NTNU – Trondheim
Norwegian University of
Science and Technology

Sikkerhet i Industrielle netverk

Bjørn Ingemann Henninen

Master of Science in Computer Science
Submission date: June 2015
Supervisor: Guttorm Sindre, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

NTNU – Trondheim
Norwegian University of
Science and Technology

Industrial Control Systems Security - The Impact of Standards and Regulations

Bjørn Henninen

Submission date: June 2015
Responsible professor: Guttorm Sindre, IDI
Supervisor: Mate Csorba, Marine Cybernetics

Norwegian University of Science and Technology
Department of Computer and Information Science

Abstract

This thesis is about Industrial Control Systems (ICS) security and standards and regulations which can contribute to improve this. The thesis tries to answer which research is done on how standards would have impacted previous attacks and / or prevent future attacks, and how this research could have been conducted. This is done in three steps. First the standards and regulations are described, and similarities and differences between them are discussed. Secondly, the most known attacks on ICS systems are described. The thesis argues that this is an area where more research needs to be done, before a thorough and useful comparison between attacks and standards can be done. Lastly, the thesis looks at the development and argumentation within a selection of relevant research on ICS security against the previous mentioned standards. This gives some new insight, foremost that there is a lack of insight and research into how attacks compare against relevant security standards. A small table for comparing this is suggested as a potentially solution to the problems mentioned in the thesis. The thesis also found that standards should be viewed as a tool, rather than as a solution for improving ICS security in itself.

Sammendrag

Denne masteroppgaven handler om industrielle kontroll systemers (ICS) sikkerhet og standarder og regelverk som kan bidra til å forbedre dette. Oppgaven diskuter forskningen som til nå er gjort på hvordan standarder har påvirket tidligere angrep og / eller hindre fremtidige angrep, og hvordan videre forskning kan bli gjennomført. Dette er gjort i tre trinn. Først blir standarder og forskrifter beskrevet, og likheter og forskjeller mellom disse blir diskutert. Deretter er det en gjennomgang av de mest kjente angrep på ICS systemer. Masteroppgaven argumenterer for at dette er et område der mer forskning må gjøres før en grundig og nyttig sammenligning mellom angrep og standarder kan gjennomføres. Til sist ser avhandlingen på utvikling og argumentasjon innenfor et utvalg av relevant forskning på ICS sikkerhet, satt mot de tidligere nevnte standardene. Dette gir noe ny innsikt, først og fremst at det er mangel på forskning om hvordan angrepene kan sees i sammenheng med relevante sikkerhetsstandarder. Oppgaven foreslår en liten tabell for å sammenligne dette som en potensiell løsning på de problemer som er nevnt i oppgaven. Oppgaven viser også at standarder og reguleringer bør sees som et verktøy, snarere enn som en enkeltstående løsning for å løse ICS sikkerhet problemer.

Acknowledgements

I would really like to thank everyone that has helped me along with this thesis. When I started I did not know exactly what I got my self into and now it is done. I would like to thank my supervisor for suggesting this thesis even if it is not what was originally suggested, thank you Mate Csorba. I would like to thank my main supervisor, professor Guttorm Sindre for being patient with me and for the great help. I want to thank Magne and Anja for help, when help was most needed - I would not have finished without you. Lastly I want to thank Kristin for major support with the project and for staying up and helping beyond what I ever could have expected, thank you.

-Bjørn Henninen

Contents

List of Acronyms

AIS Automatic Identification System.

CA certification authority.

CIP critical infrastructure protection.

DCP Discovery and Configuration Protocol.

DCS Distributed Control Systems.

DOS Denial of service.

ECDIS Electronic Chart Display and Information System.

GPS Global Positioning System.

GUI Graphical user interface.

HMI Human Machine Interface.

ICMP Internet Control Message Protocol.

ICS Industrial Control Systems.

IEC International Electrotechnical Commission.

ISO International Organization for Standardization.

NERC The North American Electric Reliability Corporation.

NIST National Institute of Standards and Technology.

OWASP The Open Web Application Security Project.

PLC Programmable logic controller.

PTCP Precision Time Control Protocol.

RTU Remote Terminal Unit.

SCADA Supervisory Control and Data Acquisition.

WEP Wired Equivalent Privacy.

List of Figures

Chapter 1 Introduction

1.1 Motivation

Stuxnet, a worm from 2009, put Industrial Control Systems (ICS) security on the map and it also peaked an interest in the security community. It was the first custom-made cyberweapon targeted at national installation. I have always been interested in computer security, but given the consequences of a successful maleficent attack, the Stuxnet incidents spiked my interest as well. What is the state of security after Stuxnet, has there been any similar attacks, and how are ICS systems being protected today were all questions I became interested in. As I started to read to find answers to those questions, I got increasingly frustrated with the lack of solutions to the issues raised in the articles I read. It seemed to me that the articles only pointed out which security issues there were, but that they often neglected to point to potential solutions. My pre-project, and the proofreading of another master thesis, made me aware of the fact that there were some standards which were created as potential solutions to the issues raised by several of the articles. I decided I wanted to know more about what kinds of regulations and guidelines there are for the prevention of attacks and how they can help national and private companies to protect themselves. And thus the topic for my thesis was decided. The extra focus on security in ICS can be seen as the increase in disclosures of vulnerabilities as well in 1.1

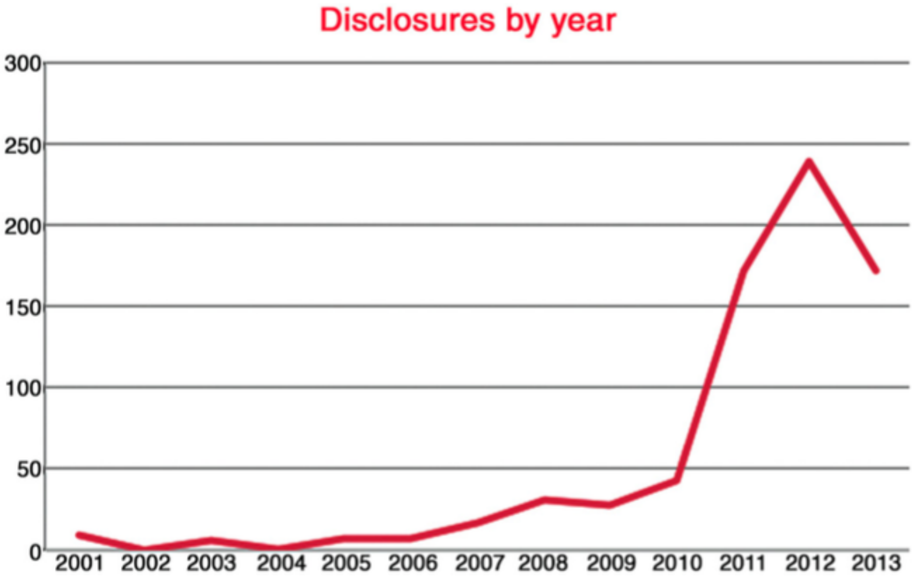


Figure 1.1: Number of disclosures over time [1, p. 54]

1.2 Problem description and limitations

The first set of ICS was created to automate processes in industrial and electrical plants. The first ICS protocols were created with efficiency in mind and with as little as possible overhead since bandwidth was limited. This left the protocols without several of the security features expected from modern communication protocols. Features such as authentication and encryption, which mitigates trivial attacks, and which seems to be able to cause major damages in an ICS setting, is excluded from the protocols. What I want to find out is if there is a system used for classifying attacks and problems, and mapping them against standards and regulations. The mapping would be to see which standard would mitigate the problems, the damages, or both.

1. RQ1 Is there a approach for finding which standard would mitigate or prevent a given attack?
2. RQ2 What standards are used for security in ICS and how do they compare?
3. RQ2.1 How do newer articles on ICS security compare to current standards?
4. RQ3 What kind of attacks are leveraged against ICS?

1.3 Research method

This thesis consists of a review of the most relevant concepts and academic literature on Industrial Controls Systems (ICS) security. The ICS security research area is a steadily increasing field, and thus it has not been possible to review all regulations, standards and articles which have been written about, and on, ICS security. The focus of this thesis is, as the main research question states, selection of standards which could mitigate or prevent an attack, and the selected articles and concepts are therefore related to this area.

This thesis provides an overview of four of the main regulation and standards on ICS security. Those four are presented and discussed because they are some of the most used and the leading regulations and / or standards in this field. The backgrounds for the regulations and standards are also varied in how they are created, for which purpose they were created, and how often they are reviewed. This lays the foundation for a discussion about which areas they cover, similarities and difference, and what the standards contribute towards.

The articles in this thesis are selected because they illustrate the problems which exist in ICS security. The seven articles are selected because they all highlight and illustrate the larger security issues in ICS and the impact those issues could have. The articles further contributes to showing why more evaluation of the regulation and standards on ICS security, such as this thesis, is necessary. Furthermore, the articles are also selected because they give a broad specter of issues which can be discussed in relation to the four standards and regulations presented in this paper. Six of the articles that are selected are written in the last five years, which means that the issues they discuss are contemporary issues. However, the last article is written in 2006. This article is included because it will be able to set a context for the contemporary articles, which will make it possible to discuss how the focus on the security issues have changed in the last 10 years.

1.4 Structure

This will be a short walk-through of how this thesis is structured and what each chapter contains. Chapter 2 will give an introduction to ICS. This will consist of what parts are present in an ICS and what protocols are present in the network. The chapter will also give an introduction to which standards and organizations are present for ICS security. The chapter will end with a discussion of the different standards and regulations, and how they compare to each other. Chapter 3 will give an overview of the most important incidences regarding ICS security. Stuxnet, zotob Doqu, Flame and Night Dragon will be discussed more thoroughly, with a focus on their impact on computer security research. Some other threats and attacks will also be mentioned. Chapter 4 will shortly go through common attacks and common attackers, with an emphasis on how attacks can be classified. This will give a short introduction to the types of attacks that are the most common, but the chapter is kept short since the discussions around this area would be enough to cover a thesis in itself. Chapter 5 will go through methods for finding vulnerabilities in ICS. The chapter looks at some of the programs which exist for this purpose, what the different programs can do, and what the differences are between them. Chapter 6 will review newer articles on ICS security. The articles will then be discussed against each other, with the purpose of seeing where the main issues in ICS security lays. Chapter 7 will discuss those articles and compare their view and discussion on security issues with what the standards and regulations say they should be able to do. The discussion in this chapter will look at the differences between what the standards do, and where the main security concerns are.

Chapter 2 Introduction to Industrial Control Systems

Industrial Control Systems (ICS) is a collective term for control systems used in industry. ICS is used in several different areas and for several different types of systems, such as power plants, traffic regulations and nuclear facilities. This gives it a potential for impact on our every day life. First, this chapter will introduce ICS and the terms related to it. Then the current standards and regulations that can impact ICS security will be introduced. This will show how diverse the regulations and standard on the topic is, and it will lay the foundation for later discussions. The chapter will give an understanding of what ICS is and which/how standards are used to increase security.

2.1 Industrial Control Systems

The terms ICS and Supervisory Control and Data Acquisition (SCADA) are often used interchangeably, which may cause some confusion. However, despite the popular misuse, ICS refers to any systems, devices, networks, and controls used to operate and/or automate an industrial process[1]. The term ICS is a general term that encompasses several types of control systems. This includes SCADA systems, Distributed Control Systems (DCS), and other control system configurations, such as skid-mounted Programmable logic controller (PLC)s, which is often found in the industrial sectors and critical infrastructures [7]. This means a SCADA system is an ICS, but not the other way around. Each of the different ICS has their own security considerations and policies, but as time has passed the different systems have evolved, and the lines between them have become more blurred. For instance SCADA and DCS have come closer, and both systems now monitors and reads data to represent them to an operator in order to control industrial processes or manufacturing.

2.1.1 Areas

ICS are used in several areas, such as nuclear power plants, electrical distribution, traffic regulation, production and industry. An attack on any of those systems could directly impact daily life as we know it. ICS are used in industries such as communications, electricity, oil, gas and water and they all depend on ICS to function. This also includes chemical facilities and nuclear facilities.

2.2 Operations and Assets

As some of the terminologies now have been discussed, an introduction to each of the parts that a ICS can consist of, and how these parts function together, will follow. A typical layout of an ICS can be seen in 2.1

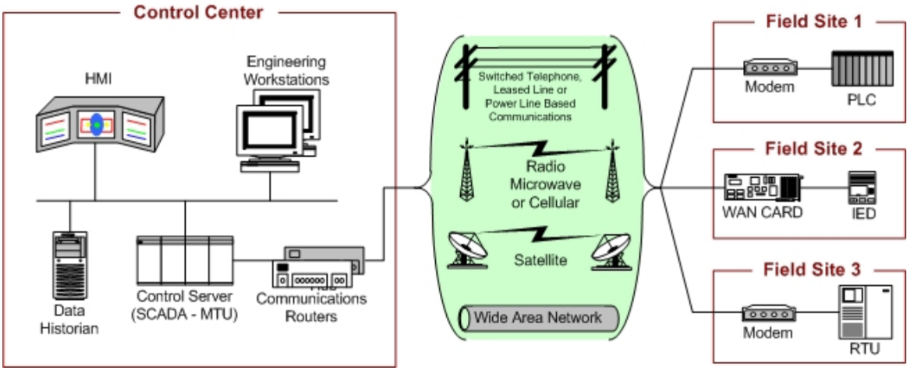


Figure 2.1: A general layout of a SCADA system [2, p. 2-7]

2.2.1 Programmable Logic Controllers

A programmable logic controller is an industrial computer that automates production in a facility. They are built to operate in the rough environment that is the production line. They are built to fit in whichever type of facility they are needed in and with special input and output designed for several types of production. Most PLC run specific programs for their job designed just for that production line. They are connected to the sensors and motors needed to automate the production. As PLC operate in real time they are designed for doing a specific job at a set time, according to its logical programming. The programming language used in most PLC are given by the IEC 61131-3 standard. The standard was first published in 1993 and has been updated twice, first in 2003 with version 2, and in 2013 with version 3.

2.2.2 Remote Terminal Units

The Remote Terminal Unit (RTU) is, as the name suggests, a unit that is remote and sends data back to a central. The central could be a PLC and Master Terminal Unit (MTU), or a straight to the Human Machine Interface (HMI). It could be connecting through a link that is not always sending data, such as a radio link cellular data connection, or other similar links. Most RTU communicate through industrial protocols and thereby they overlap somewhat by being both PLCs and RTUs. As such the RTU can be seen as a remote PLC that might be connecting to the network on a schedule.

2.2.3 Human Machine Interfaces

HMI is what the operators use to check PLC and RTU. The HMI gives the operators a graphical view of what is going on. Through the HMI the operator can interact with the process and stop it if needed. The HMI can run modern operating systems such as Windows, and can be a common computer or a specialized hardened computer with a touchscreen. It can be seen as a Graphical user interface (GUI) for overlooking the complex logic underneath. There is seldom any authentication for the consoles since this might cause problems in an emergency situation.

2.2.4 Workstations

Workstations enable monitoring of the system, such as the HMI, but with limited access to control the system. These supervisory stations can reside in a variety of locations throughout the industry network or business networks. When a workstation monitors a control system remotely, it is important to make special considerations when establishing, controlling and monitoring the connection between the workstation and the ICS, as the supervisory system could otherwise easily become an open attack vector to the ICS[1, p67].

2.2.5 Data Historians

A data historian is a software system that is typically used for storing trend style data. That is, point values, events and alarms and other information over time. The data collected is stored centrally within a database and is often referred to as tags. Data historians may contain information used by both business and industrial networks, and can therefore pose a security risk as a data historian in a less secure zone (the business network) could be used as a vector into more secure zones

2.3 Protocols

A protocol is, in an ICS context, a standard for network messages where the standard gives the syntax and the semantics of the messages sent. Which there are several different protocols that are present in an ICS environment. Some are vendor-specific while some are open protocols used by several vendors. Most of the vendor specific protocols, is proprietary. Examples of this is the Simens S7. This proprietary aspects of the protocols makes any review of the protocol difficult, since it has to be reverse engineered to be understood. In a security perspective will this mean that a complete security check can not be done on the protocol, since only parts of the protocol will be known to the engineers.

2.3.1 Modbus

The Modbus protocol is one of the oldest protocols. It was designed already in 1979, but it is still widely used throughout the ICS setting. "The Modbus serial communications protocol is a de facto standard designed to integrate PLCs, computers, terminals, sensors and actuators."[8] The protocol operates at the application layer of the OSI model, where it uses a request and reply method. This is used for gathering sensor data to more complex systems in the network. It is also used for giving overview data from PLC to the HMI.

2.3.2 OLE for Process Control

The OPC is a technical specification that defines a set of standard industrial software interfaces based upon Microsoft's OLE/COM technology [9]. It is designed to simplify integration of various forms of data on systems from different vendors. The original standard was released in 1996 and included standard sets of objects, interfaces, and methods to support interoperability in industrial applications. This original set of standards is today commonly referred to as "OPC Classic", and is still one of the most widely deployed OPC specifications. One of the security concerns regarding the OPC classic is that it is rooted in the Windows operating system and thus suffers the risk of attack through exploitation of any vulnerability inherent to this OS. Microsoft has stopped support for older operating systems such as Windows XP (support for Windows XP Service Pack 3 ended in April 2014). This may introduce significant risks into OPC systems running this operation system.

2.4 Standards and Regulation

This section will introduce the relevant standards and regulations, and give a brief overview how they aim to improve SCADA security. There are several other standards and regulation which could have been mentioned, such as the Federal Information Security Management Act and the Homeland Security Presidential Directive Seven. The standards selected here are the ones where the majority of the regulations are relevant for ICS security, rather than just minor parts. Some of the standards are mandatory, and organizations that are not complying with them might be penalized by serious fines. Other standards are rather to be considered "best practices", and they are thus not mandatory to implement. Industrial networks are important to most nations, but not all regulate them. Other nations however, regulates them heavily. An example would be in North America where The North American Electric Reliability Corporation (NERC) gives strict directives for security in the North American bulk electric system through NERC CIP. Most standards aim to contribute to improve security, but overlapping standards might prove difficult to enforce. Some standards, such as the ISO 27002, is being developed by several of the countries that later implements them and financial situations, participation possibilities and interests therefore affects participation and impact on the outcome of the final standards. This mitigates some problems, as local law and regulation is covered in the standard by those who participated in the creation or establishment of them. It is therefore easier to implement these standards for the participating countries.

2.4.1 NERC CIP

NERC CIP is a regulation which is most known for being harshly enforced with severe fines for not complying with the NERC critical infrastructure protection (CIP). Still, it is a highly respected standard which is compatible with most other security enhancing standards. The NERC CIP standard has mandatory sections which regulates eight areas[10], which are:

1. Recovery Plans for Critical Cyber Assets
2. Systems Security Management
3. Incident Reporting and Response Planning
4. Physical Security of Critical Cyber Assets
5. Electronic Security Perimeter(s)
6. Personnel & Training
7. Security Management Controls
8. Critical Cyber Asset Identification

There are planned expansions for fourteen new sections where three are already considered pending regulatory filing. One of the upcoming sections is on physical security where the rest is regarding cyber security. The planned expansion will make the regulation even more extensive, which could cause some problems in regards to implementation of the regulation. The planned expansions will, on the other hand, ensure that the regulation cover a very wide range of issues.

2.4.2 ISO/IEC 27002

The ISO/IEC 27002:"Information technology – Security techniques – Code of practice for information security controls" standard is created by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). ISO is an organization which consists of businesses, industries and governments which are represented through national standardization organizations. The organization develops and creates standards, and their standards on information technology are just one of many in their management standards catalogue. The Electrotechnical Commission (IES) is a similar organization to ISO, but they focus on standards within the field of electrotechnical equipment and related issues. Both organizations have in common that they are consensus based and voluntary. This means that participation in the standardization work is up to each national standardization organization's interests, and that a certain percentage of the participants in the standard creation work have to agree to the standard before it can be published[11].

The ISO/IEC 27002 standard "gives guidelines for organizational information security standards and information security management practices including the selection, implementation and management of controls taking into consideration the organization's information security risk environment(s)."[12], according to ISOs own website. The standardization work is done in the US, and the standard was last updated in 2013. It is not a straight map to good security, but focuses on assessing risk and policies regarding security. This is a common trait of ISOs management standards, and it stems from their consensus and voluntary based approach to standard development[11].

2.4.3 NIST SP 800-82

The National Institute of Standards and Technology (NIST) has created a security guide for ICS called "Guide to Industrial Control Systems (ICS) Security". Its latest issue was released in May 2013, but this version is up for a second revision. This guide is not a regulation, but a set of NIST recommendations. This guide contains six chapters, where the first chapter is an introduction, the second chapter is an overview of ICS. The third chapter goes trough characteristics of ICS, vulnerabilities and risk factors. The fourth chapter is on how to develop a security program. The fifth chapter goes more in-depth on how to do networking in an ICS setting, this also includes recommendation for network segregation, defence in depth and recommendation for firewall rules. The last chapter is on ICS security controls which entail how to manage the security program, a set of operational controls such as incidence response and awareness training, and finally technical controls such as authentication, access control and audits.

2.4.4 ISA/IEC-62443 (former ISA-99)

ISA/IEC 62443 is a series of standards on ICS security. The standards are divided into four groups which fills out a broad range of topics on how to implement a secure setup. The standard was started by the International Society of Automation (ISA) and in the Standards and Practices Committee 99. This is why the standard was formerly known as ISA-99. The standard work is now being done in cooperation with the International Electrotechincal Commission, and it is known as ISA/IEC 62443, Industrial Automation and Control System Security. The standards are in the process of being developed, and it is therefor not a finished group of standards [13].

The first group for this standard is the general group that aims to identify terminology, metrics, and models. The second group deals with policies and procedures for creating a safe security program. The third group focuses on the cyber security technologies, which also covers risk management. The last group covers components needed for a secure program.

2.5 Standards, Regulations and Their Impact

As we have seen there are several different regulations and standards related to ICS security. They cover a wide range of issues and they have a varied background. The implementation requirements and procedures, the participating creators and the areas they focus on are wide-ranging. In light of this it is important to ask which impact the amount of standards and regulations have on security issues. A standard developed by a consensus based group, such as the IEC/ISO standard and the ISA/IEC standards, is more likely to gain accept than a national based standard might be. The interests and knowledge which is included in the founding of the standard will probably make the consensus based standards from international organization wider reaching and more acceptable. If the ultimate goal is to have one standard which cover all security areas, then those kind of standard might be a better solution. However, consensus-based standards are less rigorous than national standard or regulation, because those types of standards needs to cover a wider range of interests. his could mean that the standards run the risks of being interpreted and implemented different in different organizations, and that the standards therefor might not necessarily cover the security risks which would be the most beneficial. A regulation with harsh fines would often be more rigorous and more likely to, when implemented, cover the areas it was intended for, but it might for the same reasons be harder to implement successfully. The difference between the two approaches to standardization work might mean that the best solution would be to implement both consensus-based standards and stricter regulations, as to make sure that a wider range of security is covered.

A quantitative analysis of security standards on ICS found that "compared to SCADA standards ISO/IEC 17799 focus more on management and organizational issues, and less on technical issues."[14, p. 6]. This ISO standard is now called ISO 27002. The more technical standards like NERC CIP focuses more on managing specifics of the security program. Organizational standards such as ISO 27002 has a focus on creating a security culture in the organization where it is implemented. The organizational approach has a better chance of creating a security aware staff, that might cause employees to report suspicious activity at a lower threshold. A second benefit from creating a security culture is that such a culture also will have a perspective on risk and risk mitigation. ISO 27002 and NIST 800-82 has chapters on risks while ISO 27002 focuses a bit more on managing them NIST 800-82 has a section dedication to what are the risks.

Chapter 3 Incidences

ICS networks exist in our daily life without most of us knowing how it affects us. This chapter will introduce some of the incidences that have occurred and which impact they have had on ICS security. As mentioned, one of the motivations for this thesis was the widespread consequences a malicious attack could have had. A brief description about what kind of attacks, and what level of damages is at stake, is necessary to put the security discussion in a context. This will therefore be included in this chapter. Although some of the attacks described in this chapter are lesser known, the damages were severe. The structure of this chapter is not a reflection of the severity of the incidents. However, the order of the incidences is based on the connections and similarities between the attacks. Stuxnet, the most influential and best known of the attacks, is presented first. This will also be the most thorough description of any of the accidents. Not all of the attacks presented in this chapter caused widespread damages, but they are an indication of what could potentially happen.

3.1 Stuxnet

The best known attack at a SCADA system is Stuxnet. The first versions were observed as early as in 2008[15]. It is suggested that the worm was developed by the US or Israel, and up to 60 percent of the infected computers were in Iran. Thus, computer security has also become an international relations issue [16]. Stuxnet is known for being the most advanced attack ever devised in cyberspace, and it is considered to be the first worm directed at ISC. Stuxnet leveraged several vulnerabilities and used advanced coding to infiltrate and cover its own tracks. Stuxnet is a worm which propagates trough networks and installs itself on all Windows machines. Then it uses 0-days vulnerabilities to escalate privileges as to root-kit the machine to create a persistence presence. Once it is installed, it looks for the WinCC and Step 7 software, which are used to configure Siemens PLC S7 and S3 series. Stuxnet continues by infecting USB-keys so as to reach machines that are not connected to the network. It has two modes of attack; one for S3, looking for the specific PLC of type 6ES7-315-2, and one looking for a S7-413. All in all, Stuxnet was the first to use four 0-day exploits, compromising two digital certificates and injecting malicious code into industrial controllers, and finally hiding it from the operator.

Even after being analyzed it was feared that incorrect removal could cause damages. "Stuxnet could be used to cause a significant amount of damage if it is not properly removed,"[17] this was later rebuffed as Siemens posted that customers had removed it without damages [18]. The outcome of Stuxnet might have been as much as a thousand centrifuges damaged or destroyed in the Natanz Enrichment Plant [19].

3.2 Doqu

Doqu is most likely written by the same organization that wrote Stuxnet. It is presumed that as much as 50% of the source code is from Stuxnet[20]. Doqu is not directed to cause damages in ICS, which is one of the biggest differentiation between Doqu and Stuxnet. What Doqu does is logging activity and keystrokes on the infected computers. The malware infects trough several different methods, among them Word documents as an attachment. Doqu uses the information it gathers to infect more machines on the same network. Similar to Stuxnet, Doqu reports back to a control server, and by connecting to other infected machines it can access the open Internet from blocked zones[21]. This is illustrated in 3.1. The level of sophistication and obfuscation causes the malware to be hard to track and detect. At the time this is written Duqu 2.0 has just been detected and it has just attack Kaspersky Lab with a stolen Certificate[22].

3.3 Flame

Flame, Skywiper or Win32.Flame was first identified and analyzed in 2012, but it has probably been infecting machines undetected since 2010 [23] (A dll found in it even suggest it has been around since 2007). While Stuxnet has been given attention as the first cyber weapon, Flame is considered the most advanced. Flame is considerably bigger with its 20 Mb, compared to Stuxnet's 500 Kb. Flame is similar to Stuxnet and Duqu, and Kaspersky labs have proven that the developers are connected [24]. Flame, like Doqu and Stuxnet, infects machines and propagates through network and USB-keys. While Stuxnet targeted a specific PLC, Flame is focused on stealing information. Through several modules it can log keystrokes, take screen captures, listen to any connected microphones, and read and write both Bluetooth and USB units. Due to the external command structure new modules can be uploaded to the virus. The command centrals used among 50 domain-names with changing ip-addresses, in addition to using several different cryptographically

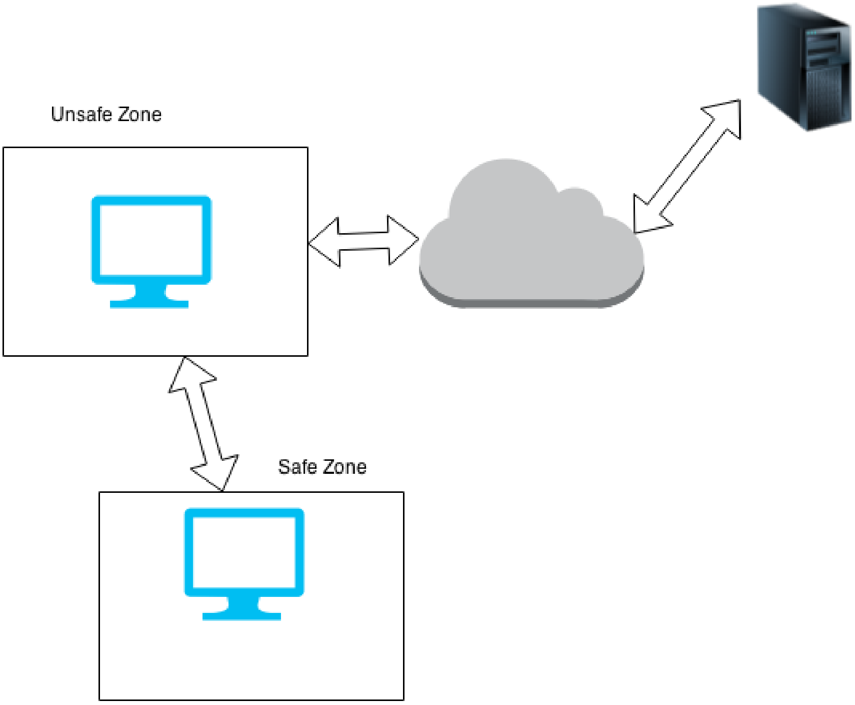


Figure 3.1: Representation of how Doqu accesses the internet from a safe zone

techniques to hide itself. It also has over 300 different tactics for evading security on the local machine. Since the program is this advanced it would have taken a lot of effort to put it together, which might indicate that a government is behind it.

3.4 Night Dragon

In February 2011 [25], McAfee disclosed the discovery of a series of attacks against energy, petrochemical, and oil companies. The attacks were traced back to mainly Chinese addresses and are believed to have started in 2009. The attack was so well hidden that it was not exposed for two years. The attack was using SQL injection techniques for the servers exposed to the Internet. While basic in nature, SQL injections might result in gaining usernames and passwords for further penetrating the network. After the initial attack the attacker used the compromised servers to access the internal network and compromise more internal servers. The paper outlines Night Dragon that had command and control servers and Remote Administration Toolkits (RATs). The attack might not have caused the physical damages that Stuxnet did, but it did manage to exfiltrate sensitive information such as financial documents regarding exploration sites and bid information.

3.5 Zotob

The malware Zotob was first discovered in 2005 and was directed at a vulnerability in Microsoft Windows 2000 [26]. Windows 2000 was, and still is, used as the underlying operating system in some control systems. The malware was developed by two men in their twenties, which caused the security community to reconsider how easy it was to develop malware. Zotob used a vulnerability in the plug and play feature that made it possible to access and control computers remotely. The malware was distributed over the Internet, and it spread fast. It also spread to control systems as some of them had Windows as the underlying operating system. Zotob was made with the guidance of the MS05-039 security advisory, which disclosed the vulnerability. The attack caused damages even if a patch was available since most companies are slow to apply such patches. Zotob caused damages that on average cost 97000$ to fix in addition to 80 hours of work to clean up for each site that was affected.[27]

3.6 Other Control systems

Marine control and auxiliary systems are exposed to threats equal to those of drilling or process control systems. The technologies used for navigation on ships has shown weaknesses in research reports. There have been incidents such as the tilting of an oil rig off the coast of Africa, or the standstill of control systems during the relocation of a rig due to malware infection[28]. In July 2013, a team from the University of Texas showed that it is possible to spoof Global Positioning System (GPS) signals and change a ship's direction by feeding the navigation system false coordinates[29]. The attack was performed outside the coast of Italy and is summarized in a youtube video[30]. On January 17th, 2013, the USS Guardian was stranded in the Philippines, which supposedly was caused by incorrect charting. The flaw might have been in the Electronic Chart Display and Information System (ECDIS) that was discovered by NCC[31]. The anti-virus vendor Trend Micro has found and demonstrated weaknesses in the Automatic Identification System (AIS), which allowed them to be able to tamper with, or even shut down, communication between a vessel and the port authority, just using a cheap radio kit available at stores selling radio equipment[32]. There are possibilities to detect this spoofing[33], but it will take some time before regulations can catch up.

The tracking system for ships, known as AIS, is used for several tasks, such as:[32]

- Collision Avoidance
- Maritime Security
- Aids to Navigation (AtoN)
- Search and rescue, Accident investigation
- Binary messages, e.g. weather forecasting

This article[32] shows that simple attacks such as man in the middle and replay attacks work against AIS and that they could cause damages.

3.7 Summary

All of the attacks mentioned have or could cause severe damages. Stuxnet is an example of this, when the worm caused the Natanz Enrichment plant to shutdown and it also damaged as much as thousand centrifuges. Zotob is another example because it caused damages which ended up costing on average around 97000$ for each of the hit sites. There has not jet been an attack which was intentionally directed towards causing human casualties, as it has been theorized by movies such as Die Hard 4[34], but this is a possibility in the future. As this chapter have shown, there are several other types of attacks which have been conducted, and several areas where standards might potentially have helped, which will be discussed in later chapters.

Chapter 4 Attackers and attack methods

There is a saying that goes "knowing is half the battle", which comes into mind when discussing computer security. Anyone who works on computer security needs to know what the common attacks are, and who the attackers are, to be able to do as a good job as possible. This chapter will introduce the types of known attacks and attackers. The chapter will start with defining the different types of attacks before it goes through different definitions of types of attackers. This will make it easier to discuss the problems addressed by the articles which will be presented later in this thesis.

4.1 Attackers

Attackers can be grouped in external or internal attackers, and further in intentional or unintentional attackers. An internal attacker would be a person with information about the system and the security mechanisms. Usually they have physical access to the system which might render some security measures useless. An intentional internal attacker might also have valid user accounts and thereby access to critical systems. While using valid user accounts accessing systems and changing them might not raise suspicion among the security team. An internal unintentional attacker would be one that attacks the system without knowing it, which might consist of opening a phishing email or using a USB-key infected with malicious code.

External intentional attackers could be professional attackers or as seen in 3.5 non professionals. While a professional attacker might tailor his attack to a specific protocol, system, or even a specific setup as seen in 3.1. There are websites such as exploit-db[35] that has exploit code available for use, which lowers the threshold for less technical attackers to perform attacks on systems. An external unintentional attacker would be an infected machine that is used to attack another system. Most such attacks are classified as a denial-of-service attacks and are described in 4.3.2

4.2 Attacks

There are two types of attack according to the Certified Ethical Hacker guide: passive and active attacks[36]. An active attack is when the attacker attacks the system directly. Passive attacks are commonly used to gather information on the victims system. Passive attacks are used while planning an attack to make the attack more tailored. This could be the gathering of password, network layouts, and network names.

4.3 Vulnerabilities

A vulnerability is defined by NIST as "[w]eakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source."[7, p. B-9]. For an attacker to leverage a vulnerability he needs to use an exploit of that vulnerability. Such an attack might circumvent security protocols and security mechanisms.

4.3.1 Malware

Malware is malicious code, where the code is used to leverage vulnerabilities in the attack phase. Malware is a big category that consists of many other malicious code classes such as virus, worms, trojan horses, and scripts. A virus will use a vulnerability to infect itself to other computers. The delivery could be through the network, email attachments, USB-keys and so on. Viruses can cause damages by causing systems to become unstable. Worms are similar to viruses but focuses on spreading itself, and several worms spread at an alarming speed.

4.3.2 Denial-of-Service

A denial-of-service attack is when a attacker creates a situation where some service or resource is rendered unavailable. This could be through causing a crash in a necessary service or disrupt the link used. Most ICS has a system to handle a situation when resources are unavailable. This might cause the system to seek a safe state, which might be a shutdown state. Even with this system in place, a well timed attack might cause problems such as fires, oil spills, explosions or spoiled batches of products.

4.3.3 Phishing

Phishing, or spear phishing attacks, are where the attacker crafts email messages to trick the reader to click a malicious link or attachment that triggers a malware attack to compromise the computer. Such attacks are crafted using information about the organization and leveraging any information found about the organization to make it appear genuine. This could be anything from using colloquial terms and words that are used in the organization, to citing people in the organization.

4.3.4 Replay attacks

Replay attacks are when packets from the network are modified and resent. In an ICS setting a replay attack is dangerous since such an attack could disable security protocols and systems, and then replay commands which would control the manufacturing in an unsafe fashion. Such attacks require knowledge about the underlying system. Since most protocols were designed without security, such information can be gathered by listening and capturing network traffic. One of the first researchers to show such an attack was Dillon Beresford with his article "Exploiting Siemens Simatic S7 PLCs"[37].

4.3.5 Buffer Overflow

Buffer overflow attacks are when "[a] buffer overflow is the result of stuffing more data into a buffer than it can handle."[38]. This causes the nearby memory locations to be overwritten. When nearby memory is overwritten it might cause the attacker to be able to run code as needed. This is done by adding long arguments with measured out code as to overwrite the Extended Instruction Pointer (EIP), which points to the next instruction, to jump to a location with added malicious code. Buffer overflow attacks are one of the most serious vulnerabilities since it opens up the possibility of running arbitrary code. Buffer overflow is one of the most common vulnerabilities so it has a high likelihood to be exploited.

Chapter 5
Finding vulnerabilities in a ICS

Most ICS protocols were written without security in mind, and it is therefore clear that there are vulnerabilities. The difficult part is finding and fixing them. The common way vulnerabilities become known are through security researchers finding them and disclosing them to the proper vendor, and as can be seen in the figure 1.1 from chapter one, the number of disclosures is rising. This chapter will first look at different tools for searching for vulnerabilities. Then it will look at whitebox and black box testing. However, the major part of this chapter is about Fuzzing and Fuzzing tools, because this is most likely one of the best ways to blackbox test proprietary protocols. To find vulnerabilities, some of the articles chosen has used tools and methods which will be presented in this chapter. This chapter is included to provide a better understanding of the articles which will be presented in the next chapter.

5.1 Tools

There are a number of tools that can be useful when searching for vulnerabilities. One of the first things that needs to be done while searching for potential vulnerabilities, is to identify what services are running on the device in question.

5.1.1 Nmap

The most used of such tool is Nmap. Nmap is a free open source utility for network exploration and security auditing, and can be found at their webpage [39]. Nmap is able to work with the TCP protocol and can scan a wide set of hosts to determine the state of common ports. More intensive scans can reveal unknown open ports or running services. The default scan scans the most used ports, and scans can be configured for specific ports, and for specific hosts or ranges of hosts. Nmap has options for determining the operating system of the host.

5.1.2 Vulnerability scanners

Vulnerability scanners such as Nessus can scan for known vulnerabilities on the network [40]. Nessus has plug-ins for running tests on SCADA networks. There are other modules for simulating potentially harmful Denial of service (DOS) attacks.

5.1.3 Wireshark

One of the most used packet capture and analysing tools, Wireshark described by its own website as: "It lets you capture and interactively browse the traffic running on a computer network. It has a rich and powerful feature set and is world's most popular tool of its kind"[41]. Wireshark is an open-source tool that is freely available trough the Wireshark website.

5.2 Whitebox testing and blackbox testing

Whitebox testing, also known as glass box, structural, clear box and open box testing is a software testing technique where the tester uses his knowledge of the system to do testing for specific inputs and expected outputs. The drawback is if the tester is not sufficiently familiar with the system, in which case a whitebox test might not be equally accurate. Blackbox testing takes the other approach, and the tester may know little or nothing about the system he is testing. When not knowing, the tester might recreate internal workings of the system as to do better testing. If the tester recreates the inner works of the system it is called reverse engineering. For many ICS environments blackbox testing is the only option since most of the inner workings of the system is proprietary technology and thereby, by definition, whitebox testing can not be done.

5.3 Fuzzing

Fuzzing is defined by The Open Web Application Security Project (OWASP) as "a Black Box software testing technique, which basically consists in finding implementation bugs using malformed/semi-malformed data injection in an automated fashion."[42]. A fuzzing test is then to overload the system with data formed in such a way as to create problems. This could be to crash a system by overloading a buffer, which is done by increasing the payload sent to the software for each pass. This will cause a crash if the system is vulnerable. Afterwards, the crash can be analyzed to see what caused it. Another method is to select some data to fuzz on, and mutating it with special characters and syntax to create crashes. The two methods are called mutation-based and generation-based fuzzing. The fuzzer Zulu is a mutation-based fuzzer and it will be discussed later in this chapter.

5.4 Fuzzing Tools

There are good commercial tools and open source tools for fuzzing. As always, getting a open source tool to work is a bit harder and documentation could be scarce. Both open source and commercial tools will be described in the next section. However, due to the price of the majority of the commercial tools, only the free open source products has been tested. The commercial tools section will therefore be based on the creators description, while the open source product is based on both description and on testing.

5.4.1 Commercial Tools

Commercial tools have the benefit of being created for professional use. In the commercial tools the fuzzer seems to be a small part in a big package of tools for finding vulnerabilities in both software and on other attack surfaces. Commercial tools can have a free version for demonstration, but this is not common for security tools for industrial systems. Since it is meant for use in a professional environment, it has a price. This causes the tools to only be used in professional settings, since the cost can be quite high, especially for specialised software solutions.

5.4.2 Achilles

Wurldtech is a company from Vancouver that has developed Achilles. Achilles is a well known brand in the industries and several major companies use them, such as Shell, Siemens, Honeywell and BP according to Wurldtechs website[43]. Wurldtech also offers an Achilles Certification for products, solutions, practices and communication. Achilles has the ability to test several communication channels from low level Internet Control Message Protocol (ICMP) to higher level TCP/UDP, and even Windows or Linux performance. This tool has not been tested because it is an expensive commercial tool intended for industries.

5.4.3 Defensics

Codenomicon is a Finnish company which provides fuzzing solutions. Their solution is called Defensics and provides support for fuzzing on several protocols and services. Defensics lists several available protocols for fuzzing, or black-box testing, as they refer to it. Examples of the protocols listed are Profinet Discovery, Configuration Protocol (DCP) and Precision Time Control Protocol (PTCP) [44].

5.4.4 Open source

Open source tools are freely available for anyone to experiment with. The source code can be modified by whoever downloaded the tools. The downside with open source tools is that they may be poorly maintained and rarely updated. This can cause problems, such as outdated code, not runnable code, or that they are not updated for newer protocols. Although open source can run into those problems, there are also several advantages with them. The main advantage is the cost of getting the tools. Whereas the commercial tools are expensive and created for use by professionals with a financial backing, open source tools are free and can be downloaded instantly. Another upside is that anyone can look at the code and modify it to fit their purposes, but this requires a certain amount of knowledge about programming. Without the necessary skills in programming, can this be seen as a drawback rather than an upside, because the users will not have the same opportunities to properly modify the tools to fit their requirements. However, in active projects there are channels for contacting the creators to ask questions and report bugs, which can improve the quality and the applicability of the open source tools.

5.4.5 Scapy

Scapy is a powerful interactive packet manipulation program. It also allows for making invalid packets, editing packets, and resending them etc. There are several other programs based on Scapy since it is easy to use, but still very powerful. The process of building a packet is by just adding the fields of information that is needed or not needed, and then sending the packet. The possibilities are many for just creating a skeleton packet and varying the fields that are interesting to see changes in. Scape is available from the developers website[45]

5.4.6 Profuzz

Profuzz is based on Scapy and is hosted on github[46]. Profuzz is a command line based fuzzer that has capabilities of fuzzing the following profinet protocols.

1. Alarm Frame Random (AFR)
2. Alarm Frames Ordered (AFO)
3. Profinet IO (PN-IO)
4. Discovery and Configuration Protocol (DCP)
5. Precision Time Control Protocol (PTCP) (only in beta)

Profuzz is no longer updated, and it was last updated two years ago according to its Github page. It was written by Dmitrijs Solovjovs, Tobias Leitenmaier and Daniel Mayer under the supervision of professor Roland Koch. Profuzz was a student project at University of Applied Sciences Augsburg in 2012, which explains why it is no longer updated.

5.4.7 Zulu

The Zulu fuzzer was released at the Nullcon conference in 2014 with the paper explaining the use of Zulu [47]. It is a fuzzer made for Windows only using python. It is a fuzzer with a graphical user interface that has several input methods. The reason it caused interest was the ease of use and its wide capabilities with mutators for fuzzing. Zulu is a mutation based fuzzer and it has several common mutators, and some uncommon ones. Even with all this it is still powerful and simple to use. Some of the features are:

- import a set of captured packages in pcap format
- select the packets that are useful to fuzz
- select the mutators that gives meaning to use.

Zulu is available from github[48] and installing it is a matter of tracking down the dependencies. After testing fuzzers and trying to make one, Zulu was chosen as the fuzzer for the pre-project for this master thesis. Its ease of use and still powerful capabilities makes it an easy choice. The two parts that were a downside to Zulu was the network proxy.

The fuzzer has the possibility to create a proof of concept (POF) based on what packet seemingly caused a crash. On top of that it can send that POF to you by email. The second part was the overview. Since you can only see hex and byte representation of the packets it is difficult to see what the bytes represent. Combining Zulu with Wireshark is essential for choosing the right fields to do fuzzing on. Zulu was chosen over the other tested fuzzer for ease of use, however there were some drawbacks. The first drawback was the need for using another program alongside it to find the fuzzing points, the second was the dependencies that was to be installed before Zulu worked. Although the ease of use outshines the drawbacks, Profuzz is command-line based, and a small error in the command causes it to not work at all.

5.5 Summary

The tools presented and the methods are used by the professionals that are searching for vulnerabilities.[49, p. 78]. The SCADA Strangelove team uses this tools and they are one of the foremost specialists on ICS security. The tools combined with knowledge about security will get results.

Chapter 6 Articles

Computer security is a large research field and there are plenty of articles written about it. The journals *Computers and Security* and *Communication and Information Security* are examples of where this type of research is published. This chapter will go through some of the main articles on ICS security, defence models and SCADA security and challenges. Each of the selected articles will be presented before they are discussed against each other in the ending of the chapter. Moreover, the chapter will provide an overview of which security issues the research community is, and has been, most concerned about. It will also highlight the main security issues, which is where a standard or regulation should potentially have the largest effect.

6.1 How Effective Are the Prevailing Attack-Defense Models for Cybersecurity Anyway?

This article discusses strategies for Cybersecurity. The main focus is a discussion on three strategies for Cybersecurity; the common single point of failure, the Active and Adaptive defence strategies, and lastly the Bastion Model and the Defence-in-Depth Strategy. The article is written by He, Chan, Zhang, Wu and Wang in 2013 and it was published in Intelligent Systems, IEEE, in the Sept.-Oct. 2014 issue [3].

The first method is the common single point of failure where the defence is a single centralized controller. An example of this would be the certification authority (CA), where if a single one is compromised they all are. This is illustrated in 6.1 the parties have to trust the Arbitor. This strategy is not advised, and moving away from it is advised whereas sometimes the organization is not aware that this is the strategy they are deploying. The article gives some tools for checking the security protocols used, they are On-the-Fly Model-Checker (OFMC), Failures-Divergences Refinement (FDR), Simple Promela Interpreter (SPIN), and Prism. The OFMC uses a suite of 36 protocols to identify attack paths. Then the article discusses passive vs active defense methods. Here it is discussed that the standard passive method with vulnerability scanners, access control and so on, functions best when defending known attacks, but does poorly on predicting the next attack. The passive method does not give a picture of the intention of the attacker as to give and predict the plan to be used. While static defence is the same, some strides have been taken in moving target defence, but it also has it share of problems. For example, changing the IP address to defeat denial-of-service (DoS) attacks. While the intention is good, the attacker can just change to use the domain name so as to continuously look up the new IP address.

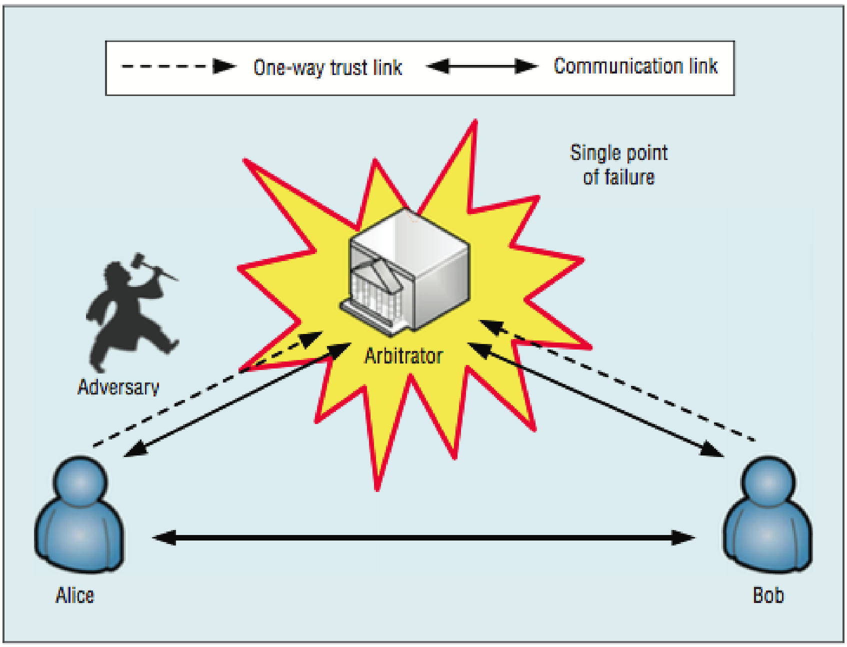


Figure 6.1: Single point of failure [3, p. 15]

The next method presented in the article is the Active and Adaptive defence strategies. An example of this would be intrusion detection and response systems. This sort of system does not aim to prevent attacks, but it aims to mitigate damages and it attempts to locate and / or harm the attacker. This kind of system might have a set of responses based on the attack, or it might just give the administrator feedback that the system is under attack. The sort of response given can be greatly automated, but the automation depends on rules, and it is therefore limited. The automatic response would not be given if the system encounter something which is outside its given rules. Automatic systems would react fast when they are triggered, but, as the article points out, it is the trigger that is the problem. The trigger might respond too late and cause the defense to kick in after the damage is done.

The advice given in the article is: "It's better to make proactive, real-time defense decisions during an earlier stage of an attack."[3]

The dynamic between attacker and defender is such that to make an effective defender that defender need to know the weaknesses of the system but also the intentions of the attacker. This causes the defender to look at the system with the eyes of the attacker and establish security protocols that mitigate the plan of attack he comes up with. The article sums up active defence by comparing it with game theory where the conclusion is to use multiple attack-defense models to lower the probability of a successful attack.

The third and last strategy is the Bastion Model and Defence-in-Depth Strategy. The Bastion model is the most common security model. This type of defence is where you place a firewall between you and the attackers and trust that this will keep the system safe. The problem is that this is a single point of failure. The common next step is the defence-in-depth, where you have multiple layers of security added as you go deeper into the network. This will cause the most critical system to be the most protected. The advantages of this strategy is that the attacker must spend time and resources to gain access in lower value parts of the network, and the attack would lose momentum. Another advantage is that this approach uses more than one security mechanism, thereby eliminating the single point of failure. The drawback of the approach is that it will require the maintainance of more devices and rule sets, and the complexity will go up.

The article points out that the attacker might come from inside the organization. Thus the attack is more serious and harder to predict and prevent. While firewalls are good for blocking things out, they are not so good at defending from the inside. Designing a system that is also able to monitor insiders requires more complexion and hardware. The article concludes that no system can guarantee 100 percent security. What should be the goal is 100 percent risk acceptance.

6.2 SCADA deep inside: protocols and security mechanisms

This presentation is written by the SCADA Strangelove team and goes trough several specialised attack vectors that they have found in their research. The presentation is not published in a security journal, but is available online [50]. It introduces the state of affairs as they see it and their view is very bleak. They point to three problems for this industry.

1. There are many vendors, each with their own technologies and their own protocols.
2. The systems are not updated, the policy is do not touch it if it works.

3. The patch management cycle.

The focus in the presentation is on industrial protocols. The presentation also presents a discussion of the setup of a typical SCADA network and security. From the presenters experience, the setup usually use default setups, or weak setups, and most of the workstations are full of malware. If there is a wireless setup it is at best protected by Wired Equivalent Privacy (WEP). This means that there is no protection at all, since the WEP protocol has been broken for a long time. The presentation also point to weak physical security as another problem. Further, the presentation argues that industrial protocols are not as contained as they should be. They run on several layers of networks which are too widely accessible and also easy to detect. Additionally, some of the network traffic is easy to intercept and reproduce. In this presentation there was a demonstration of a vulnerability CVE-2014-2252[51] where the advisory says "An attacker could cause the device to go into defect mode if specially crafted PROFINET packets are sent to the device. A cold restart is required to recover the system"[50, p. 23]. The specially crafted packets are then showed to be a basic set packet in the Profinet Discovery and Configuration Protocol, where the fields for IP adress, network mask, and gateway all are set to 0.0.0.0 .

6.3 Exploiting Siemens Simatic S7 PLCs

This article was written by Dillon Beresford and presented in July at Blackhat in 2011[37]. The article is written on the topic the Siemens S7 PLC from a security perspective. It presented the method used for analysing the protocol and showed that ISO-TSAP packets were sent in plain text, thus making the process of editing and resending easy. It is noteworthy that this article points out that "[f]or example, if a motor on a centrifuge was configured to rotate at a specific number of revolutions per minute (RPM), we could either change that rate (potentially causing damage to the centrifuge) or stop it altogether (potentially damaging other equipment in the plant)"[37]. This is most likely what happened with Stuxnet. The article also added an addition to Metasploit for easy use. The article moves on to list six attacks that were presented at Black Hat 2011

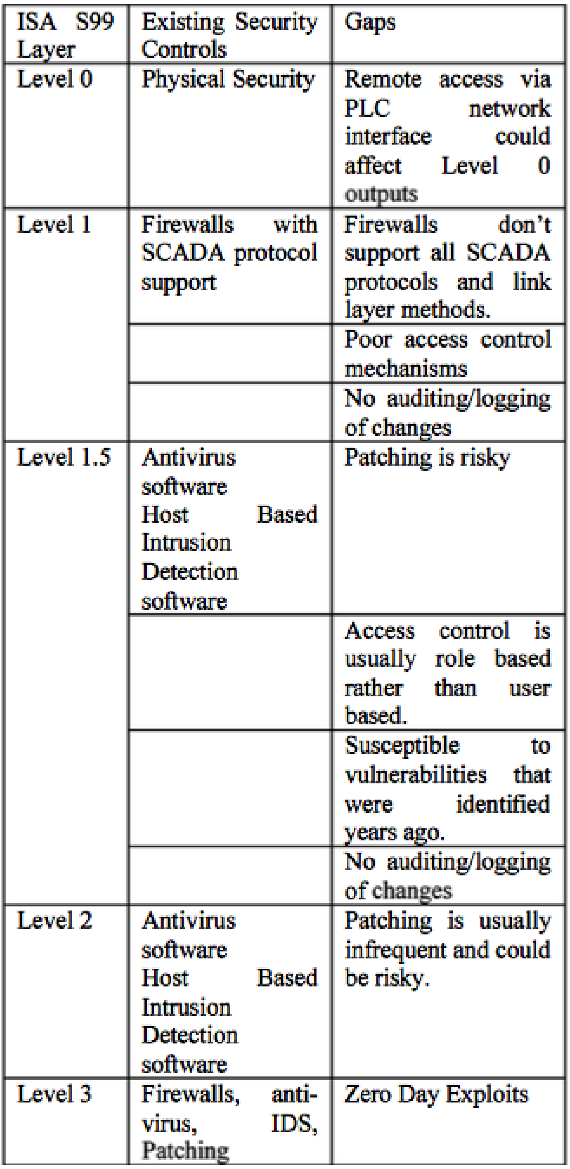
1. TCP Replay over ISO-TSAP attack
2. S7 Authentication Bypass
3. CPU Stop/Start Attack
4. Memory Read/Write Logic Attack
5. Decrypting Siemens Simatic firmware
6. Getting a Shell on a PLC

The first attack only has five steps with the added Metasploit module. First capture ISO-TSAP traffic, then export the TCP stream, dissect and discard unneeded packets, paste to the module, and finally exploit. The example given is an attack which can open closed coils or close open coils The authentication bypass is by building authenticated packets, and since the authentication sessions never expired this leads to a situation where if one session is captured it can be reused. Through probing the PLC information leakage is discovered, which is used to fingerprint the PLC, but also leakage of the logic on the PLC, such as tag names, data-block names, firmware version, services, and PLC name. This is information that is very relevant for the attacker if the plan is to rewrite logic and cause damage. The example given is that an update rewriting LED1 to LED2 and vice versa could cause the operator to not notice errors in the system. The article moves on to rewriting LED with ON_BTN that could result in disaster when the corresponding buttons do not function as they should. Lastly, the article points out that Siemens PLC runs x86 Linux as an operating system and everything is running as root. An attacker gaining a remote shell on a PLC will therefore have complete access.

The articles moves on to demonstrate the disclosure process and the handling by the vendor. Parts of the disclosure process reads as "he said, she said", but with some underlying critique of the vendor. The article concludes that ICS needs new protocols designed for the world of cyber attacks. ICS security might have an impact on a lot of people since a successfull attack would cause damages in the real world.

6.4 Survey of SCADA Security Challenges and Potential Attack Vectors

This article is written by Robert E. Johnson, III and published at Internet Technology and Secured Transactions (ICITST), 2010 International Conference by the IEEE[4]. This survey shows what kind of attacks are used against ICS. It then proceeds to discuss two attacks. The first attack is also discussed in [37] for a specific PLC. This article claims that the attack is possible on other PLC since they are mostly built the same. The attack is to bypass the logic and thereby giving the ability to write to memory. This is possible since the SCADA does not authenticate or authorize requests. The second attack is called "Brute "Force Output" Attack". It uses the concept of "forcing output" that SCADA vendors have added to do troubleshooting. Since there is no authentication this gives an attacker the option to remotely start or stop a process. This attack is both effective and easy since it does not require the attacker to understand the logic in the PLC.



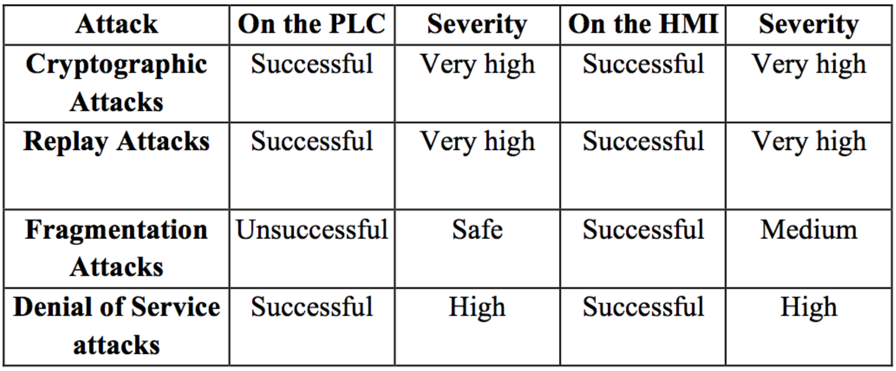
ISA S99 Layer	Existing Security Controls	Gaps
Level 0	Physical Security	Remote access via PLC network interface could affect Level 0 outputs
Level 1	Firewalls with SCADA protocol support	Firewalls don't support all SCADA protocols and link layer methods.
		Poor access control mechanisms
		No auditing/logging of changes
Level 1.5	Antivirus software Host Based Intrusion Detection software	Patching is risky
		Access control is usually role based rather than user based.
		Susceptible to vulnerabilities that were identified years ago.
		No auditing/logging of changes
Level 2	Antivirus software Host Based Intrusion Detection software	Patching is usually infrequent and could be risky.
Level 3	Firewalls, anti-virus, IDS, Patching	Zero Day Exploits

Figure 6.2: Table of vulnerabilities [4, p. 4]

6.5 Internal security attacks on SCADA systems

This article[5] introduces internal attacks on the PLC and the HMI. It is written by Naoum Sayegh Ali Chehab Imad H. Elhajj Ayman Kayssi in 2013, and was published in the conference Communications and Information Technology (ICCIT), 2013 Third International Conference on, in June 19-21 2013.

The authors used four types of attack on both the PLC and the HMI. These four types of attack were cryptographic attacks, replay attacks, fragmentation attacks, and denial of service attacks. The only attack that failed was the fragmentation attack on the PLC. It consisted of sending malformed IP packets to cause problems. They concluded that the PLC discarded all malformed packets. The article sums up the attacks with a table found in fig 6.3



Attack	On the PLC	Severity	On the HMI	Severity
Cryptographic Attacks	Successful	Very high	Successful	Very high
Replay Attacks	Successful	Very high	Successful	Very high
Fragmentation Attacks	Unsuccessful	Safe	Successful	Medium
Denial of Service attacks	Successful	High	Successful	High

Figure 6.3: Result table for internal attacks[5, p. 26]

6.6 A Survey of SCADA and Critical Infrastructure Incidents

This survey[6] uses the taxonomy which was created by Kjærland[52] to better understand and classify ICS security incidences. The survey is written by Bill Miller and Dale C. Rowe Ph.D and published in RIIT '12 Proceedings of the 1st Annual conference on Research in information technology in 2012. The survey found that the attackers were mostly disrupting operations and disclosing information. Most of the attacks were root compromise attacks with trojans, misuse of resources, and user compromise. This means that few of the attacks were with worms and viruses, and only one attack was a denial of service attack. One of the limiting factors of this report is that they have only analysed fifteen attacks. The taxonomy can be found in fig 6.4.

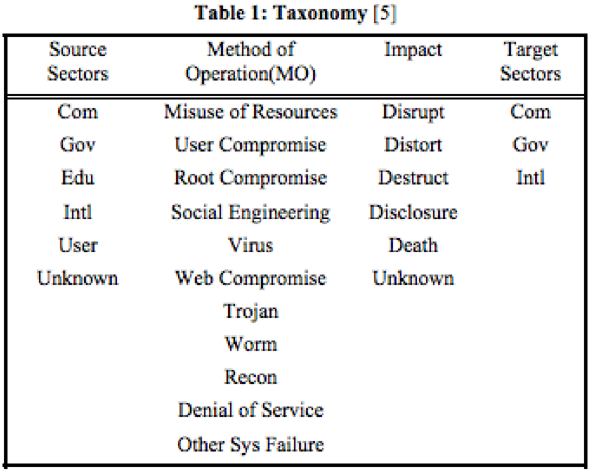


Table 1: Taxonomy [5]

Source Sectors	Method of Operation(MO)	Impact	Target Sectors
Com	Misuse of Resources	Disrupt	Com
Gov	User Compromise	Distort	Gov
Edu	Root Compromise	Destruct	Intl
Intl	Social Engineering	Disclosure	
User	Virus	Death	
Unknown	Web Compromise	Unknown	
	Trojan		
	Worm		
	Recon		
	Denial of Service		
	Other Sys Failure		

Figure 6.4: Taxonomy for classifying attacks [6, p. 51]

6.7 Security issues in SCADA networks

This article[53], written by Vinay M. Igure, Sean A. Laughter, and Ronald D. Williams, was published in Computers & Security Volume 25, Issue 7, October 2006. The article sheds light on security in an SCADA system before Stuxnet was carried out. The article discusses six areas of SCADA security and this thesis will cover them briefly.

The first area the article discuss is access control, where the problem is defining the perimeter of the SCADA network. The second problem with access control is that in a SCADA setting it is based on passwords that are passed in clear text. The second area is firewalls and intrusion detection systems. However, the points made here are no longer of concern. The third area is protocol vulnerability assessment, where the point is to understand what vulnerabilities lies in the current protocols. When the vulnerabilities are understood in context of the protocol there is room for developing security mechanisms that handle them. The fourth area is cryptography and key management, where neither of these are present in SCADA protocols. Since SCADA protocols do not support cryptography there have been efforts into retrofitting the protocol with cryptography. Cryptography is meaningless without key management, which is problematic in SCADA due to the way it is implemented. The fifth area is Device and Operating System security where the content is outdated. The last area is Security management where it contains issues handled by newer security standards such as ISO 27002

6.8 The articles and their issues

As the articles presented have shown, there are several issues in ISC security. Each one of the articles shows problems which needs to be addressed, but most of the articles just describe the problems in ICS. They do not offer any solutions or feedback on how to fix the issues they raise. This is illustrated clearly in the ending of SCADA Strangelove, an independent group of security researchers, who ask "What will happen if you send another packet, another value?". The answer they give is "Yes, you´re right" and then they follow up with a picture of a nuclear mushroom [50, p. 80-81].

The article "Security issues in SCADA networks" is written and published earlier than the rest of the articles reviewed in this chapter. It is thus striking how several of the problems and issues which was raised and discussed in the article from 2006 are problems that ISC security research still deals with, nine years later. One example of this is the issue with the need for more secure protocols, which was raised in the article from 2006. This issue is raised again in the article "Exploiting Siemens Simatic S7 PLCs" when they argue that "[w]e need secure protocols in ICS. The product vendors have the ability to make this a reality."[37, p. 23]. Although the issue was raised already in 2006, it does not seem to have inspired to the creation of any functioning solution, and the article from 2011 thus saw it as necessary to raise the issue yet again. The recurrence of the same issues over a longer time frame shows the importance of evaluating the potential solutions which have been created since those issues first were raised.

The presentation from the SCADA Strangelove team claims that there is no security on ICS protocols. Support for their argument can be found in the article "Internal security attacks on SCADA systems". The internal attack proved that several trivial attacks were effective against both the PLC and the HMI station. Most of these attacks, however, needs physical access to the ICS network, a network that should not be available from the Internet. This means that most of the attacks which are mentioned in the "Internal Security attacks on SCADA Systems" will only be possible if the attacker is already on the ICS network. However, this is not a necessity for all the types of attacks, and it is therefore essential to have security measures prepared against those attacks, as well as those which could happen in cases where the network is connected to the Internet. The SCADA Strangelove presentation states that in their work the network was almost always available from the Internet. The survey on ICS attacks would have been very interesting if it had analysed more incidents, and included a metric for internal attacks and external attacks. Attacking a ICS network should be harder. Deploying the Defence in depth strategy and enforcing ICS networks not to be exposed to the Internet, should be a minimum. Defence in depth would, as discussed in "How Effective Are the Prevailing Attack-Defense Models for Cybersecurity Anyway?"[3], add another layer of security and move away from the single point of failure strategy. Additionally, adding defence in depth would make it more problematic for a single dishonest insider or a single technical error to cause a harmful result.

Chapter 7
Outcome

The previous sections have shown the wide variety in standard and regulations. It has also shown types of attack, former incidents and the research which is currently being done on the field. This chapter will combine the two. It will first discuss security standards against the attacks discussed earlier and against the articles discussed in chapter six. The second part of this chapter will propose a table for categorizing attacks against which standard would be effective against it. Two example tables are created based on the previously discussed attacks.

7.1 Standards as a solution?

As we have seen, there are several standards on the ICS field. Some of the standards have a best practice approach and provide guidelines (Example: ISO/IEC, NIST), while others are based on regulation and must be followed (Example: Nirc cip). What they all have in common is a shared aim to raise the level of security. ICS specific standards have a higher focus on technical aspects of the security, while more generic standards such as ISO 27002 "focus more on management and organizational issues, and less on technical issues"[14, p. 6]. The focus on the technical side might give a better protection against technical attacks than a management and organizational standard might achieve. This is something "A Survey of SCADA and Critical Infrastructure Incidents" indicated as well. However, the survey also indicated that one of the most referenced technical attacks, denial of service attacks, are among one of the least common attack types. Management and organizational standards might therefore be more useful against some types of attacks, such as spear phishing attacks, because a management and organization standard would possible establish routines against those types of attacks. The difference between the two types of standards means that implementing both standards could potentially be better for achieving a high level of security. However, not enough research is done on what kind of attacks the different standards and regulations would protect against. The "Survey on SCADA and Critical Infrastructure Incidents"[6] is a step towards more research on the area, but it says nothing about which security standards were implemented, and the survey is limited in the selection of incidents which are analyzed.

A common theme in the standards and regulations for security is the requirement that patching should be done as soon as it is available. In general patching system, when patches are available, would this arguable be the best solution. The situation are not the same in ICS patching, because this might cause a risk to the availability of the system, which could cause downtime. This could be a risk for the companies involved because downtime could mean lost money and reputation. This risk might explain why patching on a regular schedule is not always done, or why no patches have been applied in a long time.

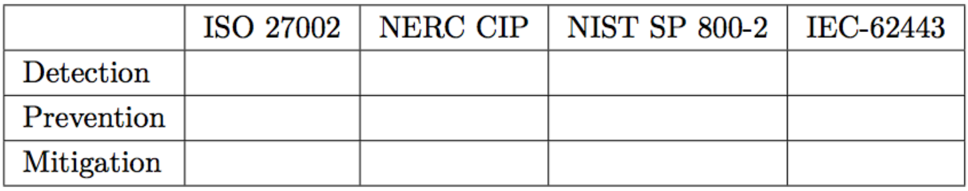
The best practice approach from the ISO 27002 and NIST 800-82 will be better in creating a security culture in the security program. A good security culture where the users and operators of the system is aware of the security risks. The ISO standard also includes a auditing procedure as to keep up the standard and to monitor the security culture. If spear phishing was talked about in the workplace this might cause fewer successful phishing attacks. "Prior exposure to phishing education is associated with less susceptibility to phishing, suggesting that phishing education may be an effective tool. Also, more risk-averse participants tended to fall for fewer phish."[54, p. 380-381]

How well a standard will work depends on the environment it is installed in. The humans who work with the networks, computers, plants, or wherever else the standard is implemented, will be able to impact the usefulness of the standard [11]. The leadership must be engaged in the security to make sure it is updated, and to make sure that the employees follow the requirements set by the standards. Training of the employees so that they are updated and aware of the security issues is also important. The specifics of a standard can cover a wide range of issues, but without participation from knowledgeable employees will the standard, and in particular the management and organization standards, not be able to contribute much towards an increased security level.

As this section have shown, there are several issues with standards and several places where a standard will not be able to provided the wanted protection. It is therefore probably more precise to consider the standards as a part of the solution, rather than the solution in itself. Attacks like Stuxnet, which exploits unknown vulnerabilities, will always cause some damage, but a security standard might have an reducing effect on the damage done. What the exact contributions of the different standards are, needs more research. An example is the Denial of Service attack. This is, as we have seen, a lot less common than what the amount of literature on it would suggest. This is something which the standards protection could have contributed towards, but other reasons such as it simply being a less common attack than previously assumed might also be the reason. By conducting more research on standards, their implementation and their effect, will standards potentially be able to contribute towards increased ICS security in a larger degree than they are today, but the amount of time and money spent on creating standards shows that they do contribute today as well.

7.2 Proposed Approach

The amount of work, time and money that goes into creating standards and regulations to improve security in relations to ICS shows that this is something which developers and users believe is useful. It is, however, hard to see exactly what the standards and regulations do, and in particular how they compare against each other in different security questions. This problem is one of the main findings in this thesis. There is currently no easy way to compare attacks, viruses, vulnerabilities or other (new) weaknesses against the current standards and regulations. One solution to those problems would be to develop a table, such as the one seen in figure 7.1. Once the table is established, can this be used in reports of an attack. The table could also be filled out with colors, so as to easy indicate which standard and regulation would have been the best solution. The suggestion is to color code the table, where a green square indicates that it is a positive, a yellow is a partial and red is a negative, with regards to the corresponding line.



	ISO 27002	NERC CIP	NIST SP 800-2	IEC-62443
Detection				
Prevention				
Mitigation				

Figure 7.1: Suggested table for proposed approach

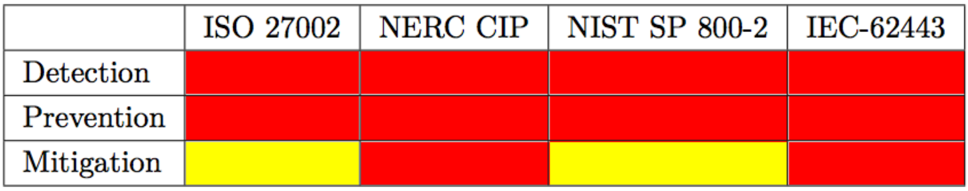
The main metrics which is proposed to look at in such a table is "Detection", "Prevention" and "Mitigation". Detection, whether or not the attack would have been discovered if the standard had been implemented, is included because this is one of the most important aspects in a ICS security context. The attack can only be stopped or prevented if awareness of it exists. Detection is therefore also included on as the top matrix in the table. Prevention, the second metrics in this table, is also important, but perhaps not as important as detection because you can not prevent an attack you which have not been detected. Prevention is in particular important for future research because it looks at if implementation of a regulation or a standard would have prevented the attack. If a standard or a regulation get a high percentage of not-preventive (red color), the standard might need to be improved before it is implemented. The last matrix, mitigation, looks at if the standards would actually have created a difference. This matrix would show the general usefulness of the implementation of the standard or the regulation against the attacks the table it is discussed up against. Although this table only includes three metrics, others could be added as well to improve the aspects on the research. Which others to add would depend on what the table users want to get out of the table, and it can be individually tailored to fit certain types of research or questions. The three which is included in the table presented here should, however, always be included because they answer the fundamental questions for increased security.

Although there are several advantages with using a simpler system, such as the table presented here, to gain an overview of how standards and regulations would have impacted attacks, does this approach also have some drawbacks. The main drawback is the requirement this approach sets to knowledge. Understanding and knowledge about the different standards and regulations, what they aim to do and which requirements they set, is a necessity before such a table can be created. The creation of such a table also requires knowledge about the attack. This type of research often takes a while t be conducted, and it will therefore often be a while before the table can be created. An example of this is the Duqu 2.0, which was just discovered. The knowledge which has been generated about this attack so far would probably be insufficient to create a proper table. A preliminary table which would give some insights could, however, probably be created

This table would make it easier to do quantitative research or comparative research on how standards and regulation would have impacted the attack. This would also make it easy for the business or industry to decide on which standards would be the most beneficial for them, because they can easily see which standards and regulations would offer the better protection against the attacks they are most concerned about. In the long run could such an approach to the standards and regulations approach contributed to improved standards or regulations, because their weakness will be easy to discover, and thereby easier to improve. The most important contribution of the table is that it gives a starting point for needed further research on the impact of standards on ICS security.

7.2.1 Example 1: Stuxnet

Stuxnet, which was described in chapter 3.1, was a worm which greatly impacted ICS security research. In this example is four of the standards and regulations impact on this standard presented in the table. As can be seen from the Figure 7.2 would not a implementation of any of the standards have detected the attack. The reason for this is because the attack used 0-day vulnerability. This is therefore color coded in red. Even if the attack had been detected, would none of the standards have helped to prevent it, because of the complexity of the attack. The third category would be compared to other sites where the same attack technique was used, and if the same level of damage was achieved both get a red. If some damages were mitigated then that should be a yellow, e.g. partial mitigation. The yellow section in the mitigation metrics of the table shows that more matrices could possible have been added to improve the understanding of the attack and how the standards effected it. As mentioned before, this table is by no means a finished solution, but it could be seen as a start for trying to set standards against attacks. Already can the severity of this attack be seen in how large sections of the table was color coded red. Possible could no standard or regulation have helped to prevent this types of attacks, which is one of the main reasons for why standards and regulations should be considered as a tool which will contribute, rather than as a stand-alone solution.

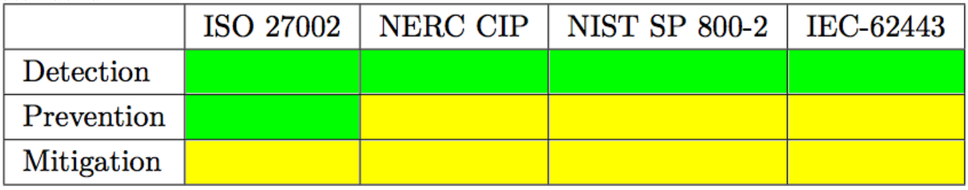


	ISO 27002	NERC CIP	NIST SP 800-2	IEC-62443
Detection	🟥	🟥	🟥	🟥
Prevention	🟥	🟥	🟥	🟥
Mitigation	🟨	🟥	🟨	🟥

Figure 7.2: Suggestion applied on Stuxnet

7.2.2 Example 2: Zotob

The color coding in example 2, Zotob shows striking differences from the table in example one. All the four standards in the table would have helped to detect the attack, and the ISO/IEC 27002 standard would also prevent it. Since the attack was based of a Microsoft security advisory and as such there were patches available. The other three standards would have partly helped to prevent it. As in Example 1, does Mitigation varied from the result found in the other columns, which again indicates that adding more matrices would have been useful. Based on the two tables created for the examples here, is it clear that standards can contribute and the table shows easily where they will provide an increased ICS security and where they can not, thus making it easier for companies, users or industries to select the appropriate standard.



	ISO 27002	NERC CIP	NIST SP 800-2	IEC-62443
Detection	🟩	🟩	🟩	🟩
Prevention	🟩	🟨	🟨	🟨
Mitigation	🟨	🟨	🟨	🟨

Figure 7.3: Suggestion applied on Zotob

Chapter 8 Conclusion: Standards and their impact

Security should be seen as a continuous process where it is important to be prepared for future attacks. This is in particular important for ICS security, which is at a higher level due to the potential physical damages and possible outcomes of a successful attack and breach of security. A standard could be a potential tool towards increasing the security, but a standard should not be seen as a single standing solution to the many issues with ISC security which have been highlighted here. A standard in itself also comes with many potential problems, but as this thesis has shown, there are several places where it can contribute. It can help with structuring the organization and management of security, and thus providing security against those types of attacks. Standards can also help against more technical security issues. It is likely that existing standards have contributed to limiting the consequences of attacks, and that they will do the same in the future. Some of the attacks described in this thesis would have been better protected against if one of the mentioned standards had been implemented. It is safe to assume that the field of ISC security today would have been different if it had not been for the standards and that the standards and regulations have played an important role.

This thesis did not find any well developed approach to how to evaluate the standards and regulations against attacks. The thesis also found the same on issues which have been highlighted in research on ICS security. The lack of this makes it harder to evaluate the impact of the standards, and to select the standards which would be the most appropriate for a set situation. The suggested approach in chapter 7.2 is a table which maps attacks against standards. This type of table give an easier overview of what a standard can contribute with, and it is also something which could easy be developed to a more complex system for a more thorough research. The tables would also be easy to compare in larger number, and thus better show an overview of the weakness and strengths of different standards and regulations.

As this thesis have shown there are several types of standards and regulations developed to improve security, but the implication of selecting one over another is not properly known. Several of the standards can, and probably should, be combined to increase the security level, but not enough research has been done on this. This thesis should be seen as a starting point for future research on the impact of standards and regulations, rather than as a solution to the questions posed in the thesis.

8.1 Further works

This thesis have shown that future works on the issues raised would be a useful contribution to knowledge. An interesting point for further research would be to map previous attacks against the standards and regulations that were in place. This would be a continuation of the work done in "A Survey of SCADA and Critical Infrastructure Incidents" but with a broader taxonomy. If such a research was to take place, more incidences should be included as to get a broader understanding of what kind of attacks are effective against which standards. This type of study would be improved through including a check if the standard is implemented correctly as to find out if the fault was in the standard, or in the implementation of the standard. This is the same problem as in Security issues in SCADA networks with regards to protocols: "When analyzing any protocol, it is useful to distinguish between two categories of vulnerabilities: those that are inherent in the protocol specification itself and those that are the result of improper implementation of the protocol"[53, p. 502]. Conducting the further research which has been proposed in this thesis can highlight the impact standards and regulations can have, it can make it easier for firms, industries and users to select the appropriate standards and it can contribute towards better and more efficient standards and regulations in the future.

References

[1] E. D. Knapp and J. T. Langill, *Industrial Network Security: Securing critical infrastructure networks for smart grid, SCADA, and other Industrial Control Systems*. Syngress, 2014.

[2] K. Stouffer, J. Falco, and K. Scarfone, "Guide to industrial control systems (ics) security," *NIST Special Publication*, vol. 800, no. 82, pp. 16–16, 2007.

[3] D. He, S. Chan, and Y. Zhang, "How effective are the prevailing attack-defense models for cyber security anyway?," 2013.

[4] R. E. Johnson, "Survey of scada security challenges and potential attack vectors," in *Internet Technology and Secured Transactions (ICITST), 2010 International Conference for*, pp. 1–5, IEEE, 2010.

[5] N. Sayegh, A. Chehab, I. H. Elhajj, and A. Kayssi, "Internal security attacks on scada systems," in *Communications and Information Technology (ICCIT), 2013 Third International Conference on*, pp. 22–27, IEEE, 2013.

[6] B. Miller and D. Rowe, "A survey scada of and critical infrastructure incidents," in *Proceedings of the 1st Annual conference on Research in information technology*, pp. 51–56, ACM, 2012.

[7] K. Stouffer, J. Falco, and K. Scarfone, "Guide to industrial control systems (ics) security," *NIST special publication*, pp. 800–82, 2011.

[8] B. Drury, *Control Techniques Drives and Controls Handbook*. 2009.

[9] L. Zheng and H. Nakagawa, "Opc (ole for process control) specification and its developments," in *SICE 2002. Proceedings of the 41st SICE Annual Conference*, vol. 2, pp. 917–920, IEEE, 2002.

[10] NERC, "Nerc description." http://www.nerc.com/pa/Stand/Pages/CIPStandards.aspx. accessed 2015-02-05.

[11] C. N. Murphy and J. Yates, *The International Organization for Standardization (ISO): global governance through voluntary consensus*. Routledge, 2009.

[12] ISO, "Iso 27002 description." http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=54533. accessed 2015-02-05.

[13] T. I. S. of Automation, "Isa99 website." https://www.isa.org/isa99/. accessed 2015-03-03.

[14] T. Sommestad, G. N. Ericsson, and J. Nordlander, "Scada system cyber security—a comparison of standards," in *Power and Energy Society General Meeting, 2010 IEEE*, pp. 1–8, IEEE, 2010.

[15] N. Falliere, L. O. Murchu, and E. Chien, "W32. stuxnet dossier," *White paper, Symantec Corp., Security Response*, 2011.

[16] K. Hearn, P. A. Williams, and R. J. Mahncke, "International relations and cyber attacks: Official and unofficial discourse," 2010.

[17] R. McMillan, "Siemens: Stuxnet worm hit industrial systems." http://www.pcworld.com/article/205420/article.html. accessed 2015-11-04.

[18] Siemens, "Siemens recommandation." https://support.industry.siemens.com/cs/document/43876783?lc=en-WW#Recommended_procedure%200408. accessed 2015-11-02.

[19] P. B. David Albright and C. Walrond, "Did stuxnet take out 1,000 centrifuges at the natanz enrichment plant? preliminary assessment," 2010.

[20] L. O. Murchu, "Debate the stuxnet authors are behind the duqu trojan." http://www.scmagazine.com/debate-the-stuxnet-authors-are-behind-the-duqu-trojan/article/217171/. accessed 2015-22-02.

[21] Symantec, "W32.duqu: The precursor to the next stuxnet." http://www.symantec.com/connect/w32_duqu_precursor_next_stuxnet. accessed 2015-01-02.

[22] K. ZETTER, "Attackers stole certificate from foxconn to hack kaspersky with duqu 2.0." http://www.wired.com/2015/06/foxconn-hack-kaspersky-duqu-2/. accessed 2015-18-06.

[23] sKyWIper Analysis Team, *sKyWIper (a.k.a. Flame a.k.a. Flamer):A complex malware for targeted attacks*. 2012.

[24] K. Lab, "Resource 207: Kaspersky lab research proves that stuxnet and flame developers are connected." http://www.kaspersky.com/about/news/virus/2012/Resource_207_Kaspersky_Lab_Research_Proves_that_Stuxnet_and_Flame_Developers_are_Connected. accessed 2015-01-02.

[25] Mcafee, "Global energy cyberattacks:"night dragon"," *White paper, McAfee.*, 2011.

[26] P. Mangan, "W32.zotob.d." http://www.symantec.com/security_response/writeup.jsp?docid=2005-081609-4733-99. accessed 2015-18-02.

[27] J. Stith, “Zotob damages hit $97k, could be worse.” http://archive.securitypronews.com/2005/1027.html. accessed 2015-02-03.

[28] J. Wagstaff, “Reuters article.” http://www.reuters.com/article/2014/04/24/us-cybersecurity-shipping-idUSBREA3M20820140424. accessed 2015-02-02.

[29] U. news, “Newsarticle on gps attack.” http://www.utexas.edu/news/2013/07/29/ut-austin-researchers-successfully-spoof-an-80-million-yacht-at-sea/. accessed 2015-03-02.

[30] U. news, “Youtube movie explaining attack.” https://www.youtube.com/watch?v=ctw9ECgJ8L0. accessed 2015-03-01.

[31] Y. Dyryavyy, “Preparing for cyber battleships – electronic chart display and information system security,” *White paper, NCC*, 2014.

[32] A. P. Marco Balduzzi, Kyle Wihoit, “Hey captain, where’s your ship? attacking vessel tracking systems for fun and profit,” *White paper, NCC*, 2013.

[33] F. Katsilieris, P. Braca, and S. Coraluppi, “Detection of malicious ais position spoofing by exploiting radar information,” in *Information Fusion (FUSION), 2013 16th International Conference on*, pp. 1196–1203, IEEE, 2013.

[34] L. W. (Director).

[35] O. security, “Exploit database.” https://www.exploit-db.com. accessed 2015-01-02.

[36] R. Shirey, “Rfc 2828: Internet security glossary,” *The Internet Society*, p. 13, 2000.

[37] D. Beresford, “Exploiting siemens simatic s7 plcs,” *Black Hat USA*, 2011.

[38] A. One, “Smashing the stack for fun and profit,” *Phrack magazine*, vol. 7, no. 49, pp. 14–16, 1996.

[39] G. Lyon, “Nmap website.” http://nmap.org/. accessed 2015-02-02.

[40] T. N. Security, “Nessus website.” http://www.tenable.com/products/nessus-vulnerability-scanner. accessed 2015-02-02.

[41] Wireshark, “Wireshark website.” https://www.wireshark.org/faq.html#q1.1. accessed 2015-01-03.

[42] OWASP, “Owasp fuzzing.” https://www.owasp.org/index.php/Fuzzing. accessed 2015-01-02.

[43] Wurldtech, “Wurldtech website.” http://www.wurldtech.com/. accessed 2015-01-07.

[44] Codenomicon, “Codenomicon website.” http://www.codenomicon.com/. accessed 2015-01-12.

[45] Secdev, “Secdev website wtih scapy.” http://www.secdev.org/projects/scapy/. accessed 2015-02-14.

[46] D. M. Dmitrijs Solovjovs, Tobias Leitenmaier, “Github hosting profuzz.” https://github.com/HSASec/ProFuzz. accessed 2015-03-03.

[47] A. Davis, “Fuzzing the easy way, using zulu,” *White paper, NCC*, 2014.

[48] A. Davis, “Github hosting zulu.” https://github.com/nccgroup/Zulu. accessed -09-08.

[49] A. Timorin, “Scada deep inside: protocols and security mechanisms.” http://www.slideshare.net/AlexanderTimorin/scada-deep-inside-protocols-and-security-mechanisms-40672525. october edition of the presentation discussed earlier, accessed 2015-02-03.

[50] A. Timorin, “Scada deep inside: protocols and security mechanisms,” *Presentation, Scadastrangelove team*, 2014.

[51] N. V. Database, “Website for cve-2014-2252.” https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-2252. accessed 2015-04-04.

[52] M. Kjaerland, “A taxonomy and comparison of computer security incidents from the commercial and government sectors,” *Computers & Security*, vol. 25, no. 7, pp. 522–538, 2006.

[53] V. M. Igure, S. A. Laughter, and R. D. Williams, “Security issues in scada networks,” *Computers & Security*, vol. 25, no. 7, pp. 498–506, 2006.

[54] S. Sheng, M. Holbrook, P. Kumaraguru, L. F. Cranor, and J. Downs, “Who falls for phish?: a demographic analysis of phishing susceptibility and effectiveness of interventions,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 373–382, ACM, 2010.