devices directly controlled from the malicious PLC and external controllers fed faulty information or even control commands. Where possible this will be executed on field devices which are not specifically part of the HMI, say frequency drives attached to other process and control values. The *Envelope analysis* strategy is the detection strategy raising alarm when the process deviates from normal operation.

7.1.3. EVALUATION METRICS

The IDS prototype has a discrete flow of information as input which can be defined as a vector of samples: \mathbf{X} . Each of these samples is either malicious or non malicious. The output vector \mathbf{Y} generated by the IDS then indicates if an alarm should be raised for a sample, or not. Let then note the following:

- M denotes whether a given input, \mathbf{x}_i , is malicious (denoted by M) or non-malicious (denoted by $\neg M$).
- A denotes whether a given output, \mathbf{y}_i , raises an alarm (denoted by A) or not (denoted by $\neg A$).

Sample, x_i , in the input vector $\mathbf{X} = \{x_1, x_2, \dots, x_i, \dots, x_n\}$ then leads to the following state:

$$\mathbf{X} \rightarrow \{M, \neg M\}$$

$$\mathbf{Y} \rightarrow \{A, \neg A\}$$

These two vectors can be used to calculate evaluation metrics by making use of a technique called binary classification, the standard method to evaluate performance of intrusion detection systems.

BINARY CLASSIFICATION

In binary classification a rule is used to separate a list of elements into two distinct sets, often marked as positives and negatives or simply 1's and 0's. While evaluating an intrusion detection system however a more descriptive label for these are *malicious* and *non-malicious*. In a perfect world the ruling process will not make any errors and place every element into its corresponding subset. Realistically the ruling process used by the prototype is likely to have erroneous judgement. Such non-perfect judgement can have multiple reasons but is most often the result of a sensitivity setting within the IDS. In the prototype it is expected that the sample rate will be one such parameter. This then gives rise to two types of possible errors, named Type I and Type II errors. Together with correct judgement these can then be used to build a confusion matrix, displayed in 7.2, which lists the four possible outcomes for each input element: *True positive* (TP), *False positive* (FP), *True negative* (TN), and *False negative* (FN). More specifically:

- True positive: Malicious sample is correctly identified

$$P_{TP} \equiv P_R[X = M|Y = A]$$

- False positive: Non malicious sample is incorrectly identified (Type I error)

$$P_{FP} \equiv P_R[X = \neg M|Y = A]$$

- True negative: Non malicious sample is correctly rejected

$$P_{TN} \equiv P_R[X = \neg M|Y = \neg A]$$

- False negative: Malicious sample is incorrectly rejected (Type II error)

$$P_{FN} \equiv P_R[X = M|Y = \neg A]$$

	Sample condition	
	Malicious (positive)	Benign (negative)
IDS Outcome Malicious (positive)	True Positive	False Positive (Type I)
Benign (negative)	False Negative (Type II)	True negative

Figure 7.2: The confusion matrix

This classification system is used to determine metrics such as the the true positive rate (TPR) and positive predictive value (PPV) in pattern recognition. The field of medicine on the other hand is interested in the TPR and true negative rate (TNR). For the prototype evaluation the true positive rate and false alarm rate are of interest, which are discussed in more detail below.

- **True Positive Rate (TPR)¹:** The TPR represents the percentage of malicious samples that were correctly identified as malicious. This relates to the rate of "missed" malicious samples and the rate of raised alarms compared to actual malicious events. When there are a lot of missed malicious events the IDS will not be functional and fail at its core design intentions: The detection of malicious events.

$$TPR = \frac{TP}{TP + FN} \quad (7.1)$$

¹The TPR is also known as sensitivity or recall

- **False Alarm Rate (FAR)²:** The FAR represents the rate of non malicious input elements that are incorrectly identified. These are the samples that will raise an alarm while there should be none. When there is an overload of false alarms this can overburden the operators and move them to simply ignore all the alarms, assuming faulty detection.

$$FAR = \frac{FP}{FP + TN} \quad (7.2)$$

These metrics are a representation of the performance profile for the prototype. However there is no specific bar that is seen as an acceptable operational level because this would not be a realistic approach to determine a successful performance. A successful performance will heavily depends on the demands from the deployment location, especially when separate subsystems warrant a distributed monitoring solution or different configuration details. For example, an elevated rate of false positives might be seen as acceptable in a critical environment when this means that all the malicious samples are correctly identified. The aim for the evaluation then is not to apply a pass-fail test to the binary classification but instead discuss the prototype performance in relation to the binary classification method.

7.1.4. CHALLENGES

Throughout the evaluation process various challenges arise. These challenges, and how to manage them, are discussed next.

CONTINUOUS INPUT: SAMPLE RATES

Intrusion detection systems using the traditional approach of evaluation network traffic, such as the concept proposed by [96], make use of individual network packets to feed the IDS. Obviously this approach does not work for the prototype because it is based on continuous signals from field devices. This makes evaluation of the IDS based on the traditional method of detection rates for individual packets more challenging. The detection mechanism of the prototype is based on samples, which are taken at a specific interval. This interval, however, means that malicious events can be missed completely for the simple reason that there is no sample. This means that this interval time, or inversely the sample rate, is an important parameter in the evaluation process. A slow sample rate, such as displayed in figure 7.3, increases the chance that malicious events are missed by the prototype. Simply increasing the sample rate will not necessarily improve the prototype performance as a sample might still be missed, as displayed in figure 7.4, or falsely identified. It is expected that there will be a trade off between slower and faster sample rates. Evaluation of this expectation is done by including the sample rate in the testing phase and performing multiple experiments for a range of sample rates. The detection rates can then be plotted against the sample rate to verify the dependency expectation.

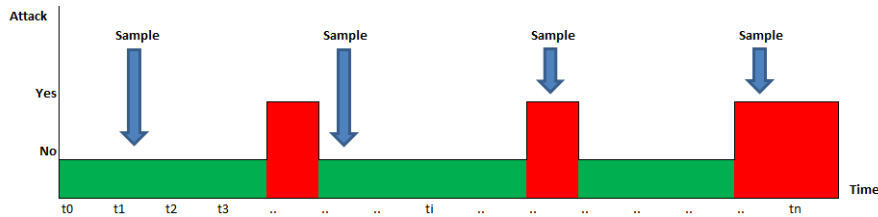


Figure 7.3: Low sample coverage

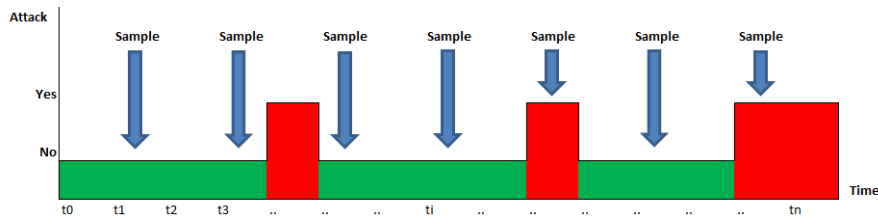


Figure 7.4: High sample coverage

²The FAR is also known as the false detection rate.

It has to be noted that this discussion indicates that an attack sequence might not be detected for the simple reason that the sampling time was malconfigured or simply not at a sufficient level. On the other hand the attacker might have knowledge on the IDS configuration, which is not altogether unlikely as demonstrated by Stuxnet. This makes it important that operators choose a sample rate falling well within the response rate of the process being monitored. Such a sample rate would guarantee that a malicious sample will be sent to the monitoring system before physical damage occurs.

SENSOR ACCURACY

A challenge which arises due to the use of field devices for information is that of sensor accuracy and possible deviations from the actual process state. This is especially true for the dredging practice where sensors often report incorrect value's. The accuracy and deviation depends on the quality of the used field devices, but is also affected by the operational environment and wear of the devices themselves. When field devices are of an analog nature, as assumed by the prototype design, the reported value can be influenced by interference effects on the wires. The worst case for the detection strategy is when the reported value to the ICS deviates from the sample used as IDS important. The detection strategies can take this into account by having a deviation range which specifies an acceptable range of acceptance. This deviation is expected to have an influence on the performance of the IDS because a non malicious sample near the detection threshold can be moved over this threshold by a deviation of its sensor value. The deviation on field device values has to be included into the source system to mimic a real physical system. This will be accomplished by adding a random fluctuation to the samples obtained from the field devices.

A benefit of the prototype monitoring system is that it can be extended to detect device failure. Once a field device nears the end of its life it often starts to demonstrate behaviour and fluctuations that fall outside of normal operations, these could be monitored and replaced during the next scheduled service moment. The effect of this is a decrease in system downtime and better usage of equipment.

DISCRETE TIME

A challenge encountered during the feasibility experiment is that of discrete time versus (real) system time. The IDS itself is designed to run in real time, i.e. on the CPU clock. The source system on the other hand is modelled with a time step to simulate a physical system. This time-step can be made to mimic real time, but also to increase simulation speed. The challenges surfaces once the source system is running different from real-time because the prototype is only capable of taking samples in real-time. In other words a sample time of 1 second for the IDS might translate to 1 hour passed within the source model. This has major consequences for the evaluation as 3600 samples are missed due to a timing mismatch. Figure 7.5 illustrates this with one signal.

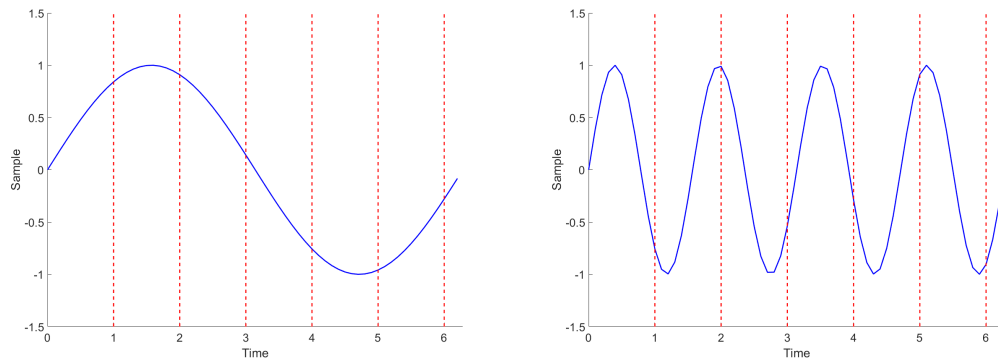


Figure 7.5: Importance of sample-rate.

This challenge is resolved by providing an optional argument to the source of the prototype which accepts a clock instance for simulation purposes. When this optional argument is not used during the prototype start up phase the internal system clock will be used, as would be the case in normal operation. When the prototype is instantiated in a simulation environment and thus provided with the optional clock instance that clock will be used. The simulation clock in turn is maintained by the source model, making the IDS oblivious to a real or simulation environment. This change to the prototype however requires a one time update to all methods such that they operate the same independently from the clock used.

After code refactoring the simulator is able to increase simulation speed, removing the limitations of real-time only processing and enables the building of simulations which might otherwise last hours.

EXPERIMENTAL SAMPLE SIZE

An important part of the experimental phase is to ensure a statistically significant result which reflects the performance of the prototype. This means that a large enough sample size is required and thus multiple iterations for each experiment have to be run. Every iteration will have a different seed which will have a random influence on the system and experiments. In the social sciences the required sample size is determined by statistics and the researched populace to get a good representation of the whole. Unfortunately this statistical approach to determine the required sample size beforehand is not possible with the prototype. This is because the prototype is not a processing a standard group of samples out of which a random drawing is made to represent the whole.

Another approach is required to determine the required sample size. The determination of a proper sample size has to be done in an iterative manner, where after every iteration run the results are checked on their stability. Once the results have stabilized it can be said that a sufficient number of seeds have been evaluated. Determining result stability is done by looking at the moving average.

Moving average Say that every experiment returns a result R_i where i equals the seed or iteration. After three iterations there are three results: R_1, R_2, R_3 . From these the average after seed 2 and seed 3 can be determined by:

$$A_1 = \frac{R_1 + R_2}{2}$$

$$A_2 = \frac{R_1 + R_2 + R_3}{3}$$

This represents the moving average of the experiment and can be plotted against the number of simulations run to obtain a graphical representation. Once a sufficiently stable state has been reached the simulations can be stopped. This process can be put into a mathematical context using the following equation:

$$A_n = \frac{\sum_{i=0}^n R_i}{n} \quad \forall n > 1$$

Extra benefits can be retrieved with respect to the stability when the delta between A_i and A_{i-1} are determined, mathematically written as:

$$D_n = |A_n - A_{n-1}| \quad \forall n > 2 \quad (7.3)$$

This delta can be seen as a direct representation of system stability. Once below a threshold of 5% the results can be said to be stable for further processing.

Simulation state When using this method it is important to realise that while the moving average might be converging to a stable value, the actual simulation output could in actuality be diverging. This has been represented by figure 7.6. As such both the rolling average as the actual output should be taken into consideration when determining that enough samples have been taken.

There is an additional benefit to this, which is that if instability is detected in either the rolling average or the simulation output something is wrong with the prototype or source model and further improvements are required.

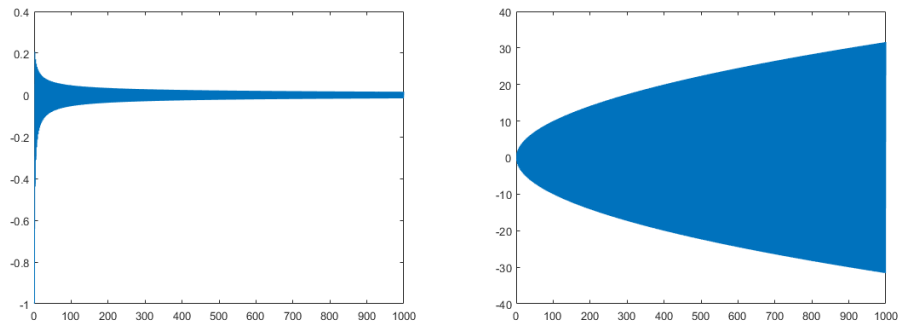


Figure 7.6: Left: The rolling average is converging to a stable situation. Right: The system output is diverging.

7.2. SOURCE MODEL

The source model will be based on a simplified version of a workhorse in the dredging industry: the trailing suction hopper dredger (TSHD). The TSHD is selected because it directly relates to the offshore and dredging industry. The TSHD concept will first be explained to present the reader with a basic understanding of such a vessel. This will then be followed by a discussion on the simplified TSHD model to represent the source control system.

7.2.1. TRAILING SUCTION HOPPER DREDGER

Trailing suction hopper dredgers can be compared to large sailing vacuum cleaners that Hoover (dredge) up sediment from the bottom of maritime environments. They fall within the category of hydraulic dredgers which make use of centrifugal pumps in the dredging process. The sediment which is dredged by TSHD are generally sand, gravel and clays of various specifications.

Considered the workhorse of the dredging industry the TSHDs are used throughout various maritime projects. These projects can both be maintenance as construction based. Maintenance work relates to the deepening and upkeep of waterways and harbours. Construction on the other hand is the reclamation and creation of new land, such as the well known reclamation works commissioned in the east and the Rotterdam harbour extension de tweede Maasvlakte. Figure 7.7 displays a trailing suction hopper dredger.

Using a TSHD has various benefits and advantages over other equipment. They are robust vessels that are able to work under extreme offshore conditions. Additionally the ship does not dredge in a fixed position and they are self propelled, often fitted with dynamic positioning systems. This is opposed to fixed dredging vessels that require anchors and cables. A result is that this lowers interference with other maritime traffic while working in busy water ways such as harbours.

The main components of the TSHD are the suction pipe(s), hopper, drag head and the vessel's hull. As can be seen on the image the suction pipes are lowered next to the vessel into the water. The end of the pipes are connected to the drag head which job it is to loosen the sediment and dredge it into the suction pipe. The suction pipe then further transports the sediment water mixture onboard where it will be deposited into the hopper. Once the hopper is fully loaded it sails to its destination and deposits the sediment as required.

HOPPER

The hopper is the part of the vessel that is designated to hold the dredged material while en-route to the deposit location. This component easily takes up the most space on board the vessel and will directly influence its size. Figure 7.8 depicts a cross section of a hopper along its length.

As can be seen in figure 7.8 the hopper is filled by the suction pumps. The mixture entering the hopper then goes through a process called sedimentation, where the solid particles fall towards the bottom of the hopper. It is important to realise this is a very simplistic realisation however, in reality this is a highly complex domain into which much research is being done.

The hopper contains an overflow mechanism which enables the vessel to get rid of excess water. This overflow comes in two flavours, fixed and dynamic, which is related to the carrying capacity of the vessel and its design cargo.



Figure 7.7: Trailing Suction Hopper Dredger (Source: <http://www.theartofdredging.com>)

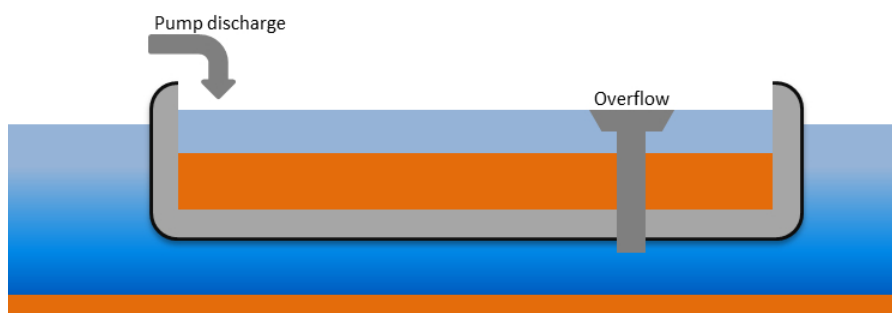


Figure 7.8: Hopper with overflow.

CARRYING CAPACITY

The carrying capacity for a TSHD depends on the loading mechanism used, for which there are essentially two: The *Constant Tonnage System* (CTS) and the *Constant Volume System* (CVS). Both of these are displayed in figure 7.9 and will now be further elaborated:

- **CVS:** This method makes use of a fixed overflow level, which means that the effective volume of the hopper will not change. Such a system is designed for filling hoppers with low density soils. This system does not have a constant hopper tonnage throughout the loading phase.
- **CTS:** This method makes use of an adjustable overflow level, which means that the effective volume of the hopper can change. Such a system is designed with a maximum tonnage in mind. When the hopper contents reaches maximum tonnage the overflow level is lowered such that the tonnage of the hopper remains the same throughout loading. This system is designed for filling hoppers with high density soils.

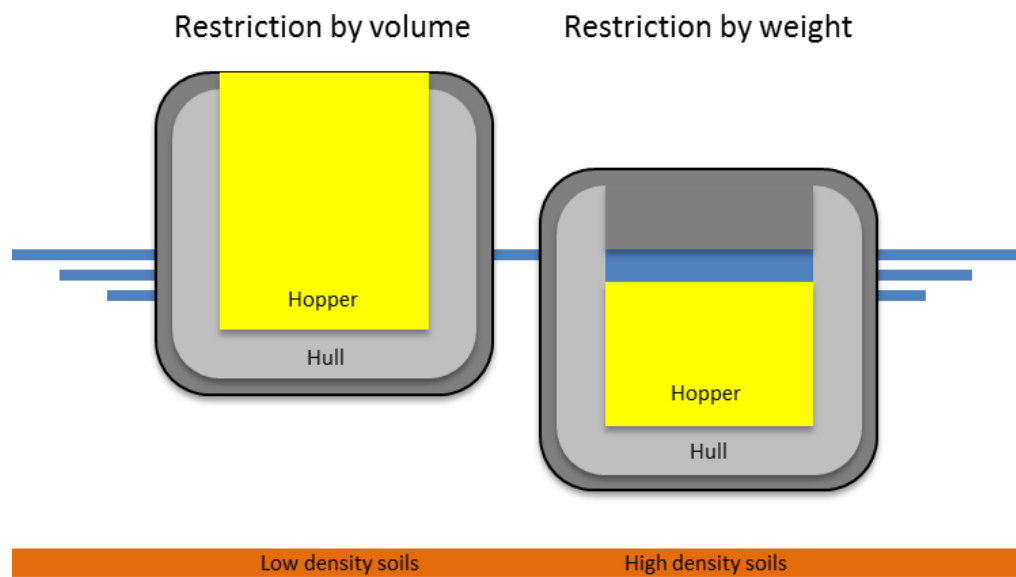


Figure 7.9: Carrying capacity.

LOADING CYCLE

Earlier it was mentioned that the TSHD has various phases throughout the loading process which makes up the loading cycle of a trailing suction hopper dredger. The loading cycle consists of seven phases, each of which will shortly be described below and are also represented in figure 7.9. Awareness of these cycles is important when deciding on how to design the source model and what should or should not be included.

- **Phase 1: Achieving minimum draught** The first phase of the loading cycle starts directly after loading has completed. During this phase the overflow is lowered so the remaining water is able to flow out of the hopper and sailing towards the dump area is started.
- **Phase 2 - Sailing to the discharge area** The second phase is where the ship sails to its destination. By dividing the distance (minus the distance sailed in phase 1) with the sailing speed, the sailing time is gained.
- **Phase 3 - Depositing of the cargo** The third phase of the cycle is the unloading of the vessel by depositing the cargo on the target site.
- **Phase 4 - Sailing to dredging area** Once the cargo has been discharged the empty vessel sails towards the dredging area. Sometimes the hopper will be filled with water to achieve a more desirable sailing efficiency, or depth when starting dredging.

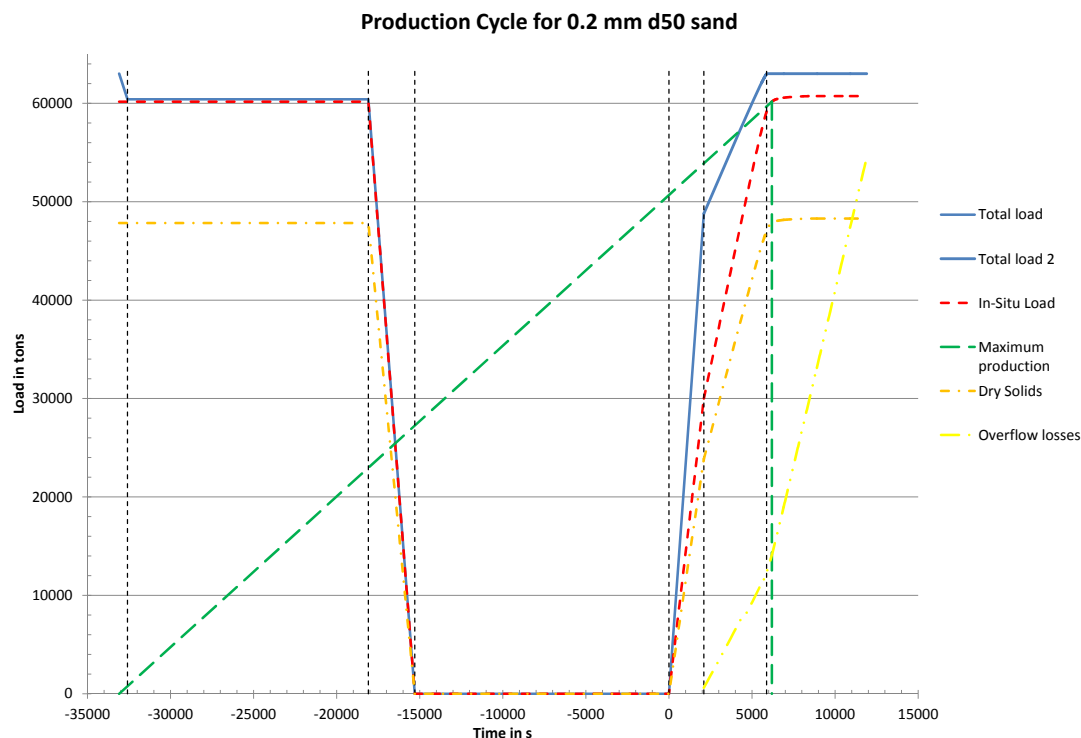


Figure 7.10: Production Cycle for a hopper dredge.

- **Phase 5 - Dredging till overflow** This phase is the start of the actual loading of the vessel. The hopper is continuously filled with slurry until the overflow level is reached. At the end of this phase, assuming that the sand has settled, the volume and the height of the sand bed can be determined.
- **Phase 6 - Constant Volume Dredging** Phase 6 and phase 7 are the most difficult phases to model. This is due to the increasing importance of scouring³, turbulence and overflow losses.
- **Phase 7 - Constant Tonnage Dredging** The last phase of the loading process, the Constant Tonnage Dredging, is characterized by a quickly decreasing settling efficiency. During this phase, no additional weight is added to the hopper. Therefore, the volume flow into the hopper does not equal the volume flow out ($Q_{in} \neq Q_{out}$), but the mass flow in does equal the mass flow out ($M_{in} = M_{out}$). This implies that while the sediment is rising the water level is decreasing, which then results in an even more rapidly increasing flow velocity of the sediment bed.

The values at the end of phase 7 are the input variables for phase 1, thus creating a continuous process.

7.2.2. THE MODEL

The TSHD model will be based on the results of a student project to design an trailing suction hopper dredger [105]. The simulation model can be created using three approaches: 1) First principles⁴, 2) Empirical input and output obtained from the field, 3) A combination of both. This quickly narrows down to first principles as obtaining such empirical data is infeasible within this works scope. The approach used is to start out with the most simplistic first principles model with the option to increase complexity after initial evaluation. Reasoning for this is that the goal is to evaluate the IDS system and not to create the most complex and realistic TSHD model.

A TSHD has either a fixed or dynamic overflow system. Within the student report use was made of a dynamic overflow system, which will also be used in the model as this offers a possible vector for malicious behaviour. The main risk within the loading cycle is that this overflow does not work as intended. This has the

³Erosion of the sediment due to increasing water flows.

⁴In physics the first principles approach relates to something which is based directly on established science.

potential to cause the system to overload and sink the vessel. There are other parts that could malfunction however, such as a suction pump not turning off. In those cases however lowering the overflow would win time and safety by simply ensuring excess cargo is discharged overboard. Exploiting this risk the other phases within the cycle are not necessary for prototype evaluation, which aims to detect anomalous behaviour of any system and should be independent of loading phases. Modelling every phase is thus not be worth the time costs for the benefit it provides.

The source model is built in such a way that there is a physical model representing the TSHD, and a separate controller that influences the state of the modelled TSHD. This mimics the functionality of a real controller, which also operators on a process. The physical model include the following main components: The hopper, the dynamic overflow and the inflow pipeline. In reality there would be many other parts involved but for the purposes of the evaluation these are not required and will be presented abstractly within the model.

The control network receives the sensor information from the physical model and processes this. After processing the controller computes the required change and send a control message that influences the state of the source system. This has been represented with figure 7.11, displaying the field devices in play and their connection to the controller.

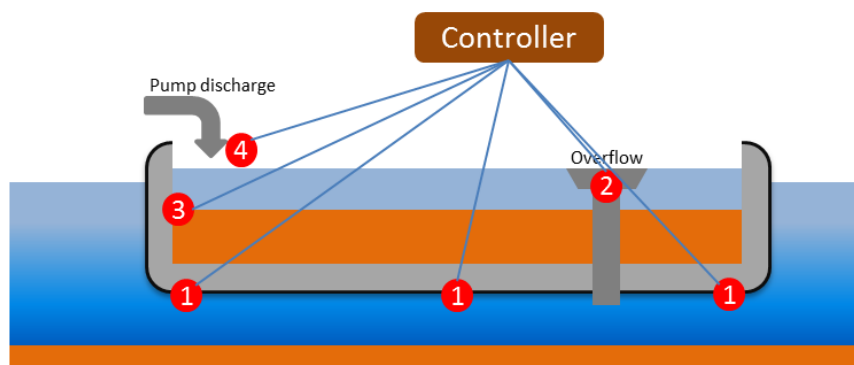


Figure 7.11: The TSHD modelled network.

MODEL DIMENSIONS

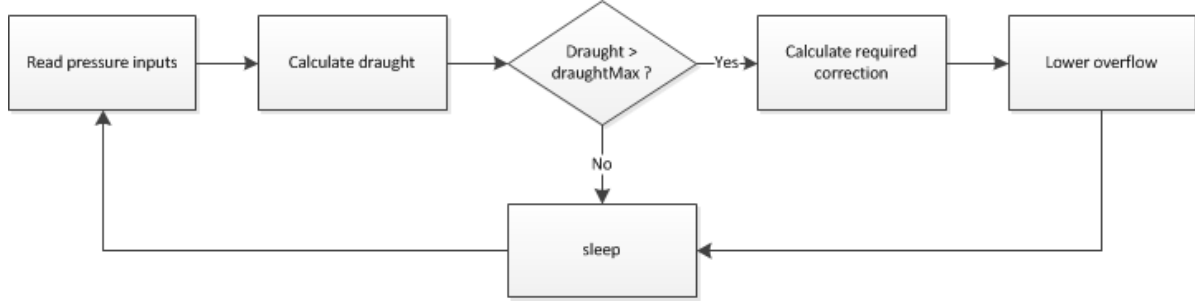
The dimensions and parameters used for the model are based on the student report for the subject *OE5671* and are listed in the tables below:

Environment			Vessel		
Parameter	Value	Unit	Parameter	Value	Unit
seawater density	1025	[kg/m ³]	Length ⁵	270	[m]
particle density	2650	[kg/m ³]	Width	31	[m]
water density	1025	[kg/m ³]	Depth	15.5	[m]
d50	0.2	[mm]	Draught	13	[m]
d15	0.1	[mm]	Block coefficient	1.0	[-]

Hopper		
Parameter	Value	Unit
Length	115.2	[m]
Width	20.6	[m]
Depth	14.7	[m]
Draught	13	[m]
Hopper density	2000	[kg/m ³]
Mixture density	1200	[kg/m ³]
Settling efficiency	1	[-]

CONTROLLER LOGIC

The goal of the controller is to keep the draught of the TSHD stable once the maximum draught has been reached during the loading phase. This (near) continuous process is done in two steps: 1) Determine current draught, 2) Determine new overflow position. The flowchart below represents this process in more details:



Draught calculation The draught of the system is not measured directly and is thus not known to the controller. It can however be estimated by using the pressure sensors along the keel of the vessel. The pressures can be used to calculate the current draught and its pitch⁶. Using 3 pressure sensors the average draught is than calculated as:

$$\bar{D} = \frac{\sum_1^3 D_i}{n} = \frac{\sum_1^3 p_i}{3 \cdot \rho_{water} \cdot g}$$

Where,

D_i	Draught at sensor i	[m]
n	Number of draught sensors	[-]
p_i	Pressure at sensor i	$[\frac{N}{m^2}]$
ρ_{water}	Density of water	$[\frac{kg}{m^3}]$
g	Gravitational acceleration	$[\frac{m}{s^2}]$

Overflow adjustment calculation The adjustment required for the overflow can be determined when the current draught is known. If the current draught is lower then the maximum draught the overflow will remain in place. Otherwise the following will determine the required change. First the required change in hopper water level is determined by:

$$y = (D_{max} - \bar{D}) \cdot \frac{A_{wl}}{A_h}$$

This required change in water level can be converted to the required change in overflow height by:

$$\Delta H_o = (H_o - H_{wl}) + y$$

Where,

D_{max}	Max draught	[m]
\bar{D}	Average draught	[-]
A_{wl}	Waterline area	$[m^2]$
A_h	Hopper area	$[m^2]$
H_o	Height overflow	[m]
H_{wl}	Height waterline cargo	[m]

7.3. EXPERIMENTS

The evaluation strategy is defined and a source model to monitor has been created. The next step in the evaluation process are the initial experiments. There are a total of three experiments which aim to evaluate the prototype and the proposed detection strategies. Each is based on vulnerabilities identified by the threat model and inherent weaknesses in the source model. These experiments are: Cyber incident, Manipulation attack and Envelope escalation, each targeting a specific detection strategy. The specifics will be further discussed in the respective section, where the results will also be presented. The discussion of these results can be found in the next section 7.4.

⁶The angle along the length of the vessel

7.3.1. EXPERIMENT I: CYBER INCIDENTS

A cyber incident occurs when a unwanted situation occurs but there is no malicious intent trying to cause the situation. An example is the overflowing of a tank because a control engineer has entered the wrong maximum volume for said tank. While these events might resemble a cyber attack, the differentiator is intent. This experiment will evaluate the *value analysis strategy* and determine to what extent it is functioning as expected; Namely that a warning is presented when the physical process deviates from system specifications and possibly transitions into a physically impossible or critical state. A graphical representation of this approach is depicted in figure 7.12. The evaluation is accomplished by introducing two incidents that each target a different part of the system. During each iteration one of these is randomly selected. The two incidents used are then: The first case mimics the changing of a system set-point. The second case alters process logic used within the controller. Both of these incidents mimic an operator, engineering or operational error which can cause the physical process to move outside of its (safe) operating specification.



Figure 7.12: Value analysis

- First case: Set-point tampering** During a recent renovation to the TSHD some changes were made to the vessel. As a result the information within the control system has also been updated to reflect this change. At this point it is well possible for the control engineer responsible for these changes to (accidentally) enter the calibrate the wrong value or change the wrong set-point. This could be due to a calculation error, unclear instructions or even a simple typing error. The TSHD is a constant tonnage dredger and as such designed to start displacing water out of its hopper once she reaches maximum draught -and thus maximum tonnage. If the maximum draught is changed to a lower value the contents of the hopper will be lower than expected, which can cause delays to the project. If this draught is changed to a higher value however the contents of the hopper will be allowed to surpass the maximum capacity of the vessel. This can have serious problems on vessel hydrodynamics and can lead to grounding and sinking of the vessel. In this incident the maximum draught set-point within the controller is set to 125% of the actual value. The expected consequence of this action is that the vessel will keep on loading past its maximum capacity and cause the ship to sink due to overloading.
- Second case: Malicious controller** After having gone through dry-dock maintenance the dredger is on its way to the next project. Unbeknownst to the crew there is a problem with the control system operating the overflow, causing it to stop working correctly. This could have happened due to a software bug transferred to the controller after updating the firmware. Due to the malfunctioning controller the overflow does not respond to any input and thus does not lower when it is expected to. Because there is no connection to the draught and loading condition the vessel will keep loading itself, even past its designed capacity. Especially at night, when the contents of the hopper and draught of the vessel are hard to see, the consequences for this can be disastrous. In this incident experiment the command which lowers the overflow has been disabled. The expected consequence of this change is that the vessel will keep on loading past its maximum capacity and eventually cause the ship to ground or sink due to overloading.

Experiment parameters To gain insights into the general effectiveness of the prototype and value analysis detection strategy the experiment will be conducted multiple times. Each iteration will have a different seed value which cause the modelled system to behave slightly different and randomly selects which of the two previously discussed incidents is selected. Every seed is repeated for the range of sample rates between [1, 60] seconds, because it is expected that a slower sample rate will have a negative effect on the detection rate.

Before processing the experimental results it is important to ensure that enough iterations have been executed and the results are stable. This insures that the detection strategy has been evaluated in a meaningful manner, as was previously discussed in 7.1.4. Stability can be determined by plotting the cumulative mean delta - using eq. 7.3 - for the true positive percentage after each iteration and comparing the result to a threshold. Figure 7.13 represents this graph for each of the sixteen sample rates tested in the experiment. After ten iterations the delta drops below the 0.5%, as clearly indicated by the graph, and with increasing iterations moves towards zero. With a threshold set at 5% this graph proves that sufficient iterations are available for further analysis.

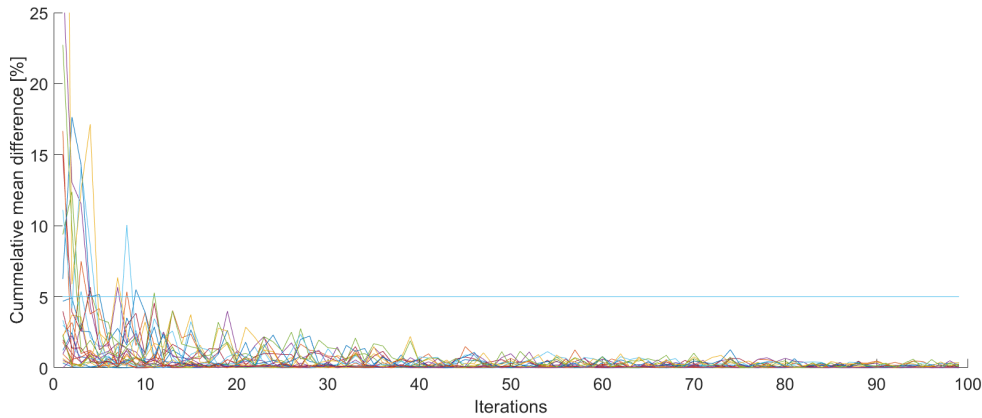


Figure 7.13: Stabilisation of cumulative mean

There is one important note however, which is that verification of a stable delta will not be sufficient by itself. It is important to verify that the data from which the moving average is determined does not demonstrate unexpected patterns for which the delta just happens to stabilize. An example of this has previously been given on page 81. In this case the obtained experimental data does not exhibit any such patterns.

BINARY CLASSIFICATION

The raw results obtained from the experiments are lists of true/false values, one list for each seed and sample rate with one value for each sample. This output format provides little insight into the detection strategy and prototype, let alone the influence that sample rate has on the whole. This problem is solved by using the binary classification method specified in section 7.1, which converts this output into the more understandable metrics "true positive rate" and "false alarm rate". These metrics are discussed next.

True positive rate After converting the raw results into a true positive rating for each iteration, the mean value over all seeds is determined for every sample-rate. The resulting rates are then plotted against their corresponding sample rate to provide a figure which should indicate the relation between sample-rate and TPR. As it turns out, the sample rate has hardly any influence on the TPR, which is surprising as a certain correlation was expected. It was believed that a decrease in sample rate would result in a decrease in detection rate as more malicious samples would be missed. This discrepancy will be further discussed in the evaluation of the initial experiments in section 7.4 on page 95. This is not the only insight to be gathered from the obtained results however. They indicate that when the prototype picks up a malicious sample the "value analysis" strategy in its current implementation and setting has a 88.7% change to correctly identify the sample as such.

False alarm rate Using the same process as the one used for the true positive rate, the false alarm rate is determined. Following the TPR results the sample rate appears to have hardly any influence over the FAR. This behaviour will be further investigated and discussed in section 7.4. What the results also return is that

for every non-malicious sample fed to the prototype and investigated by the "value analysis" strategy there is a 1.31% chance that it is wrongly identified as malicious and sends out a false alarm.

7.3.2. EXPERIMENT II: MANIPULATION ATTACK

The second experiment aims to evaluate the prototype while making use of the "value analysis" detection strategy. This is done by determining to what extent it is living up to its expectations to raise an alarm once the physical state deviates from the state reported by the control system. Figure 7.14 depicts a graphical representation of this detection method. The experiment evaluates this method by forcing the controller to report a false state to the control system for one of three field devices. This produces three possible incidents, one of which is chosen at random during each iteration:

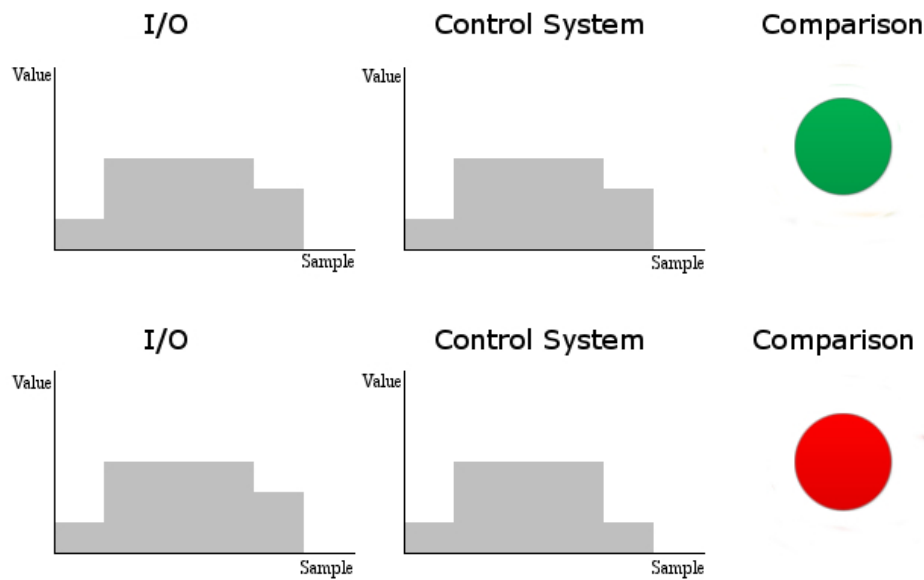


Figure 7.14: Value analysis

- **First case: Static cargo** At a randomly determined time during the loading phase the reported hopper contents to the control systems goes faulty and starts reporting a constant value, even though the hopper is still being loaded and the cargo is thus increasing. This constant value is maintained through the whole loading phase or until the vessel sinks.
- **Second case: Static overflow** At a randomly determined time during the loading phase of the hopper the reported overflow level goes faulty and starts reporting a constant value, even though the overflow should be changing depending on the draught of the vessel. This constant value is maintained throughout the loading phase or until the vessel sinks.
- **Third case: Misleading draught** The modelled hopper is a constant tonnage system, meaning that at a specific draught the ship will not sink any further even though sediment is still being dredged. Instead the overflow is lowered to keep the tonnage constant. This incident causes the reported draught to never reach the specified maximum level by changing its actual value. This keeps the vessel from lowering the overflow which in return can cause the overloading of the vessel if no timely action is undertaken. This incident will not occur at a random moment, but at 90% maximum draught. At that moment the value is kept constant by the controller effectively tricking the larger system into continual loading.

Experiment parameters Following the previous experiment, the second experiment will conduct multiple runs with varying seed values to gain insights into the effectiveness of the prototype and this detection strategy. There is one difference however, which is that this detection strategy has a deviation parameter because the sample value is allowed to deviate slightly from that reported by the control system. There could

be multiple reasons for this, such as difference in recording time or simply interference along the path when for analog modulated signals.

As for the previous experiment, it is important to have a sufficient number of simulations to enable a meaningful discussion about the results. The cumulative mean delta is plotted in figure 7.15 for a range of sample rates. It can be seen that after fifty iterations the difference has dropped below the 5% threshold, indicating sufficient results have been obtained. The second verification is done by looking at the input used to create the cumulative mean, represented by figure 7.16, which shows no convergent or divergent behaviour as specified in section 7.1 page 81. Thus it can be confirmed that sufficient iterations have been run.

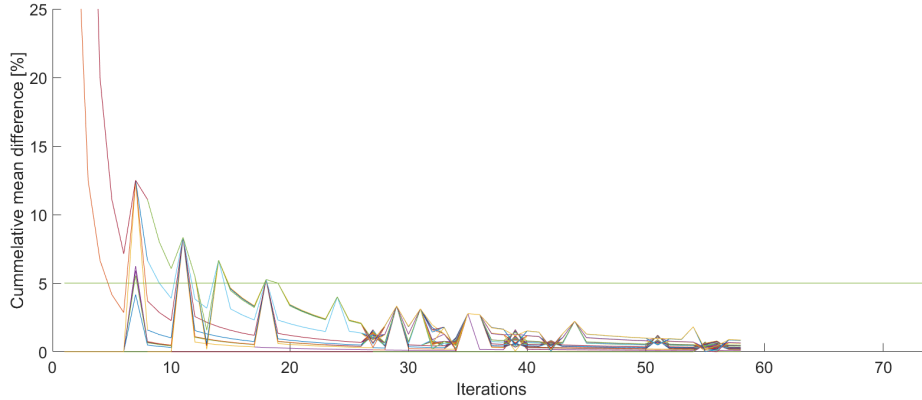


Figure 7.15: Stabilisation of cumulative mean

BINARY CLASSIFICATION

With sufficient information for a stable evaluation the next step is to determine the true positive rate and false alarm rate, done through the process of binary classification. This follows the first experiment and has previously been discussed in section 7.1 on page 7.1.3.

True positive rate Running the experimental output through the binary classification process returns the true positive rates for each seed, sample rate and deviation. Taking the mean values over all the seeds gives a list of true positive rates related to the sample rate and deviation. These mean results are plotted against one another, the result of which is displayed in figure 7.16. This graph indicates that the sample rate has hardly any influence on the sample rate, which mimics the same behaviour displayed by the first experiment. This behaviour will be further discussed in 7.4. However, there is an interesting observation that deviates from the first experiment which is that there appears a fluctuation in the lower sample rates. This can be attributed to the change in sampled malicious versus non malicious ratio due to the change in sample rate. When focussing on the deviation, this clearly has an influence on the detection rate. When the deviation allowed between sample value and ICS value increases, the true positive rate decreases. This means that the detection method increasingly fails to correctly detect malicious samples, which is expected behaviour. When the deviation increases, so does the allowed range of values before any sample will be deemed malicious. The detection rate for a deviation rate of 1% or lower has a 100% detection rate, indicating that all malicious samples are correctly identified, which is expected behaviour. A low deviation rate will quickly classify any deviating sample as malicious. This might however negatively influence the FAR. Further discussion of the results is done in 7.4 on page 95.

False alarm rate The false alarm rate is computed from the results much like the true positive rate, of which the mean is taken for every seed. The obtained results are then plotted against the sample rate and deviation, which is represented by figure 7.17. Following suit to the true positive rate, the false alarm rate is also independent from the sample rate. Interestingly, the minor fluctuation that is found in the true positive rate does not carry over to the false alarm rate. The reason for this is likely due to the low number of false alarms reported. The behaviour shown for deviation dependence is different from the true positive rate, there is no visible correlation between the deviation and false alarm rate. This is actually expected behaviour which

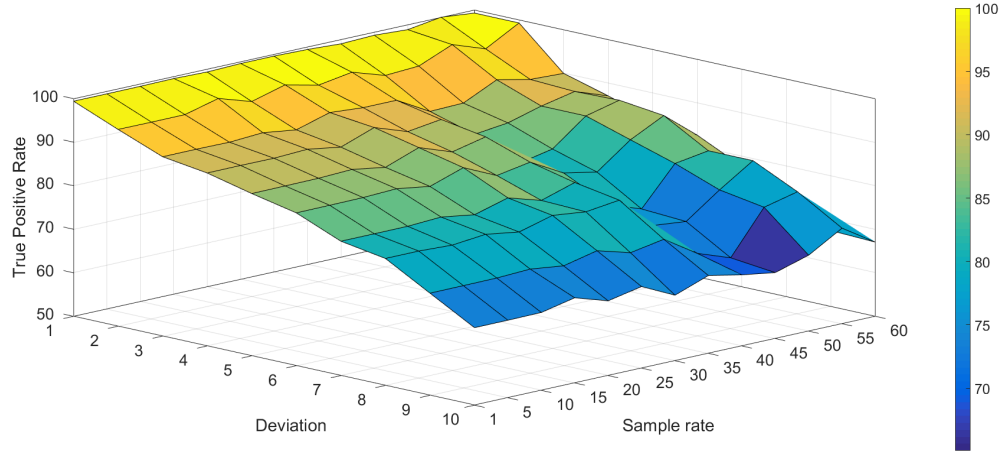


Figure 7.16: True positive rate

is due to the design of the experiment where the values reported by the field devices do not fluctuate sufficiently to cause significant false alarms. Furthermore, when a sample is non malicious there is no logic that randomly "changes" some output given by the ICS that would cause a false negative alarm. The highest FAR is $9.5 \cdot 10^{-5}\%$. The reason for this not being 0% is attributed to the uncertainty on the reported value which is present in the real field devices due to imperfections and, wear and tear.

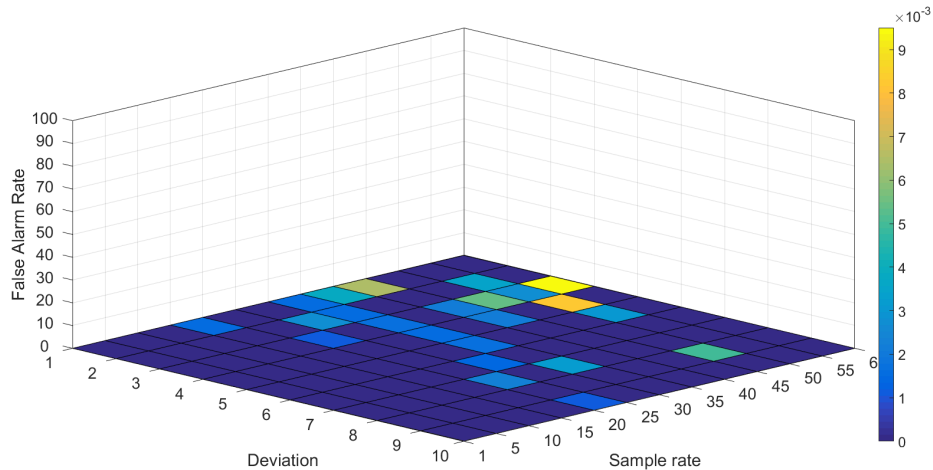


Figure 7.17: False alarm rate

7.3.3. EXPERIMENT III: ENVELOPE ESCALATION

The main aim for this experiment is to evaluate the functionality of the "envelope escalation" detection method used by the prototype. This method should raise an alarm if the physical process deviates from an established safety envelope within which the system is inherently secure. A graphical explanation is given in figure 7.18. This evaluation is accomplished using three possible incidents that will each move the system outside of such an envelope. One such incident, further discussed next, will be randomly selected for each iteration cycle.

- **First case: Static overflow** At a randomly determined time during the loading phase of the hopper the reported overflow level goes faulty and starts reporting a constant value, even though the overflow

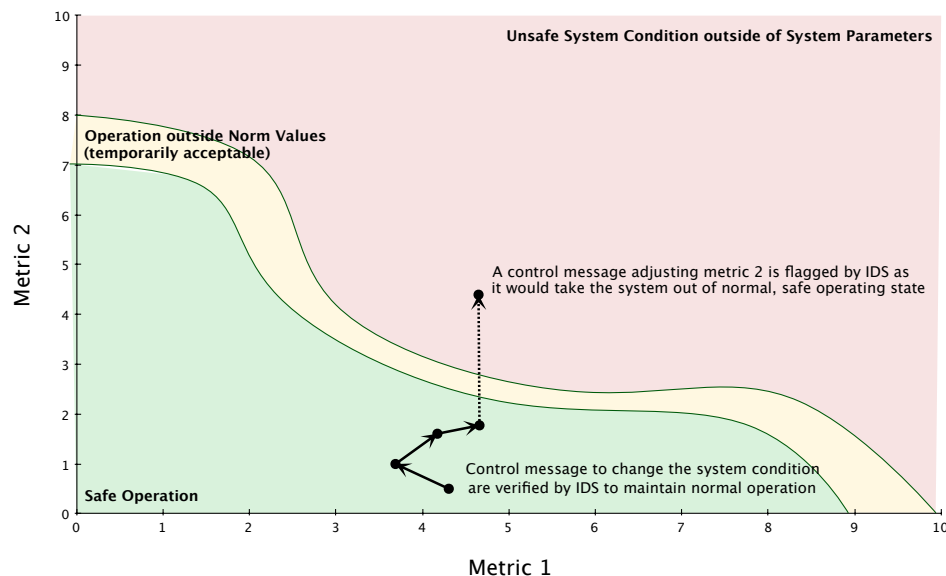


Figure 7.18: Envelope analysis (Source: Dr. ir. C. Doerr)

should be changing depending on the draught of the vessel. This constant value is maintained throughout the loading phase or until appropriate action has been undertaken.

- Second case: Set-point tampering** During a recent renovation to the TSHD a change was made to the changed the maximum draught of the vessel. As a result the information within the control system will also have to be updated to reflect this change. At this point it is well possible for the control engineer responsible for this chance to (accidentally) enter the wrong calibration or set-point. This could be due to a calculation error, unclear instructions or even a simple typing error. The TSHD is a constant tonnage dredger and as such designed to start displacing water out of its hopper once she reaches maximum draught -and thus maximum tonnage. If the maximum draught is changed to a lower value the contents of the hopper will be lower than expected, which can cause delays to the project. If this draught is changed to a higher value however the contents of the hopper will be allowed to surpass the maximum capacity of the vessel. This can have serious problems on vessel hydrodynamics and can lead to grounding and sinking of the vessel. In this incident the maximum draught set-point within the controller is set to 125% of the actual value. The expected consequence of this action is that the vessel will keep on loading past its maximum capacity and cause the ship to sink due to overloading.
- Third case: Slowed slurry flow** When slurry⁷ is pumped through a pipeline at a sufficient velocity all particles can be assumed to be moving and homogeneously spread over the pipe cross-section. When this flow speed is reduced however there will be a point where the particles start to fall towards the pipe bottom and form a bed. The speed at which this occurs is called the critical speed. This bed can lead to the forming of a plug inside the pipeline with serious consequences as a result. This attack reduces the velocity in the pipeline to below the critical speed limit, which could be a simple mistake by the operators. On the other hand this could also be due to sensor errors or targeted malicious activity.

Experiment parameters Following the first experiments, there will be multiple runs with varying seed values to gain insights into the effectiveness of the prototype and this detection strategy. For every experiment the IDS will be tested for a range of sample rates as it is expected that a decreasing sample rate will have a negative effect on the detection rate.

Like the previous experiments the first step is to determine that sufficient iterations have been run. Representing the percentage difference between cumulative means at i and $i - 1$ for the range of sample rates is figure 7.19. This graph shows that after fifty iterations the difference has dropped below the 0.5% threshold, indicating sufficient iterations have been run. Furthermore, the strategy has previously discussed that the cumulative mean alone is not enough because the raw data itself could still be fluctuating. Verification of the

⁷A mixture of sediment and water.

raw data does not show any sign of such behaviour, thus confirming that sufficient iterations are available for further evaluation.

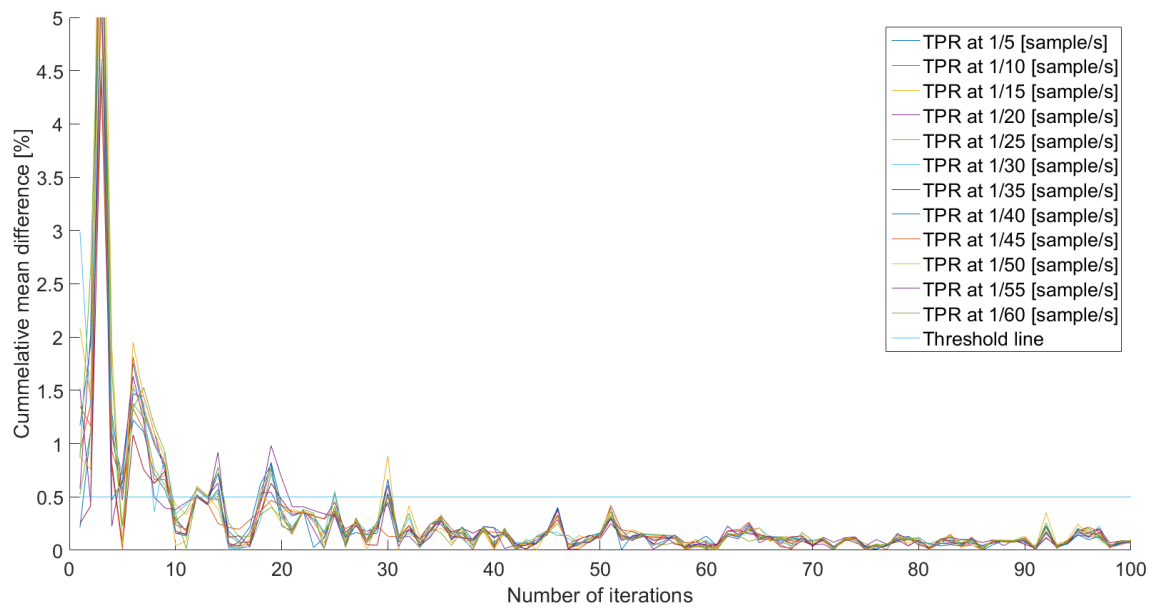


Figure 7.19: Stabilisation of cumulative mean

BINARY CLASSIFICATION

Following the verification of sufficient available data is the determination of the detection rates, which is done using binary classification and will return specifically the true positive rate and the false alarm rate. The process behind this classification has been explained previously in section 7.1 on page 75. Next the TPR and FAR will be discussed.

True positive rate After obtaining the raw results they were turned into a true positive rate and plotted against their corresponding sample rate. Much like the previous experiments it was expected that with an slower sample rate the detection rate would drop. This however, as previously, is not the case and the rate appears unaffected by a changing sample rate. This will be further investigated in section 7.4. Independence from sample rate is not the only thing learned from these results however. They indicated that when the prototype picks up a malicious sample, the detection strategy has an 92.3% chance to be correctly identify it as being malicious.

False alarm rate Processing the results using binary classification and plotting the outcome against the related sample rate gives figure 7.20. The sample rate seems to have a minor negative effect on the false alarm rate, which after the previous experiments is unexpected. The reason for this is most likely to be found in the lower number of total samples used by a slower sample rate. This will be further discussed in section 7.4. Looking at the graph it is clear that there is a 10.7% chance that a non malicious sample will be marked as malicious by the envelope escalation strategy.

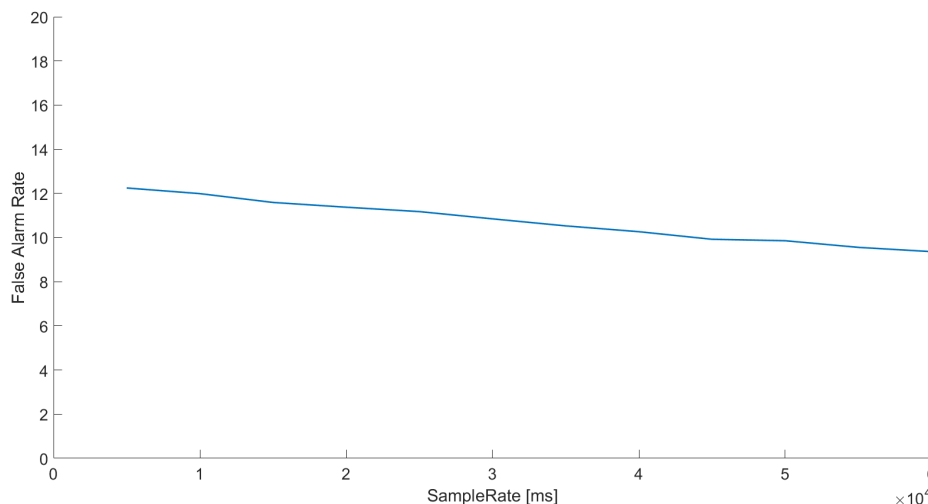


Figure 7.20: False alarm rate

7.4. EVALUATION AND IMPROVEMENTS

The previous section discussed three experiments aimed at evaluating the prototype along with the proposed detections strategies, and presented the results found. This section builds on this by discussing these results in more depth and proposes several improvements that potentially increases performance.

7.4.1. VALUE ANALYSIS

The first experiment investigated the detection rates for the value analysis, which resulted in the following rates:

True positive rate	False alarm rate
88.7%	1.31%

At first glance these values look satisfactory, out of every 100 malicious samples 88 will be detected whereas for every 100 non malicious samples only 1 will raise the alarm. However, consider the following: A sensor of which a sample is taken every 10 seconds, meaning 86400 samples on a daily basis. This computes to a total of 1131 alarms on a normal day, when no incidents are occurring! This is obviously not a workable situation. As for the TPR, if the sample rate is not sufficiently high there is the risk that an attack won't be discovered until the process has reached the critical state already. It will be up to operators to decide what threshold is seen as acceptable.

It is thus important to investigate why these rates are not better such that improvements can be made, or those configuring the system correctly informed. The problem here is simply that there are too many false positives and false alarms. Figure 7.21 indicates where these are originating from. The field device has a specific noise, it's reported value will fluctuate. This thus means that at times the value will be reported to be higher or lower than the actual signal, which then triggers the threshold as this behaviour has not been taken into consideration.

One way to lower either the false positives, or the true positives, is to add a certain deviation to this threshold; This is represented by figure 7.22. The problem here however is that by adding a deviation in either direction the other will be negatively influenced, they are inversely connected. The best configuration thus depends on the system being monitored and the preference of the operators: Faster detection and more false positives, or less false positives and slower detection.

Adding a deviation to the threshold is not the only way to improve this detection strategy however. When configuring the monitoring system the information and assumptions with regard to the malicious threshold will also be important and influence results. Say that a pipeline is designed to withstand at maximum 100 bar, but throughout normal operations should not surpass 60 bar. If the threshold is now placed at $60 + \max(\text{sensor noise})$ there should not be any false positive values in normal operations. When this line is crossed however the state can be considered to be an anomaly and an alarm should be raised, all before the critical state is actually reached. This is displayed by figure 7.23.

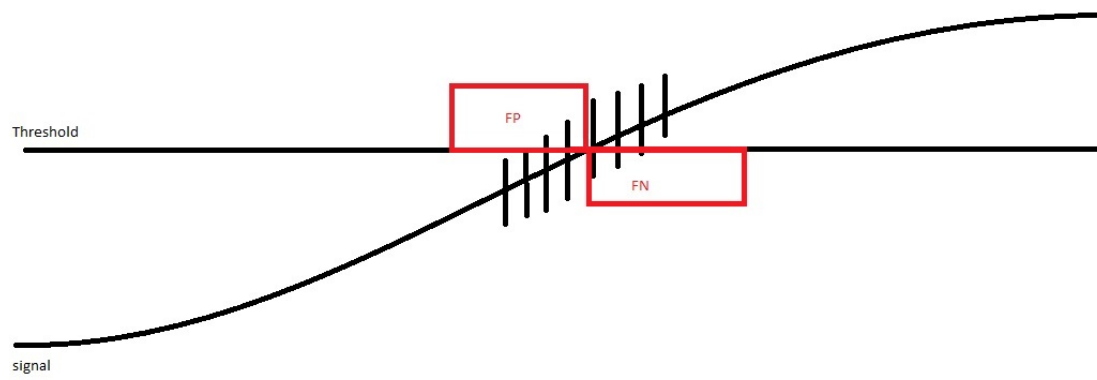


Figure 7.21: Value analysis threshold

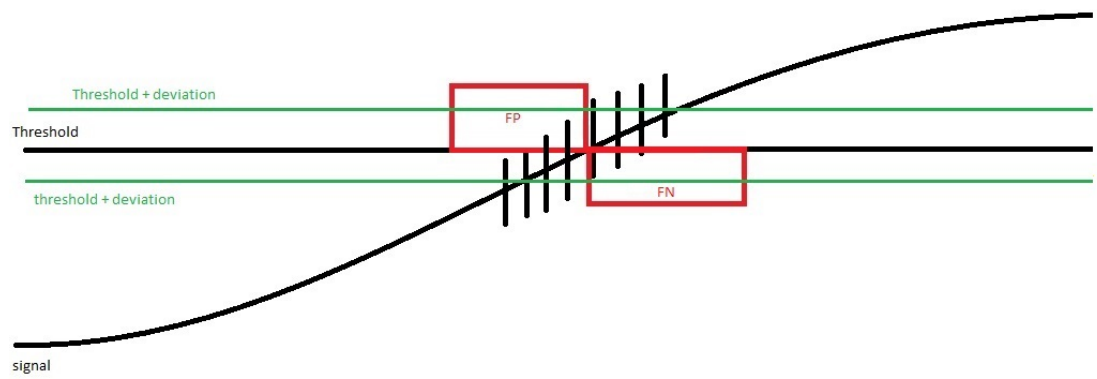


Figure 7.22: Value analysis threshold

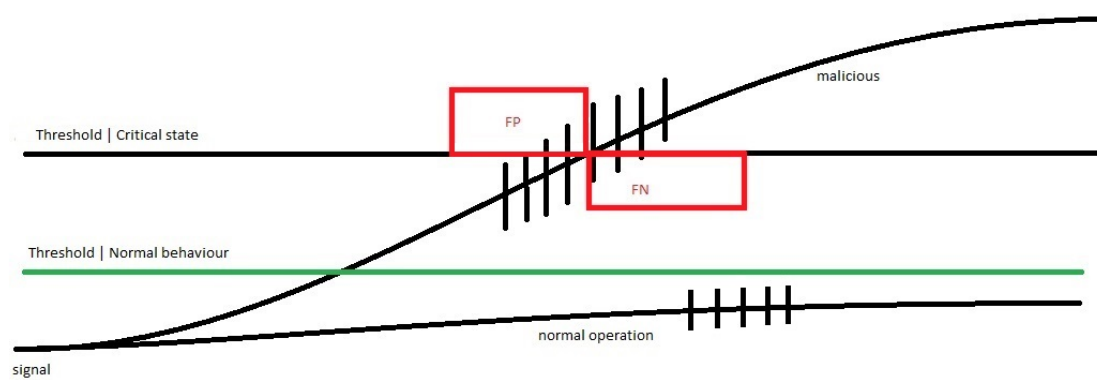


Figure 7.23: Value analysis threshold

Interestingly if the critical state is defined as malicious, obviously an experiment would report false positives until the critical state is reached. As such this can cause the results to be skewed based on semantics. This offers room for future research however, especially when taking the semantics into consideration, and should not be ignored when such a system is actually deployed.

7.4.2. CONSISTENCY ANALYSIS

The Manipulation attack investigated the detection rates for the consistency analysis and presented the following results:

True positive rate	False alarm rate
100%	$6.3 \cdot 10^{-6}\%$

Which would correspond to a deviation of 1% and represent the most favourable outcome. The following table however more accurately describes the range:

	True positive rate	False alarm rate
Min:	65.0%	$6.3 \cdot 10^{-4}\%$
Mean:	84.4%	$6.2 \cdot 10^{-4}\%$
Max:	100%	$9.5 \cdot 10^{-3}\%$

Where these results vary with the deviation that is an acceptable difference between sample value and ICS value. Although the 100% positive rate raises questions at first, it is consistent with what is expected. Having a low variation means that the sample value should be nearly equal to what is reported by the ICS, even small deviation from this will raise the alarm. Having a larger variation will inversely mean that there is more room for the ICS to report false values.

There is a possible downside with a low deviation however, which has to do with valid reasons for deviation between the two values. One such example could be interference on the wire. Nonetheless the results do not reflect this downside, which is due to the fact that the model did not incorporate potential sources of non malicious deviations. As such any non malicious value will be completely equal for both samples and no alarms will be raised. Even when the deviation would be set to 0%.

Figure 7.17 however does display some variation in the false alarm rate, which makes it seem that some non malicious values are in fact being reported as malicious. This however is likely to be an artefact due to the limits of the processing system. It would be good however that future research verifies this claim prior to using this detection scheme.

7.4.3. ENVELOPE ANALYSIS

The Manipulation attack investigated the detection rates for the consistency analysis and presented the following results:

True positive rate	False alarm rate
92.3%	10.7%

Looking at these results the current implementation of the envelope detection strategy is clearly not sufficient. For every 100 samples the envelope will report 10 alarms on average, which will overload the operators very quickly. Digging into the analysis specifics it turns out that one of the used boundaries for the envelope is not as accurate as initially thought due to inherent complexities. The boundary contains statistics to counter the noise presented by the field devices to increase the true positive rate, this however increases the chance that false positives arise. As such a lower FAR was expected to some extent. Due to the current experimental setup however it is difficult to determine exactly where the pain point lies, and thus to verify the above assumption.

These results do raise important lessons. First the creation of boundaries should be kept as simple as possible, because increased complexity makes it harder to determine what specifically is happening and might confuse operators. This will also improve the validation steps prior to site deployment. Second is that (complex) boundaries require testing prior to their commissioning. Running unit tests on each boundary will provide early feedback and make full evaluation of the envelope easier as their operational limits are well known. Such tests could include the option to plot the results versus the boundaries, visualising the boundary evaluation and make it possible to determine where the configuration has negative influence on the results.

7.4.4. SAMPLE RATE

From the initial experiments it is clear that the sample rate has no effect on the success of the prototypes detection strategies. Intuitively however it must have, which can logically be derived by imagining an infinitely slow rate. Here no samples would be taken and there would be nothing to evaluate, the detection rates would thus drop to zero. Alternatively, an infinitely fast detection rate would exhaust the available resources causing the prototype to malfunction. This puts into question either the evaluation method or the effectiveness of the detection strategies on a real-time continuous system where sample rates play a role.

So-far the problem seems that the binary classification method does not take all the actual malicious events into account, some are simply not sampled. The physical model however did track these and as such the classification can be updated by determining the rates using the actual number of malicious and non malicious events from the source system. Plotting the updated results for experiment I then results in figure 7.26 for the true positive rate, which now demonstrates a dependency on the sample rate.

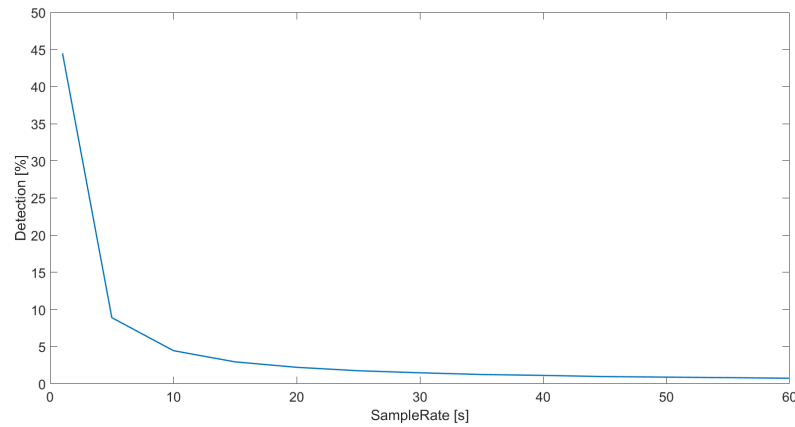


Figure 7.24: Detection rate, corrected for actual malicious events.

This adjustment appears to resolve the issue, however there is one problem. Binary classification aims to provide metrics to evaluate the detection strategies when a sample is taken: The rate with which malicious samples are identified as malicious, and the chance that any non malicious samples raises an alarm. Samples which are not analysed should not be used to evaluate the analysis strategies. At the heart of this conundrum sits a logical fallacy: It is expected that the sample rate has an influence on the detection of malicious events, performance which can be determined through binary classification. However they are not on equal terms:

- The expectation is based on a discrete system of n "samples" where some samples would be malicious in nature. From this a subset ($\frac{n * t_{step}}{t_{sample\ rate}}$) would be sampled and evaluated. Slowing down the sample rate would then increasingly miss malicious samples, which directly influences the true positive rate.
- The binary classification methods only take those samples into account that have actually been sampled and thus evaluated. A slower sample rate will also have less samples. This decouples the statistics from the sample rate, explaining the initial experiments.

As it stands it seems that the only significant impact the sample rate has is with respect to the number of samples used in the classification calculation.

Binary classification essentially sorts a vector of elements which are independent of one another. The physical system however represents a continuous process where individual instances are correlated with those "nearby". In other words: When one samples is malicious, it is highly likely the next will be malicious also. What then is mainly effected by the sample-rate is the time it takes to detect anomalous behaviour, and not binary classification. This time to detection is displayed in figure 7.25 which elaborates this graphically. Clearly, even if the first sample does not detect the malicious behaviour the next samples will. This is completely different from the traditional monitoring systems which are dealing with individual packets. It is thus demonstrated that the sample rate relates to the time it takes to detect malicious behaviour. Binary classification on the other hand will provide insight into the chance that a specific sample is correctly identified, regardless of sample time. This understanding opens up the way for improvements to the prototype.

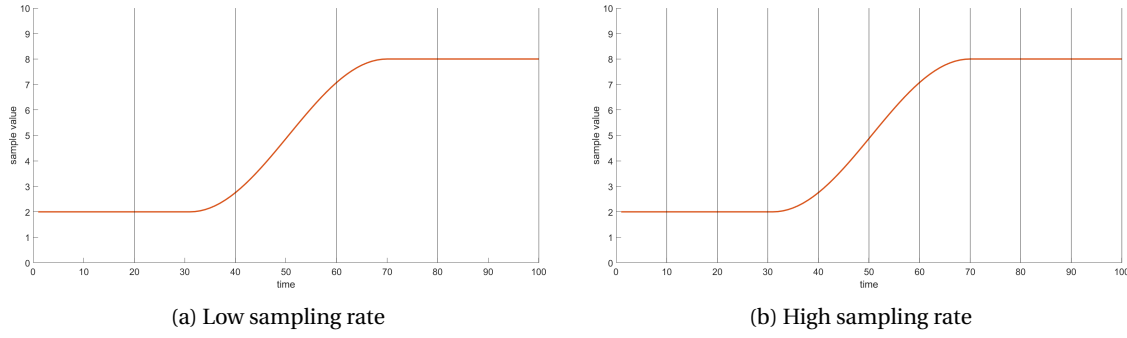


Figure 7.25: Low and high sampling rates

7.4.5. IMPROVEMENTS

The initial experiments and their subsequent evaluation have presented insights into areas where there is room for improvements that can further improve the prototype. These methods focus on improvements build on top of the current detection scheme, not changes to the detection strategies in themselves. Much of the effectiveness of the analysis strategies depend on the boundaries used, the improvements presented here are to improve results independent of such use case specific situations. The two improvements that will be discussed are:

1. Cumulative binary classification
2. Dynamic sample adjustment

CUMULATIVE BINARY CLASSIFICATION

The monitoring system works on a physical system, this means that each sample is correlated to some extent to the nearby samples. This correlation can be exploited by using the knowledge of previous sample(s) when determining the state of a current sample. A branch of statistics known as conditional probability can be used here to answer what is effectively the following question: "What is the chance that this sample is malicious, given the result of the previous sample?". Mathematically this is defined as $P_i(M|T_i, T_{i-1})$. This can be determined by making use of Bayes' theorem:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B|A)P(A) + P(B|\neg A)P(\neg A)}$$

The approach can also be extended to include more than just one historic sample. For sample $(i-2, i-3, \dots, i-n)$ this then becomes:

$$P(M|T_i, \dots, T_n) = \frac{P(T_i, \dots, T_n|M) \cdot P(M)}{P(T_i, \dots, T_n|M) \cdot P(M) + P(T_i, \dots, T_n|\neg M) \cdot P(\neg M)} \quad (7.4)$$

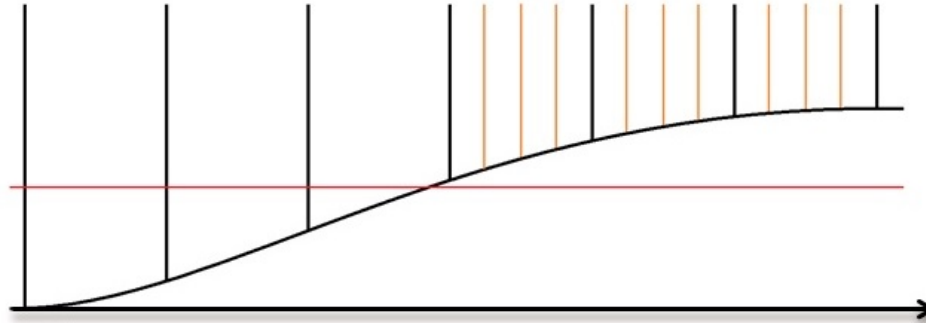
The expected benefit of this approach is that it lowers the false metrics, especially the false alarms, as it is increasingly unlikely for subsequent samples in a row to be wrongly identified. As such it will make the system less sensitive to spiked behaviour. When a normal process starts displaying anomalous behaviour however it will take more time to identify this correctly as the history used by the method still contains the non malicious samples suggesting a false positive has resulted as opposed to an actual malicious sample. This will cause an increase in the time it take to detect malicious behaviour is occurring.

DYNAMIC SAMPLING ADJUSTMENT

The initial experiments and subsequent analysis have shown that the sample rate influences the time it takes to detect malicious events. In an ideal world the sample rate would thus be near continuous. Unfortunately this is not feasible as it would drain resources and overload operators with false alarms. Not all the detection strategies are 100% infallible.

This problem could be circumvented by implementing a more dynamic approach to the sample rate however. The idea is to supply a default and fast rate to each event block and then have the IDS set the actual sample rate based on a boolean statement. This boolean statement depends on the output of the detection strategies, where an malicious sample leads to the maximum rate and a non malicious sample leads to the

default sample rate. To prevent that the rate flickers because of false detection, the boolean statement should include some sort of "timer" before reset to default is allowed. This dynamic improvement is represented by the figure below:



The benefits of this approach are that of an improved detection time while not having extra resource requirements nor overloading operators with false alarms. The downside is however that this will not decrease the time until first detection, but speed up the confirmation due to having more malicious samples coming in. As an example: When the sample rate is once every 10 minutes it would normally take another 10 minutes for additional information, with a dynamic sample rate this could be decreased to minutes or even seconds.

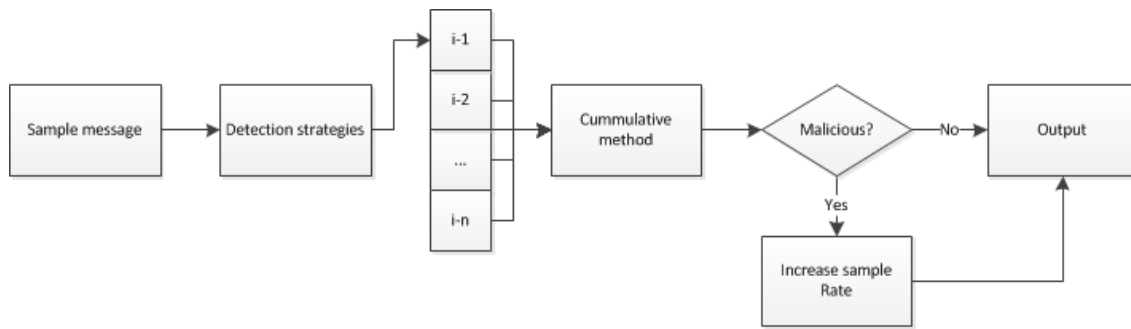
COMBINING METHODS

The Bayesian improvement has the downside that it will take longer before malicious behaviour has been detected because it requires multiple samples to convince the algorithm. The benefit of the dynamic sample rate is that it is possible to increase the sample rate when an anomalous event is thought to be ongoing. This then might negate the negative effects of the Bayesian method. However such an increased sample rate will increase the risk for more false metrics which might overload operators, which is likely negated due to the lower sensitivity of the Bayesian approach. If both proposed improvements are combined it seems possible to gain the benefits they bring, while at the same time suppressing the downsides. The next section investigates the combined improvements after adding them to the prototype.

7.5. DYNAMIC BAYESIAN EXPERIMENTS

The previous section analysed the results from the initial experiments and used this knowledge to find improvements to the prototype. These prototypes are to be implemented in a new detection scheme and then used to rerun the experiments. This section aims to do just that.

The improved analysis scheme implements both suggested improvements as part of the analysis block, which mainly functions as it did before. Once an message is received by the analysis block the sample information is passed to the detection strategies which determine if a malicious sample has been found. The difference here lies in what happens next as now the output from the analysis block does not solely depend on the finding of the detection strategies. It is determined by the cumulative results of previous samples. As such it is possible that the analysis returns not malicious even though the detection strategies indicate a malicious sample was found. At the same two internal triggers are flipped. The first is that the sample rate is increased to the pre-determined limit which will increase the received number of sample for the specific field device. Second, the detection strategies output is stored and will be used with the next sample as input for the cumulative method. These are thus not based on the analysis output but the detection strategy.



After making the required changes to implemented the improvements a new set of experiments have been run. For each seed a range of Bayesian lengths has been tested, starting with only 2 historic samples increasing up to and including 32 samples. The expectation here is that the Bayesian will have a positive effect on the true positive- and false alarm- rates. This means a rise in the true positive rate and a lowering of the false alarms.

The improved experiments are the same as those from the initial experiments, the one change is the addition of the improved detection scheme. Because it has been shown that the sample rate has no effect on detection rates it has been left out of the equation here.

TRUE POSITIVE RATE

After rerunning the experiments for the improved detection scheme the true positive rate has again been determined including the relation to the length of the Bayesian history. The resulting graph that can be created is displayed in figure 7.26. The graph itself represents all three experiments, including the different deviations which are part of the manipulation experiment.

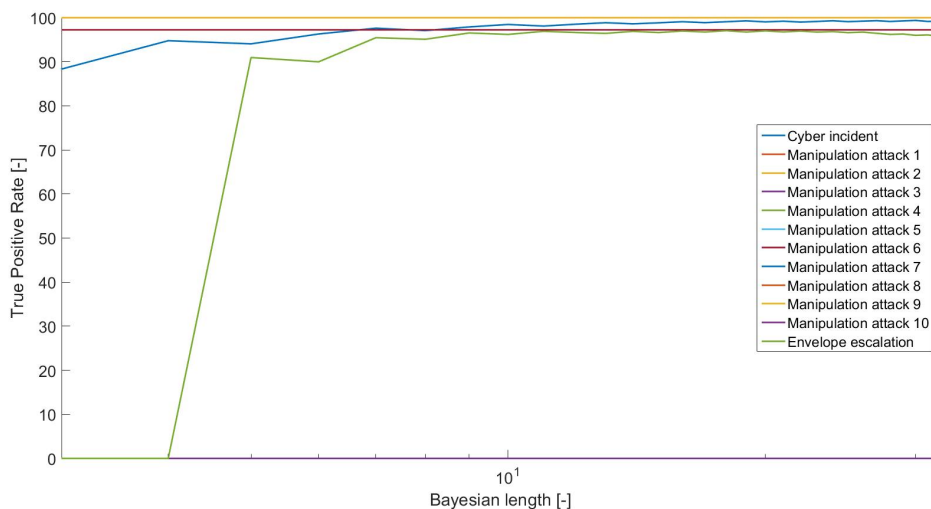


Figure 7.26: Improved true positive rate

All the experiments demonstrate that the dynamic Bayesian improvement starts with a positive correlation to the Bayesian length for these three experiments. After a certain point, which is different for each detection strategy, this rise eventually flattens out and the changes become negligible, especially considering the logarithmic X-axis. There are some interesting behaviours which can be identified. Best noticeable is that the envelope escalation starts out a 0% at a length of 4 and goes up to 90% and above for increasing lengths. This means that the added improvement detects no true positives for a small length in this case. What is also interesting is that the manipulation attack, with a deviation of 10% does not return any true positives regardless of Bayesian length.

Although the current results look good and there is a positive relation to the length of the Bayesian vector, this does not mean the improvements are a positive thing. To this end it is important to compare the improved results to that of the initial experiments. The table below does just this by listing the TPR for the initial and improved experiments.

Experiment	Initial [%]	Improved [%]
Cyber Incident	88	99.4
Manipulation attack	100	100
Envelope escalation	92	97.1

The table above shows that the cyber incident improved by more than 10% total by adding a dynamic Bayesian detection scheme. That means a perceptual improvement of 12% simply by looking at historic values. This significantly increased detection rate demonstrate the improvements which can be gained through the improved scheme. This improvement is different for the manipulation attack. While this was previously already 100%, it is unchanged. What has changed however is that for most of the deviation values the TPR has also gone up to 100%, while previously these were lower. Again indicating the benefit the improvement offers. There is one exception to this though, which was mentioned above. At a deviation of 10% there are no true positive which indicates that using the Bayesian improvement will have adverse affects if the detection rate is too low by itself. The reason for this is that there are not enough true positive samples available to convince the Bayesian from their validity. The envelope escalation experiments initially demonstrated a 92% detection rate, which increased to 97% by adding the Bayesian improvement, which is an improvement of 5%. Near the end of the graph a slight drop in the TPR can be noticed, it is unknown to what extend this continues but it is expected behaviour. When the history becomes to long it is increasingly difficult to convince the Bayesian that a value is indeed malicious - because more malicious values are required.

FALSE ALARM RATE

Mirroring the expectation of the true positive rate an improvement to the false alarm rate was expected. This means a decreased FAR after implementation of the dynamic Bayesian. After running the experiments with the new detection scheme the new false alarm rate is computed. The results for all three experiments are represented by figure 7.27. Here the FAR has been plotted against the increasing Bayesian vector length used to filter out wrong detection results.

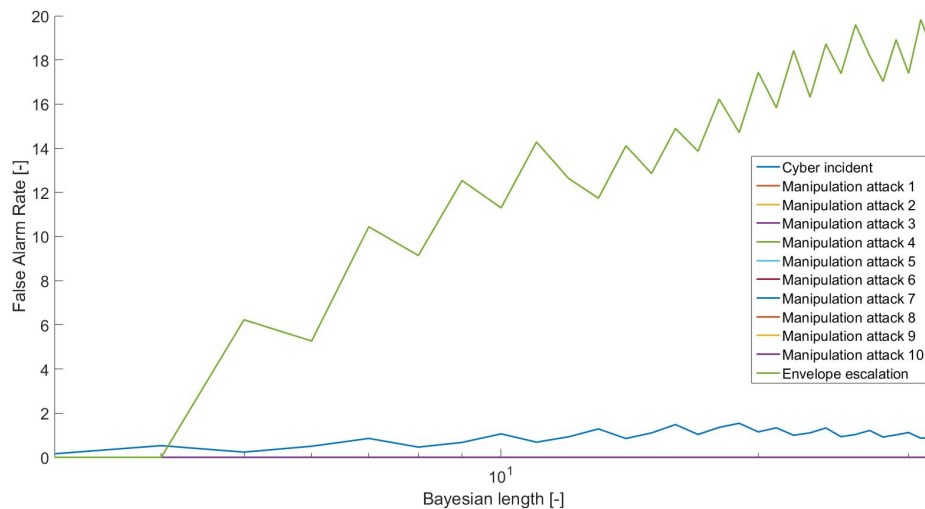


Figure 7.27: Improved false alarm rate

The figure indicates a distinct difference between the three experiments. The cyber incident shows a minor rise with increasing vector length, whereas the manipulation attack remains at a rate near 0% and the envelope escalation seems to increase logarithmically. These results seem negative as a decrease in the FAR was expected. This cannot be concluded without first looking at the actual change with respect to the initial experiments. The table below does exactly this by presenting the initial results next to the improved.

Experiment	Initial [%]	Improved [%]
Cyber Incident	1.31	0 - 1.54
Manipulation attack	$9.5 \cdot 10^{-5}$	0 - $1.6 \cdot 10^{-4}$
Envelope escalation	10.7	0 - 19.8307

This table demonstrates that addition of the dynamic Bayesian can have a positive impact on the FAR, but the net effect will depend on the length of the history used.

Adding the dynamic Bayesian improvement changed the cyber incident false alarm rate from 1.31% to 0% in the best situation, a considerable change demonstrating no false alarms within the system. The observed increase for a larger length is due to the fact that when there are relatively many false positives originating in the detection scheme it is possible that the Bayesian gets overloaded by false positives and thus is convinced much easier that a value is malicious. For the manipulation attack adding the improved scheme changed the results from $9.5 \cdot 10^{-5}\%$ to 0%. This outcome is positive and as expected, the Bayesian method filters out the false positive samples and correctly identifies these are true negatives. This result differs from the envelope escalation outcome and indicates that further investigation is needed to determine where this differing behaviour originates from. The envelope escalation experiment shows a decrease in the FAR for when there is a small history, but this quickly increases and surpasses the initial value when the length increases. The reason for this is as was described in the cyber incident. Having too many false positives near one another as input for the bayesian can overwhelm the sytem and cause it to falsely believe samples are malicious. This is further investigated next.

BAYESIAN PARAMETERS

Implementing and testing the dynamic Bayesian improvement has revealed three things worth discussing as it directly relates to the obtained results.

Missing vector length The first anomaly that should be discussed is that in figure 7.26 and 7.27 there are no results for a Bayesian vector of length 2. The reason for this is that under certain circumstances the Bayesian scheme will never mark a sample as malicious when there are only two samples, even when both samples say a malicious event is occurring. This can be demonstrated using the detection rates for the initial cyber incident experiment and equation 7.4. This computes the following:

$$P(M|T_1, T_2) = \frac{0.85 \cdot 0.85 \cdot 0.001}{0.85 \cdot 0.85 \cdot 0.001 + 0.05 \cdot 0.05 \cdot 0.999} = 0.22$$

With a threshold of 0.95 this means that even with both samples indicating a malicious state, the Bayesian will never be convinced and thus report non-malicious. Because of this both the number of true positives as false positives will be zero. When calculating the TPR and FAR this also means that both of these will be zero.

Historic vector length In the FAR of the envelope escalation improvement it has been observed that an increasing length of the history causes a (significant) increase in the FAR. The reason for this is that the raw data behind the detection strategy reports to have detected many malicious samples near one another. When the length of the history is sufficiently long a lot of these will be taken into consideration and can trick the Bayesian into believing there is indeed a malicious sample. This all while the reported samples were false positives that should have been filtered out.

Importance of input parameters An important lessons learned when running the experiments is that it is important to have a good estimate for the TPR and FAR used as input for the Bayesian method. The reason for this is demonstrated by the graphs 7.28 and 7.29, which represent the true positive rate and false alarm rate for the same detection method (cyber incident) but with varying input values. These values are listed in the table below.

Experiment	TPR [%]	FAR [%]
A1	88.7	98.7
A2	88.7	90
A3	88.7	80
A4	88.7	70
A5	70	98.7
A6	80	98.7
A7	99.0	98.7

The full extend of the impact that different input values have on the results and the effectiveness of the Bayesian has been left for future work. Indeed it would be good to have insight into the full range of improvements that can be expected when making use of this method.

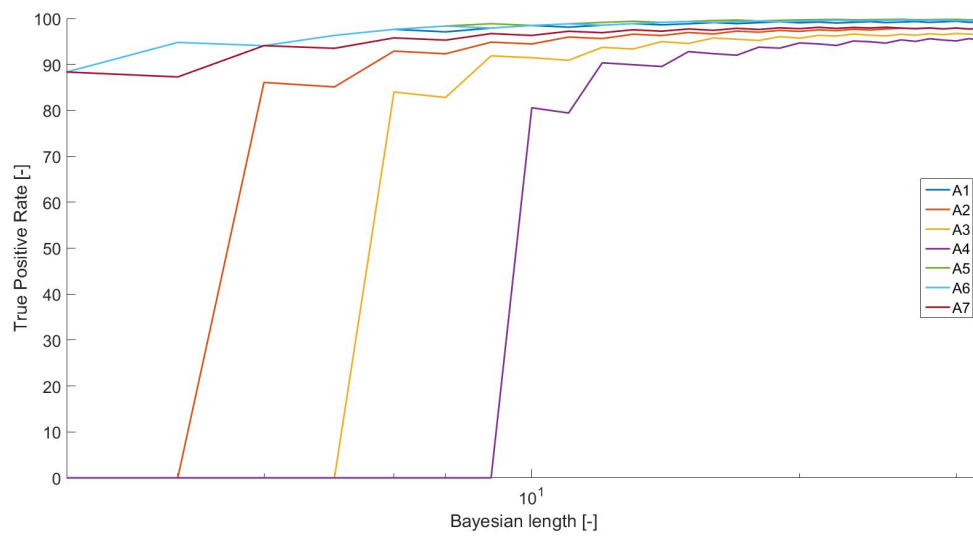


Figure 7.28: Influence of true negative rate on improved true positive rates

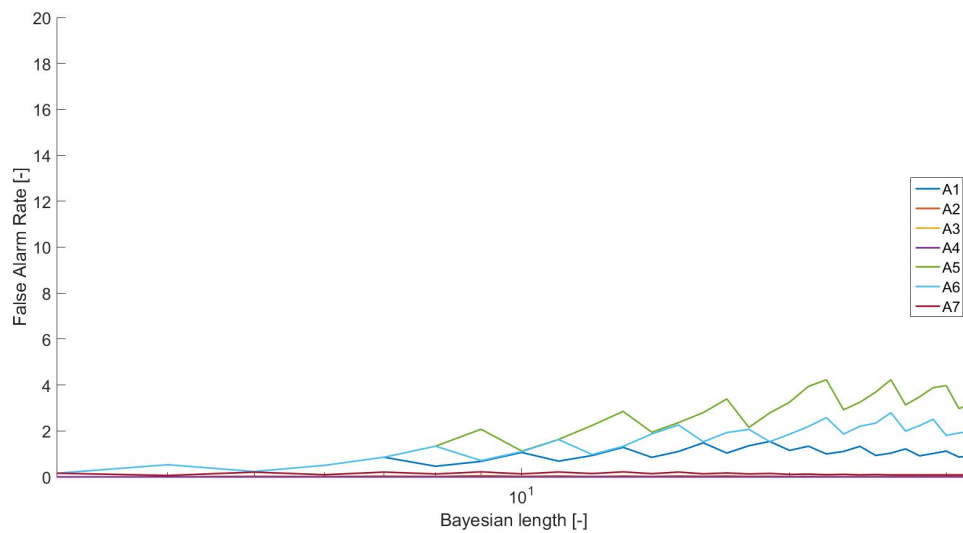


Figure 7.29: Influence of true negative rate on improved false alarm rates

7.6. DISCUSSION

This chapter demonstrated that the physical state of a process can be used to detect cyber attacks which trigger malicious behaviour, especially those that move the process towards a critical state. The used extendible design of the prototype allowed for a multi detection strategy, which has been reflected by the three experiments. The evaluation has shown that while each of these strategies has merit, they come with their own limitation.

For the value analysis strategy the experiments have indicated an detection rate of 88.7% of malicious samples and raises 1.37% false alarms. These results have been improved by implementation of the dynamic Bayesian approach, yielding a detection rate of 99.4% and a false alarm rate of 0% in the best case. The consistency analysis showed a detection rate of 100%, all malicious samples were detected for a deviation tolerance of 1%. This result decreases, considerably, for an increasing deviation value. A low deviation tolerance will be acceptable where it is known that external forces will hardly, if at all, influence the sample and ICS values. The obtained false alarm rate is $9.5 \cdot 10^{-5}$, indicating almost no false positives are raised. Applying the dynamic Bayesian method moved the FAR to 0%, where the bigger change was observed for larger deviation values which demonstrated significant improvements. A detection rate of 92% was first determined for the envelope escalation method, with a false alarm rate of 10.7. This FAR is far from satisfactory as out of every 100 non-malicious samples 10 will raise an alarm for no reason. Applying of the dynamic Bayesian method improved these results to a TPR of 97.1% and a FAR of 0, for the best cases. The FAR however demonstrated a significant increase for a larger the Bayesian history. This behaviour was explained due to overloading of the Bayesian method.

All three the detection strategies have proven to be usefull, especially when combined with the dynamic Bayesian improvement. It is important to note though that the proper configuration and working with the correct information on the physical processes being monitored are paramount for satisfactory results. This will require collaboration with a multi disciplinary approach / group of system designers, operators and security specialists.

The prototype has demonstrated that the design supports multiple detection strategies which are easily interchangeable. This will support scaling of the system when deployed on larger system. While anomalous behaviour might be detected, the current solution offers no way to prevent it from actually taking place or increasing in severity. For this to be possible controls mechanisms would have to be added that enable the IDS to steer the process back to a safe state. For the current concept attribution after an attack is not possible because the cause for the anomalous behaviour is unknown, however it does open up a path to look into possible attack vectors and warns when something is amiss.

Throughout the evaluation mention was made of the time it takes to detect a malicious sample. In the first experiments a range of sample times were taken into consideration while determining the true positive and false alarm rates, which are independent of sample time. However, the sample time will have an effect on when malicious behaviour is detected. This is because each sample from one source is correlated to the previous, bound by the physical system. With this realisation the likelihood that malicious behaviour is detected within a certain time-frame can be determine when the system is deployed.

7.6.1. FUTURE WORK

The current prototype demonstrated that it is possible to use knowledge about the physical processes under control by a ICS for intrusion detection purposes. Throughout the evaluation phase multiple interesting question arose which can be the base for future research. These will be discussed here shortly.

- In the current evaluation the detection strategies were used individually to enable the evaluation of the detection strategies. It would be interesting to see how they operate in unison and what the effect on the true positive and false alarm rates are for the whole system.
- At the base of this novel IDS are some assumptions, however the first thing that any intruder is likely to do is find a way to violate these. This makes it an interesting avenue to explore, to what extend do these assumptions hold in the field, how can they be circumvented and what are the potential consequences of this.
- Having a monitoring system that works with sensors comes with added benefits, namely it can be used for other purposes besides intrusion detection. Since the system essentially monitors the physical system it is possible to keep track of the wear and tear of various components and overall health of the system. This would make the concept a multi-use solution, making it a valuable addition to any ICS.

- The current strategies are based on boundary conditions which have been determined by analysing the system being monitored. With the current rise in machine learning and its applications it would be interesting to see how the strategies would perform when they are set by a machine learning algorithm instead.

8

CONCLUDING REMARKS

This chapter will summarize the contribution of this work, referring to the questions posted in Chapter 1. Additionally direction for future research within the Dredging industry, and the domain of ICS, with respect to cyber security will be presented.

8.1. KEY FINDING AND THEIR IMPLICATIONS

This work has touched upon multiple area's within the field of security and other industries. For each distinct topic the key-findings will be discussed here, starting with the state of the industry, followed my malware on industrial control systems and finally the main topic of a novel intrusion detection system.

STATE OF THE INDUSTRY

Starting this work little information was available that researched or discussed the current state of cyber security within the offshore and dredging industries. Interestingly a plethora of information is available towards safety, often accessible openly and shared within and outside of the field. Safety however also started out in the dark, with little information being available, which started to change by the raising of awareness on the importance of safety. In that regard the cyber security industry can take lessons from the development of the safety industry. Based on the available information the current state of cyber security within the offshore and dredging industry then displays a lack of awareness amongst industry professionals. Security is often seen as the thing that only restricts easier work-flow. Due to the limited information available, this area requires further research before definite conclusion can be made.

In an effort the further along such an investigation the CEDA¹ board has decided that a workshop for security professionals from the dredging industry will be organized. The aim behind this workshop is to enable a lively discussion and direct insight in the current state and if improvements with respect to (among other things) awareness is required. Within Heerema Fabrication Group the issue has also been raised by a presentation to the board of directors, which have led to efforts to get security further incorporate into the business processes.

Awareness however is not the only challenge that is faced by the industry. Most, if not all, equipment used throughout the maritime sector is depending on the safe and secure operation of the industrial control systems. There are case studies which demonstrate that a cyber incident within such a system (can) leads to incidents. These incidents could often have prevented if proper security hygiene would have been present. Incorporating security into the design of control systems and equipment is thus paramount and will warrant the extra cost and work involved. Collaboration and possible incorporation of these two fields can also be advanced by introducing security classes into educational curricula.

MALWARE AND ICS

Industrial control systems are increasingly important in our society, especially with the emergence of concepts such as the smart-grid and self driving cars. These systems are also operating at the core of much equipment used within the offshore and dredging industry and have long been thought to be safe from malware and cyber attacks. However a cyber attack on the Natanz nuclear enrichment facility has demonstrated

¹CEDA is an acronym for the Central Dredging Association.

what security professionals already feared. Such systems are just as susceptible to cyber attacks as traditional IT systems.

Research into the effect that malware infections can have on the operability and safety of industrial control systems is very limited however, as are publicly available case studies. Nonetheless, the risks that accompany a malware infection are severe and should not be underestimated. At this point in time only those with major resources backing them are likely to cause actual physical damage. With the discovery of Stuxnet however it is only a matter of time until this technology gets dissected and spreads over the internet. It is thus not only important that the security of ICS is taken seriously, but also that more research is undertaken to have a clear insight into the consequences that an infection by malware might have on facilities operated by ICS.

Part of this work has been the addition of a small section that serves as an example that getting access to malware is fairly straightforward due to the rise of malware as a service concepts. The demonstration makes use of the Zeus malware and demonstrates configuration, deployment and exploitation methods that present full system control to the attacker. Although Zeus targets traditional IT system, it is easy to imagine what happens when an engineering workstation gets infected. Worse yet would be the development of ICS specific malware as a service, which can lead to all sorts of extortion challenges.

INTRUSION DETECTION FOR ICS

As an effort to improve the state of security of industrial control systems this work introduced a novel concept design for an intrusion detection system. This novelty of the concept is that it takes advantage of the physical state of the physical processes. The question posed at the beginning of this work asked if it is possible to detect an intrusion into a control system by using state information, as opposed to the traditional method of using network traffic.

To be able to answer this question the concept has been build into a workable prototype. This prototype made it possible to get insight into the effectiveness of the novel approach and possibly determine design flaws made early in the process. Its design is based on the common intrusion detection framework, internal communication happens over a message bus and samples used to evaluate the malicious state of a process are taken with a certain sample rate.

These samples are processed by the detection algorithms present within the IDS. The prototype in this work contained three specific mechanisms: The consistency comparison verifies that the values reported by the control system are equal to the actual field device, this confirms that the control logic is not trying to keep operators in the dark. The value analysis has an internal list based on the engineering specifics that indicates the boundaries of parts and processes. If the field device reports a value near, or exceeding, this list an alarm should be raised as the process operates outside of expected condition. The envelope escalation method contains a multi-dimensional envelope, or boundary, that indicates safe operation condition. If a process moves out of this envelope an alarm will be raised.

After devising an evaluation plan the prototype and detection methods were evaluated, each method individually. To this end a simple model of a Trailing Suction Hopper Dredger was created which contained a controller that could be monitored by the prototype. Initial results indicated that while all three detection methods were working the results were lower than expected. To some extent improvement would be gained by further optimizing the boundaries themselves. However a second approach has been suggested and implemented, namely by taking previous results into consideration. This resulted in the dynamic Bayesian approach which significantly improved the initial results without optimisation of the boundaries used.

Throughout the process of creating and evaluating a novel IDS the following important aspects were uncovered. In traditional network approaches, when a malicious packet is missed the attack remains undetected. However when samples of the physical process are concerned there is a correlation with the past. This means that while the first sample might miss an ongoing attack, it is likely to be detected by the second. This raises the point that when working with a physical process, the time it takes to detect an attack also becomes important and enables engineers to fine-tune their systems. This factors into reaction time of the physical system: If the detection is well within the response time all is well. Seeing the importance of timing one of the recommendations for future work is to incorporate this into the evaluation model. Another realisation is that using a physical system also means that the way with which the IDS is to be evaluated will change. Making use of sample might seem to be essentially the same as network packets, but this is not the case. There is a high correlation between process samples which enable the use of Bayesian classification to improve detection rates and lower the number of false alarms send to system operators.

Looking at the result obtained by the evaluation of the prototype the conclusion is that using state information is viable way to perform intrusion detection, however the use of this approach will require more

research before being fully matured.

8.2. FUTURE AVENUES

There are several interesting research directions that have arisen during this work, which will be discussed next.

Awareness In chapter 3 the conclusion was made that the current level of cyber security awareness within the offshore and dredging industry are minimal. This was based on the limited articles, research and interviews. As such research that thoroughly investigates this level would be a good next step. The CEDA board is making a head-start into this direction by organising a workshop for industry professionals to gather their insights and opinions.

Education Even without any special technology it is possible to decrease the likelihood of a cyber breach and incident. This can be done by taking security into consideration early on throughout the design process. By applying existing technology it is possible to decrease incidents even further as the majority of security breaches can already be overcome. To this end it is worth while to consider building cyber security into the curriculum of courses such that students are introduced to the concepts early on. When we look at the neighbour field of safety this is exactly what happened, it has become an important part of any engineering course. A second benefit that introduction of security into courses will have is that it becomes easier for professionals from different fields to collaborate and communicate with each other.

Malware and ICS Chapter 4 indicated that currently there is limited information on the effects that traditional malware has on industrial control systems. Especially where it comes to multiple systems, as the existing research looked into one specific system. With the discovery of Stuxnet it was demonstrated that malware has made the transition from IT equipment to control equipment, which poses a serious threat because a lack of security will provide adversaries to bring down critical infrastructure on a whim. To make our systems more resilient against such attacks it is important to know how they respond to them, what might happen and what might not. To this end research that investigates the effect of malware on control systems would be help future endeavours in securing ICS.

Improvements to the IDS concept The current prototype has been evaluated using a simple simulated model of a TSHD, it would be interesting to observe if the same results would be found when operating on a live testing facility with actual controllers and field devices.

Based on the common intrusion detection framework the IDS should be able to work in a decentralised version, separated over multiple servers each in control of monitoring a specific subsystem. Although this is part of the design, it has not been part of the evaluation process. However with the size and distributed nature of some ICS's this would be a worthwhile endeavour to research.

Evaluating the prototype was done using a total of three experiments, each tailored to investigate a specific detection strategy. The next step would be to see how the system operates when all three detection strategies are operational at the same time and the system experiences both normal as malicious behaviour. This research would then also be able to take the detection time, the time it takes when an attack starts and when it is detected, into consideration. This was raised to be an important part of the process in this work but due to evaluation constraints not researched yet.

Business opportunity Using the physical process for intrusion detection has the benefit that the IDS has direct insight into the workings and state of a system. This insight can be leveraged as a source for business opportunity which can make adoption of the security solution much more attractive. One such example would be to measure the current wear levels of equipment, allowing for a better replacement scheduling. Shock pulse measurement is a vibration monitoring technology that does exactly that for industrial bearings. This measurement can then be used for wear and tear detection, which will decrease operational downtime, but also for malicious behaviour.

Another opportunity is to decrease commissioning time when large systems are being build. Because the IDS monitors the behaviour of both the physical process and the control system it has the potential to decrease commissioning time of these complex systems. Often the interaction between various subsystems

causes unintended problems and delays due to conflicting signals, the IDS might be able to pinpoint what is happening and where the pain point(s) can be found.



GLOSSARY AND BASIC DEFINITIONS

Within this work the following definitions and concepts are used. Most of which have been adopted as they are presented in the Guide on Information Security by the National Institute of Standards and Technology [106] and by Hadziosmanovic [38].

Industrial control system: An industrial control system is an information system used to control industrial processes such as manufacturing, product handling, production, and distribution.

Critical infrastructure: A critical infrastructure is a system and assets, whether physical or virtual, so vital to a nation that the incapacity or destruction of such systems and assets would have a debilitating impact on security, national economic security, national public health or safety, or any combination of those matters.

Threat: A threat is any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, individuals, other organizations, or a nation through an information system via unauthorized access, destruction, disclosure, or modification of information, and/or denial of service.

Threat scenario: A threat scenario is a set of discrete threat events, associated with a specific threat source or multiple threat sources, partially ordered in time.

Vulnerability: A vulnerability is a weakness in a system, system security procedures, internal controls, or implementation that could be exploited by a threat source.

Attack An attack is an attempt to gain unauthorized access to system services, resources, or information, or an attempt to compromise system integrity, availability, or confidentiality.

Incident An incident is an occurrence that actually or potentially jeopardizes the confidentiality, integrity, or availability of an information system.

Cyberspace: Cyberspace is a global domain within the information environment consisting of the interdependent network of information systems infrastructures including the Internet, telecommunications networks, computer systems, and embedded processors and controllers.

Cyber attack: A cyber attack is an attack via cyberspace, targeting an enterprise's use of cyberspace for the purpose of disrupting, disabling, destroying, or maliciously controlling a computing environment / infrastructure; or destroying the integrity of the data or stealing controlled information. Cyber security is the ability to protect or defend the use of cyberspace from cyber attacks.

Attack vector: A specific means through which an attack can target and potentially compromise a system. Multiple attack vectors are often called an attack surface.

Countermeasure: A countermeasure is a management, operational or technical control prescribed for an information system to protect the confidentiality, integrity, and availability of the system and its information. Intrusion detection is the process of monitoring the events occurring in a computer system or network and analysing them for signs of possible incidents.

Intrusion detection: Intrusion detection is the process of monitoring the events occurring in a computer system or network and analysing them for signs of possible incidents.

Malware: The handle for what is known as malicious software, which is software that is defined by its malicious intent. It goes against the system requirements as posed by the user.

Insiders Are those individuals whom have some level of legitimate access and (organizational) trust. E.g. employees and contractors.

B

PROTOTYPE UML

This appendix describes the UML for the main classes used in the prototype. To increase readability the class explanations themselves, as presented in 6.2, have been included in the corresponding section.

B.1. MESSAGE BUS

The message bus is the main communication channel between other classes and is responsible for the delivery of messages send on the channel to the correct subscribers. As was discussed previously the implementation follows that of a list based published subscribe mechanism. To make this happen the class maintains a list of subscribed classes to specific publishers in the "subscriptions" attribute. When a new message is published on the bus the class ensures the that it is delivered to the subscribers listed within the subscriptions list. In principle all messages will be treated by a first come first served basis, however to accommodate future priority traffic (e.g. for highly critical ICS subsystems and components) a priority lane is provisioned for. Table below represents the class diagram for the message bus class.

MessageBus	
-	subscriptions:Map<String, HashSet<MessageListener>
+	subscribe(MessageListener subscriber):void
+	subscribe(MessageListener subscriber, String publisher):void
+	unsubscribe(MessageListener subscriber):void
+	unsubscribe(MessageListener subscriber, String publisher):void
+	publishMessage(Message message):void
+	publishMessage(Message message, boolean priority):void
-	messageDeliveryHelper(Message message, boolean priority, String publisher):void

ATTRIBUTES

Description of the attributes used in the Message Bus class.

subscriptions This attributed provides a map between publishers and a set of subscribers which have a contract with the bus to be contacted when a message is send by said publisher.

METHODS

Description of the methods used in the Message Bus class.

subscribe: This method creates a contract between a *subscriber* and the message bus for its message delivery service. The optional "*publisher*" parameter is to be used when the listener is only interested in a specific listener. Leaving this parameter empty subscribes the listener to all messages on the bus.

unsubscribe: This method unsubscribes a MessageListener from the message bus. If a publisher is passed only that specific contract will be removed, if no publisher is given the all it's contracts will be removed.

publishmessage: This method publishes a message on the bus. The optional parameter can be used to indicate a priority message. Such a message is moved to the top of the list and as such gets priority over standard messages. This can be used to ensure a critical message gets delivered as soon as possible.

messageDeliveryHelper: This method helps with the delivery of messages to relevant subscribers. Essentially this is the paper-boy. If increased resources on the bus are required the first improvement to be made is to turn this worker into multiple multi-threaded workers.

B.2. EVENT BLOCK

The event block provides the connection between the field devices being monitored and the intrusion detection systems. As such these instances are seen as the eyes and ears of the IDS. Once new information is received, mainly due to taking an sample, the event block will wrap that information in a message container and publish it to the message bus for further processing. In the case of production this class can also be used for mapping purposes between field values and units understandable by the detection strategy.

EventBlock	
+	DEFAULT_SAMPLE_TIME:int
-	ID:long
-	IDCounter:AtomicInteger
-	monitoredDevice:FieldDevice
-	name:String
-	priority:boolean
-	publisher:String
-	ready:boolean
-	sampleRate:int
-	sampleValue:double
-	shutdown:boolean
+	connectFieldDevice(FieldDevice fDevice):void
+	run():void
+	setPriority(boolean priority):void
+	setSampleTime(int sampleTime):void
+	stop():void

ATTRIBUTES

Description of the attributes used in the Event Block class.

ID: The ID is a unique identifier that is used to track and identify a specific instance of the event block.

monitoredDevice: This is a description that is used to indicate which physical device is being monitored by the event block.

priority: The priority field indicates if an message from this event block should be treated with priority.

publisher: The publisher name used when a message is published on the message bus.

sampleRate: The sample rate determines the rate with which samples from the monitored device will be taken and send on the message bus.

sampleValue: When taking a sample from the monitored device, the sample value stores this value for further processing.

METHODS

Description of the methods used in the Event Block class.

run: Once the instance thread has been created calling this method will start running the thread. It will keep on running until shutdown is true;

setPriority: This method sets the priority of the event block and by extend thus the priority which published messages on the message bus should have.

setSampleTime: This method sets the time which should be adhered to between two individual samples.

stop: This method stops the thread from working by setting the *shutdown* attribute to true.

B.3. ANALYSIS BLOCK

The analysis block is the heart of the IDS. It is subscribed to event messages and is responsible for detecting potential malicious behaviour. This is done by adding specific detection strategies, such as those specified in previously, to the block. Once an event message has been received is is passed down to the added strategies, the result of which are added to an analysis report that is then wrapped in a message and published on the message bus for further response.

AnalysisBlock	
-	mBusMessageBus
-	mQueue:BlockingQueue
-	values:Map<String, Map<String, Double>>
-	timestamps: Map<String, Map<String, Timestamp>>
-	sourceElements:ArrayList<String>
-	strategies:ArrayList<DetectionStrategy>
+	addDetectionStrategy(DetectionStrategy strategy):void
-	addSourceElement(String sourceID):void
+	getSampleValues():Map<String, Double>
+	getSourceValues():Map<String, Double>
+	getTimestamps():Map<String, Map<String, Timestamp>>
+	getValues():Map<String, Map<String, Double>>
+	messageHandler(M message):void
+	requestSourceUpdate(String publisher, String source, String fieldDeviceID, Timestamp requestedTimestamp, long tsDeviation):boolean
-	updateInstanceState(M message, String database): void

ATTRIBUTES

Description of the attributes used in the Analyses block class.

mBusMessageBus: Contains the message bus from which event messages are received and to which result messages are published.

mQueue: This queue contains all messages received from the message bus which still require processing.

values: The values variable contains a map with values corresponding to a certain event block. This enables other event values to be used in detection strategies.

timestamps: Contains a maps with timestamps corresponding to when a sample in the values map was taken. This is used to trigger a value update when the values are too old.

sourceElements: Contains a list of all the event blocks that are of interest for this specific analysis block. If there are multiple analysis blocks in the IDS this is how messages can be dropped if they are not required.

strategies: Contains a list of all the detection strategies that are part of this specific analysis instance.

METHODS

Description of the methods used in the Analyses block class.

addDetectionStrategy: Add a detection strategy to this analysis block. The added strategies are used to determine if a sample is malicious or not.

addSourceElement: Add a source element (e.g. event block) to this block. This is used to determine which publishes to subscribe to on the message bus.

getSampleValues: Return a list of all sample values being tracked.

getSourceValues: Return a list of all source values being tracked.

getTimestamps: Return a map with all source and sample timestamps.

getValues: Return a map with all source and sample values.

messageHandler: When a message gets delivered to this block the handler determine if it should be placed on the message queue, it should be dropped, or another action should be undertaken.

requestSourceUpdate: This method publishes a message to the bus requesting the adapter to fetch a specific value stored in the historian of the ICS.

updateInstanceState: This method updates the instance state, which is based on all source elements being tracked, to its newest state.

B.4. RESPONSE BLOCK

The response block is tasked with handling the reports made by the analysis blocks. It determines the response when malicious behaviour is detected, which ranges from waiting for more information, reporting to the operators or in high risk cases even place the control system into a secure state. The latter will be out of scope for the current prototype version however.

ResponseBlock	
-	mBus:MessageBus
-	mQueue:BlockingQueue<Message>
-	logs:ArrayList<Log>
-	updateLogs(AnalysisResultMessage message):void
+	void messageHandler(M message):void
+	getLogs:Arrayist<Log>

ATTRIBUTES

Description of the attributes used in the response class.

mBus: Contains the message bus from which event messages are received and to which result messages are published.

mQueue: This queue contains all messages received from the message bus which still require processing.

logs: Contains an list with all the analysis result logs that the instance received. These are used to present the operator with the latest IDS information and are used to keep a historic record.

METHODS

Description of the methods used in the response class.

updateLogs: Adds the latest detection strategy log to the list of logs, after determining if the result is positive or negative.

messageHandler: When a message gets delivered to this block the handler determine if it should be placed on the message queue, it should be dropped, or another action should be undertaken.

getLogs: Returns the list of logs currently maintained by the thread.

BIBLIOGRAPHY

- [1] World Economic Forum, *The global information technology report 2014 - rewards and risks of big data*, (2014).
- [2] A. Quan-Haase and B. Wellman, *Hyperconnected net work*, The firm as a collaborative community , 281 (2006).
- [3] D. Maheshwari, *Robust Offshore Networks for Oil and Gas Facilities*, Ph.D. thesis, Delft University of Technology (2010).
- [4] R. R. R. Barbosa, *Anomaly detection in SCADA systems: a network based approach*, Ph.D. thesis, University of Twente, Design and Analysis of Communication Systems (DACS), Enschede (2014).
- [5] World Economic Forum, *Risk and responsibility in a hyperconnected world*, (2014).
- [6] World Economic Forum, *Global risks 2015 - 10th edition*, (2015).
- [7] CBS, *Fbi director on threat of isis, cybercrime*, CBS' "60 Minutes" (2014).
- [8] The United States Department of Justice, *Attorney general eric holder speaks at the administration trade secret strategy rollout*, .
- [9] W. Vlegels, *Analysis of Cyber Security Aspects in the Maritime Sector*, Tech. Rep. (ENISA, 2011).
- [10] N. Falliere, L. O. Murchu, and E. Chien, *W32.Stuxnet Dossier*, Tech. Rep. (Symantec, 2011).
- [11] S. Johnsen, *Mitigating accidents in oil and gas production facilities*, in *Critical Infrastructure Protection II*, The International Federation for Information Processing, Vol. 290, edited by M. Papa and S. Shenoï (Springer US, 2008) pp. 157–170.
- [12] C. P. Pfleeger and S. L. Pfleeger, *Security in Computing*, 4th ed. (Prentice Hall, 2006) ISBN-13: 978-0132390774, ISBN-10: 0132390779.
- [13] X. van der Voort, *Interview with mr x van der voort, owner vandervoort cyber security*. (2014).
- [14] S. Parkin, *The boy who stole half-life 2*, Eurogamer (2014).
- [15] B. D. Snow, *We need assurance*, in *Cryptographic Hardware and Embedded Systems*, Lecture Notes in Computer Science, Vol. 1717 (Springer Berlin Heidelberg, 1999) pp. 1–1.
- [16] R. Langner, *To Kill a Centrifuge - A Technical Analysis of What Stuxnet's Creators Tried to Achieves*, Tech. Rep. (The Langner Group, 2013).
- [17] T. J. Holt and M. Kilger, *Know your enenemy: The social dynamics of hacking*, The Honeynet Project , 17 (2012).
- [18] N. Anderson, *Confirmed: Us and israel created stuxnet, lost control of it*, (2012), last accessed on 01/09/2014.
- [19] B. Clayton and A. Segal, *Addressing Cyber Threats to Oil and Gas Suppliers*, Tech. Rep. (Council on Foreign Relations, 2013).
- [20] McAfee, *Global Energy Cyberattacks: "Night Dragon"*, Tech. Rep. (McAfee Foundstone Professional Services and McAfee Labs, 2011).
- [21] A. Förster, *Maulwürfe in nadelstreifen*, Henschel (1997).
- [22] C. Blask, *Ics cybersecurity: Water, water everywhere*, Blog (2011).

- [23] M. Abrams and J. Weiss, *Malicious control system cyber security attack case study-maroochy water services, australia*, (2008).
- [24] T. Smith, *Hacker jailed for revenge sewage attacks*, (2001), last accessed on 02/09/2014.
- [25] Z. Shauk, *Malware on oil rig computers raises security fears*, (2013).
- [26] I. N. Fovino, A. Carcano, M. Masera, and A. Trombetta, *An experimental investigation of malware attacks on scada systems*, Critical Infrastructure Protection (2009).
- [27] M. Luallen, *Breaches on the Rise in Control Systems: A SANS Survey*, Tech. Rep. (SANS, 2014) a SANS Analyst Survey.
- [28] G. McDonald, L. O. Murchu, S. Doherty, and E. Chien, *Stuxnet 0.5: The Missing Link*, Tech. Rep. (Symantec, 2013) version 1.0.
- [29] C. Bronk, *Hacks on Gas: Energy, CybersecCyber, and U.S. Defense*, Tech. Rep. (James A. Baker III Institute for Public Policy, 2014).
- [30] J. Langill, *Cyber security risks, 'heartbleed', open ssl vulnerability & ics*, in *KIACS 2014* (2014).
- [31] D. Goodin, *Aws console breach leads to demise of service with "proven" backup plan*. (2014), last accessed on 03/07/2014.
- [32] ICS-CERT, *Secure architecture design*, website, last Accessed: 19/05/2015.
- [33] *Bacnet - data communication protocol for building automation and control networks*, (2015).
- [34] *A dnp3 protocol primer*, .
- [35] *MODBUS TCP/IP messaging implementation guide.*, Modbus Organization (2015).
- [36] *Profibus - specifications and standards*, .
- [37] IEC, *International standard iec 60870-5-104*, .
- [38] D. Hadziosmanovic, *The Process Matters: Cyber Security in Industrial Control Systems*, Ph.D. thesis, Universiteit Twente (2014).
- [39] M. Cheminod, L. Durante, and A. Valenzano, *Review of security issues in industrial networks*, Industrial Informatics, IEEE Transactions on **9**, 277 (2013).
- [40] K. A. S. K. A. Stouffer, J. A. Falco and K. Kent., *SP 800-82: Guide to Industrial Control Systems (ICS) Security*, Tech. Rep. (NIST, 2013).
- [41] J. Weiss, *Cyber security of industrial control systems*, Youtube (2012), last Accessed: 22/04/2015.
- [42] M. Manion and W. M. Evan, *The y2k problem and professional responsibility: a retrospective analysis*, Technology in Society **22**, 361 (2000).
- [43] A. A. Cárdenas, S. Amin, and S. Sastry, *Research challenges for the security of control systems*, in *Proceedings of the 3rd Conference on Hot Topics in Security, HOTSEC'08* (USENIX Association, Berkeley, CA, USA, 2008) pp. 6:1–6:6.
- [44] A. Cardenas, S. Amin, B. Sinopoli, A. Giani, A. Perrig, and S. Sastry, *Challenges for securing cyber physical systems*, in *Workshop on Future Directions in Cyber-physical Systems Security* (DHS, 2009).
- [45] Z. Shauk, *Hackers hit energy companies more than others*, blog (2013).
- [46] W. Knowles, D. Prince, D. Hutchison, J. F. P. Disso, and K. Jones, *A survey of cyber security management in industrial control systems*, International Journal of Critical Infrastructure Protection **9**, 52 (2015).
- [47] R. Lee, *I am robert lee, author of "scada and me" and a cyber security expert - ama!* Forum (2014), last Accessed: 19/05/2015.

- [48] K. Wilhoit, *Who's Really Attacking Your ICS Equipment? (Part 1)*, Tech. Rep. (2013) research Paper.
- [49] K. Wilhoit, *Who's Really Attacking Your ICS Equipment? (Part 2) - The SCADA That Didn't Cry Wolf*, Tech. Rep. (2014) research Paper.
- [50] J. Storms, *Interview with mr. j storms; manager information management at heerema marine contractors*, (2015).
- [51] M. Menting, *PetroSecurity in the Digital Era: Legacy Systems vs. Cyber Threats*, Tech. Rep. (ABI Research, 2013).
- [52] National Institute of Standards and Technology, *Security maturity levels*, (2014).
- [53] SANS, *SANS Securing The Human - 2015 Security Awareness Report*, Tech. Rep. (SANS, 2015).
- [54] N. Christin, S. Egelman, T. Vidas, and J. Grossklags, *It's all about the benjamins: An empirical study on incentivizing users to ignore security advice*, in *Financial Cryptography and Data Security*, Lecture Notes in Computer Science, Vol. 7035, edited by G. Danezis (Springer Berlin Heidelberg, 2012) pp. 16–30.
- [55] M. Krotofil and A. Cárdenas, *Resilience of process control systems to cyber-physical attacks*, in *Secure IT Systems*, Lecture Notes in Computer Science, Vol. 8208, edited by H. Riis Nielson and D. Gollmann (Springer Berlin Heidelberg, 2013) pp. 166–182.
- [56] BSI, *The state of it security in germany 2014*, (2014).
- [57] M. Balduzzi, K. Wihoit, and A. Pasta, *Hey captain, where's your ship? attacking vessel tracking systems for fun and profit*, in *Hack in The Box Asia* (2013).
- [58] *Superyacht gps spoofing*, Website (2013), last Accessed: 21/05/2015.
- [59] *Offshore drill rigs at threat from computer viruses*, Website (2013), last Accessed: 21/05/2015.
- [60] C. Wueest, *Targeted Attacks Against the Energy Sector*, Tech. Rep. (Symantec, 2014).
- [61] C. Bronk and E. Tikk-Ringas, *Hack or Attack? - Shamoon and the Eveloution of Cyber Conflict*, Tech. Rep. (James A. Baker III Institute for Public Policy, 2013).
- [62] Pandalabs, *Operation "Oil Tanker" - The Phantom menace*, Tech. Rep. (Panda, 2015).
- [63] D. Parker, M. Lawrie, and P. Hudson, *A framework for understanding the development of organisational safety culture*, *Safety Science* **44**, 551 (2006).
- [64] P. Hudson, *Safety management and safety culture - the long, hard and winding road*, .
- [65] D. Rider, *Shipping plans cyber guidelines*, Website (2015).
- [66] F. Cohen, *Computer Virusus*, Tech. Rep. (Computer Security: A Global Challenge, 1984).
- [67] W. H. Ware, *Security Control for Computer Systems*, Tech. Rep. (RAND Technical Report, 1970).
- [68] J. P. Anderson, *Computer Security Technology Planning Study*, Tech. Rep. TR-73-51 (Air Force Electronic Systems Division, 1972).
- [69] J. F. Shoch and J. A. Hupp, *The "worm" programs; early experience with a distributed computation*, *Commun. ACM* **25**, 172 (1982).
- [70] A. Young and M. Yung, *Cryptovirology: extortion-based security threats and countermeasures*, in *Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on* (1996) pp. 129–140.
- [71] Q. Liao, *Ransomware: A growing threat to smes*, Southwest Decision Sciences Institute (2008).
- [72] A. Gazet, *Comparative analysis of various ransomware virii*, *Journal in Computer Virology* **6**, 77 (2010).
- [73] D. Bradbury, *The metamorphosis of malware writers*, *Computers & Security* **25**, 89 (2006).

- [74] F. De La Cuadra, *Malware: The geneology of malware*, Netw. Secur. **2007**, 17 (2007).
- [75] B. Baker and A. Chiu, *Threat spotlight: Rombertik - gazing past the smoke, mirrors, and trapdoors*, blog (2015), last Accessed: 19/05/2015.
- [76] D. Piscitello, *Conficker Summary and Review*, Tech. Rep. (ICANN, 2010).
- [77] A. Schmidt, *At the boundaries of peer production: The organization of internet security production in the cases of estonia 2007 and conficker*, Telecommunications Policy **36**, 451 (2012), global Internet Governance Research and Public Policy Challenges for the Next Decade.
- [78] J. Dittmann, B. Karpuschewski, J. Fruth, M. Petzel, and R. Munder, *An exemplary attack scenario: Threats to production engineering inspired by the conficker worm*, in *Proceedings of the First International Workshop on Digital Engineering*, IWDE '10 (ACM, New York, NY, USA, 2010) pp. 25–32.
- [79] *Bonn discovers partial solution to conficker infections*, Network Security **2009**, 2 (2009).
- [80] F. Leder and T. Werner, *Know Your Enemy: Containing Conficker - To Tame A Malware*, Tech. Rep. (The HoneyNet Project, 2009).
- [81] P. Porras, *Inside risks: Reflections on conficker*, Commun. ACM **52**, 23 (2009).
- [82] Mandiant, *APT1 - Exposing One of China's Cyber Espionage Units*, Tech. Rep. (Mandiant, 2013).
- [83] J. Thomson, *Chapter 3 - cyber security, cyber-attack and cyber-espionage*, in *High Integrity Systems and Safety Management in Hazardous Industries*, edited by J. Thomson (Butterworth-Heinemann, Boston, 2015) pp. 45 – 53.
- [84] E. D. Knapp and J. T. Langill, *Chapter 3 - industrial cyber security history and trends*, in *Industrial Network Security (Second Edition)*, edited by E. D. K. T. Langill (Syngress, Boston, 2015) second edition ed., pp. 41 – 57.
- [85] P. Shakarian, J. Shakarian, and A. Ruef, *Chapter 8 - duqu, flame, gauss, the next generation of cyber exploitation*, in *Introduction to Cyber-warfare*, edited by P. S. S. Ruef (Syngress, Boston, 2013) pp. 159 – 170.
- [86] I. Ion, R. Reeder, and S. Consolvo, “...no one can hack my mind”: *Comparing expert and non-expert security practices*, in *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)* (USENIX Association, Ottawa, 2015) pp. 327–346.
- [87] *Cybersecurity research: Addressing the legal barriers and disincentives*, (2015).
- [88] K. Scarfone and P. Mell, *Guide to Intrusion Detection and Prevention Systems (IDPS)*, National Institute of Standards and Technology (2007), special Publication 800-94.
- [89] E. M. Hutchins, M. J. Cloppert, and P. Rohan M. Amin, *Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains*, in *6th Annual International Conference on Information Warfare and Security* (2011).
- [90] D. Denning, *An intrusion-detection model*, Software Engineering, IEEE Transactions on **SE-13**, 222 (1987), also in Proc. of the 1986 Symp. on Security and Privacy, IEEE Computer Society, April 1986, pp 118-131.
- [91] V. M. Iguire, S. A. Laughter, and R. D. Williams, *Security issues in {SCADA} networks*, Computers & Security **25**, 498 (2006).
- [92] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, *Attacks against process control systems: Risk assessment, detection, and response*, in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '11 (ACM, New York, NY, USA, 2011) pp. 355–366.
- [93] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez, and E. Vazquez, *Anomaly based network intrusion detection: Techniques, systems and challenges*, Computers & Security **28**, 18 (2009).

- [94] S. Etalle, C. Gregory, D. Bolzoni, E. Zambon, and D. Trivellato, *Monitoring Industrial Control Systems to improve operations and security*, Tech. Rep. (Security Matters, 2013) an overview of the threats to Industrial Control Systems and the technologies to protect them.
- [95] S. Etalle, C. Gregory, D. Bolzoni, and E. Zambon, *Self configuring deep protocol network whitelisting*, Tech. Rep. (Security Matters, 2013) game-changing technology that boosts ICS security while reducing operational costs.
- [96] A. Cardenas, J. Baras, and K. Seamon, *A framework for the evaluation of intrusion detection systems*, in *Security and Privacy, 2006 IEEE Symposium on* (2006) pp. 15 pp.–77.
- [97] A. Cardenas, S. Amin, and S. Sastry, *Secure control: Towards survivable cyber-physical systems*, in *Distributed Computing Systems Workshops, 2008. ICDCS '08. 28th International Conference on* (2008) pp. 495–500.
- [98] I. N. Fovino, A. Carcano, T. D. L. Murel, A. Trombetta, and M. Masera, *Modbus/dnp3 state-based intrusion detection system*, in *International Conference on Advanced Information Networking and Applications* (2010).
- [99] A. Carcano, I. N. Fovino, M. Masera, and A. Trombetta, *State-based network intrusion detection systems for scada protocols: A proof of concept*, in *Critical Information Infrastructures Security*, Vol. 6027 (Springer-Verlag, 2010).
- [100] A. Carcano, A. Coletta, M. Guglielmi, M. Masera, I. N. Fovino, and A. Trombetta, *A multidimensional critical state analysis for detecting intrusions in scada systems*, in *TRANSACTIONS ON INDUSTRIAL INFORMATICS*, Vol. 7 (IEEE, 2011) pp. 179–186.
- [101] C. Doerr and J. M. Hernandez, *A computational approach to multi-level analysis of network resilience*, in *Proceedings of the 2010 Third International Conference on Dependability, DEPEND '10* (IEEE Computer Society, Washington, DC, USA, 2010) pp. 125–132.
- [102] C. Doerr, *Challenge tracing and mitigation under partial information and uncertainty*, in *Communications and Network Security (CNS), 2013 IEEE Conference on* (2013) pp. 446–453.
- [103] A. Giani, G. Karsai, T. Roosta, A. Shah, B. Sinopoli, and J. Wiley, *A testbed for secure and robust scada systems*, *SIGBED Rev.* **5**, 4:1 (2008).
- [104] H. Christiansson and E. Luijff, *Creating a european scada security testbed*, in *Critical Infrastructure Protection*, IFIP International Federation for Information Processing, Vol. 253, edited by E. Goetz and S. Shenoi (Springer US, 2008) pp. 237–247.
- [105] S. Dragt and M. Luchs, *Oe5671 - dredging equipment design project*, (2013).
- [106] A. G. G. Stoneburner and A. Feringa, *Risk Management Guide for Information Technology Systems*, *NIST Special Publication 800-30.*, Tech. Rep. (National Institute of Standards and Technology, 2002).