KTH Stockholm and Aalto University
Double Degree Programme NordSecMob

Arthur Gervais

Security Analysis of Industrial Control Systems

Master's Thesis
Espoo, June 29, 2012

Supervisors: Professor Tuomas Aura, Aalto University
Professor Peter Sjödin, KTH Stockholm
Instructor: Michael Przybilski, Nixu Oy

Aalto University
School of Science

KTH Stockholm and Aalto University
Double Degree Programme NordSecMob

ABSTRACT OF
MASTER'S THESIS

Author:	Arthur Gervais		
Title:	Security Analysis of Industrial Control Systems		
Date:	June 29, 2012	**Pages:**	90
Professorship:	NordSecMob	**Code:**	T-110
Supervisors:	Professor Tuomas Aura, Aalto University Professor Peter Sjödin, KTH Stockholm		
Instructor:	Michael Przybilski, Nixu Oy		

Industrial Control Systems (ICS) and Supervisory Control And Data Acquisition (SCADA), have lately gained the attention of IT security researchers as critical components of modern industrial infrastructure. One main reason for this attention is that ICS have not been built with security in mind and are thus particularly vulnerable when they are connected to computer networks and the Internet. ICS consists of SCADA, Programmable Logic Controller (PLC), Human-Machine Interfaces (HMI), sensors, and actuators such as motors. These components are connected to each other over fieldbus or IP-based protocols.

In this thesis, we have developed methods and tools for assessing the security of ICSs. By applying the STRIDE threat modeling methodology, we have conducted a high level threat analysis of ICSs. Based on the threat analysis, we created security analysis guidelines for Industrial Control System devices. These guidelines can be applied to many ICS devices and are mostly vendor independent. Moreover, we have integrated support for Modbus/TCP in the Scapy packet manipulation library, which can be used for robustness testing of ICS software.

In a case study, we applied our security-assessment methodology to a detailed security analysis of a demonstration ICS, consisting of current products. As a result of the analysis, we discovered several security weaknesses. Most of the discovered vulnerabilities were common IT security problems, such as web-application and software-update issues, but some are specific to ICS. For example, we show how the data visualized by the Human-Machine Interface can be altered and modified without limit. Furthermore, sensor data, such as temperature values, can be spoofed within the PLC. Moreover, we show that input validation is critical for security also in the ICS world. Thus, we disclose several security vulnerabilities in production devices. However, in the interest of responsible disclosure of security flaws, the most severe security flaws found are not detailed in the thesis.

Our analysis guidelines and the case study provide a basis for conducting vulnerability assessment on further ICS devices and entire systems. In addition, we briefly describe existing solutions for securing ICSs.

Keywords:	EtherCAT, ICS, Industrial Control System, Industriell säkerhet, IT, IT säkerhet, Modbus, Modbus/TCP, PLC, Programmable Logic Controller, Scapy, Säkerhet, Hotbildsanalys
Language:	English

Aalto University
School of Science

KTH Stockholm and Aalto University
Double Degree Programme NordSecMob

SAMMANDRAG AV
DIPLOMARBETET

Utfört av:	Arthur Gervais		
Arbetets namn:	Security Analysis of Industrial Control Systems		
Datum:	Den 29 Juni 2012	**Sidantal:**	90
Professur:	NordSecMob	**Kod:**	T-110
Övervakare:	Professor Tuomas Aura, Aalto University Professor Peter Sjödin, KTH Stockholm		
Handledare:	Michael Przybilski, Nixu Oy		

Industrial Control Systems (ICS) och Supervisory Control And Data Acquisition (SCADA) har nyligen fått uppmärksamhet av IT-säkerhetsforskare som viktiga komponenter i modern industriell infrastruktur. En viktig orsak till denna uppmärksamhet är att säkerhet inte beaktats då ICS har byggts och dylika system är därför särskilt utsatta när de ansluts till datanät och Internet. ICS består av SCADA, Programmable Logic Controller (PLC), Human-Machine Interfaces (HMI), sensorer och aktuatorer, t.ex. motorer. Dessa komponenter är anslutna till varandra via fältbuss eller IP-baserade protokoll.

I detta examensarbete har vi utvecklat metoder och verktyg för att bedöma säkerheten i industrikontrollsystem. Genom att tillämpa STRIDE- hotmodelleringsmetoden, har vi genomfört en högnivå analys av hoten mot ICS:er. På basen av hotbildsanalysen skapade vi riktlinjer för säkerhetsanalys av ICS-enheter. Dessa riktlinjer kan tillämpas på många ICS-enheter och är mestadels oberoende av leverantör. Dessutom har vi integrerat stöd för Modbus/TCP i Scapy-paketmanipulationsbiblioteket, vilket kan användas för robusthetstestning av ICS-programvara.

I en fallstudie har vi använt vår säkerhetsbedömningsmetodologi för en detaljerad säkerhetsanalys av en demonstrations-ICS, som består av aktuella produkter. Som ett resultat av analysen upptäckte vi flera säkerhetsrisker. De flesta av sårbarheterna vi upptäckte var vanliga IT-säkerhetsproblem, t.ex. problem med webbtillämpningar och uppdateringar av program, men några är specifika för ICS. Vi visar t.ex. hur data som visualiseras genom Human-Machine Interface kan ändras och modifieras hur som helst. Dessutom kan sensordata, t.ex. temperaturvärden, förfalskas inom PLC. Dessutom visar vi att validering av indata är avgörande för säkerheten också i ICS-världen. Således beskriver vi flera säkerhetsproblem i produktionsutrustning. För att ansvarsfullt meddela om säkerhetsfel har vi dock inte beskrivit i detta examensarbete de värsta felen vi hittade.

Våra analysriktlinjer och fallstudien ger underlag för att genomföra sårbarhetsanalys på ytterligare ICS-enheter och hela system. Dessutom beskriver vi kortfattat befintliga lösningar för att säkra ICS:er.

Nyckelord:	EtherCAT, ICS, Industrial Control System, Industrial security, IT, IT compared to ICS, IT security, Modbus, Modbus/TCP, PLC, Programmable Logic Controller, Scapy, Security, Threat analysis
Språk:	Engelska

Acknowledgements

I would like to thank Nixu Oy and the colleagues (especially Lauri Vuornos, Juhani Mäkelä and Michael Przybilski) for making it possible to conduct my thesis on Industrial Control Systems. The industrial environment enabled us to take advantage of the research and to apply it to practical projects. Moreover, without the help and involvement of Schneider Electric such an applied analysis would not have been possible.

Furthermore, I would like to thank Tuomas Aura, Peter Sjödin and Youakim Badr for their valuable feedback for my thesis.

Finally, my dedicated thanks goes to my girlfriend and parents who supported and helped me throughout the Thesis work with hints, valuable advices and good care.

Espoo, June 29, 2012

Arthur Gervais

Abbreviations and Acronyms

AC drive	Alternating Current drives (an AC drive in a variable-frequency drive)
ADU	Application Data Unit
AES	Advanced Encryption Standard
BED	Bruteforce Exploit Detector
CERT	Computer Emergency Response Team
CIA	Confidentiality, Integrity, Availability
CIP	Common Industrial Protocol
COTS	Commercial off-the-shelf
CVE	Common Vulnerability Entries
DCS	Distributed Control System
DoS	Denial of Service
DDoS	Distributed Denial of Service
DFD	Data Flow Diagram
DHS	Department of Homeland Security
DNP3	Distributed Network Protocol 3
Ethernet IP	Ethernet Industrial Protocol
ENISA	European Network and Information Security Agency
FBD	Function block diagram
FPGA	Field-Programmable Gate Array
FTP	File Transfer Protocol
GSM	Global System for Mobile Communications
HI	Host identifier
HIP	Host Identity Protocol
HMI	Human-Machine Interface
HTTP	Hypertext Transfer Protocol
ICS	Industrial Control System
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
ISA	International Society of Automation

ISO	International Standardization Organization
IT	Information Technologies
JTAG	Joint Test Action Group
LAN	Local Area Network
LD	Ladder diagram
LSB	Least-Significant Bit
COM	Component Object Model (Microsoft)
MSB	Most-Significant Bit
NERC	North American Electric Reliability Corporation
NIST	National Institute of Standards and Technology
OLE	Object Linking and Embedding
OPC	OLE for Process Control
OPC A&E	OPC Alarms and Events
OPC DA	OPC Data Access
OPC HDA	OPC Historical Data Access
OPC UA	OPC Unified Architecture
OWASP	Open Web Application Security Project
PC	Personal computer
PDU	Protocol Data Unit
PLC	Programmable Logic Controller
PnP	Plug-and-Play
RPM	Rounds Per Minute
RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition
SD card	Secure Digital card
SIM	Subscriber Identity Module
SMA	Secure Mobile Architecture
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
SRA	Safety, Reliability, Availability
ST	Structured Text
STRIDE	Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service and Elevation of privileges (used within the Microsoft threat model)
TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security
URL	Universal Resource Locator
USB	Universal Serial Bus
VPN	Virtual Private Network
WAN	Wide Area Network
WPA	Wi-Fi Protected Access

WSS	Web Service Security
Zero-day vulnerability	Unknown vulnerability, not fixed by the affected vendor

Contents

Chapter 1

Introduction

Industry and citizens are increasingly relying on machines and automated processes. Hereby, *Industrial Control Systems* (ICSs), are a fundamental key technology for reliably executing a given action and different tasks. ICSs heat buildings, automate production processes, carry electricity to households and accomplish many other operations.

In order to operate reliably, Industrial Control Systems have been designed to maximize reliability and safety [16] [5]. The fundamental difference between safety and security, is that security assumes a malicious attacker [6] while safety does not. Once a malicious attacker is able to compromise the security of an ICS, its safety cannot be guaranteed either. Furthermore, because of their wide use, the failure of ICSs could significantly harm our society.

Consequently, ICS vendors have a considerable responsibility to manufacture ICS devices which operate properly and withstand possible malicious attacks. The present thesis will evaluate the security of Industrial Control Systems as a whole by analyzing first the general ICS security situation and then specific ICS devices.

1.1 Problem statement and methodology

This present Master's thesis aims to investigating ICSs from an IT security standpoint. Therefore, real devices from an ICS manufacturer are analyzed. Besides evaluating different specific threats to Industrial Control Systems,

the thesis evaluates the ability of customized proof-of-concept security testing tools. Finally, the analysis of a given device creates guidelines for analyzing further devices.

In concrete terms, the goals of this thesis are:

1. Understanding how ICSs are built. Furthermore, identifying the differences and priorities between ICS and Information Technology (IT) with particular focus on security.
2. Identifying threats regarding ICSs. Conducting a theoretical threat analysis.
3. Implementing a test library which can serve as robustness tester for the industrial control protocol Modbus/TCP.
4. Presenting guidelines on how to analyze ICS hardware from an IT security perspective. Applying the guidelines on concrete industrial control hardware devices.
5. Giving an overview of existing security mechanisms between the *Programmable Logic Controller* (PLC) and the controlling *Human-Machine Interface* (HMI).

Industrial Control System represent numerous different systems and the corresponding industry is large. Within the scope of the present thesis, primarily the security-related aspects of these systems will be handled. The background and theoretical threat analysis are kept on an abstract level in order to be applicable to a wide range of ICS devices. On the other hand, the technical security evaluation applies specifically to the ICS devices used in the experiments. Nevertheless, due to the fact that Industrial Control Systems components resemble each other in functionality and logic, the results of this thesis can be applied to other ICS devices such as electronic smart meters.

1.2 Structure of the thesis

The following thesis is divided into five chapters. Chapter 2 describes the background and necessary knowledge for understanding ICSs. By taking

advantage of an existing threat analysis methodology, chapter 3 focuses on theoretical threat analysis. The real ICS devices are analyzed in chapter 5. Chapter 6 provides a brief overview of existing security solutions. Chapter 7 discusses the overall implications of Industrial Control Systems security and the employed methodology. Finally, chapter 8 concludes the thesis.

Chapter 2

Background

2.1 Industrial Control Systems (ICSs)

Industrial Control System act as a general term describing *Supervisory Control And Data Acquisition system* (SCADA), *Distributed Control System* (DCS) or *Programmable Logic Controller* (PLC) [42]. Further components, such as *Remote Terminal Unit* (RTU) count also as subcomponents of ICS (figure 2.1).

In the manufacturing industry, automation ensures the ability to meet production requirements. Furthermore, ICSs are employed, for example in purification plants, power and nuclear power plants, waste collection facilities, air-conditioning systems, automotive productions, oil and gas infrastructures, railway transportation, paper mills, mining industry, power grids and private enterprise manufacturing systems.

ICSs needs to be controlled and supervised by humans. Furthermore, ICSs might be distributed across several physical locations, and their actions might depend on each other. Therefore, communication is a crucial component in Industrial Control Systems and the communication capabilities needs to be secured.

In the following paragraphs the Industrial Control Systems terms (e.g. SCADA, DCS) will be elaborated. Furthermore, the concepts and use of PLCs and RTUs are explained.

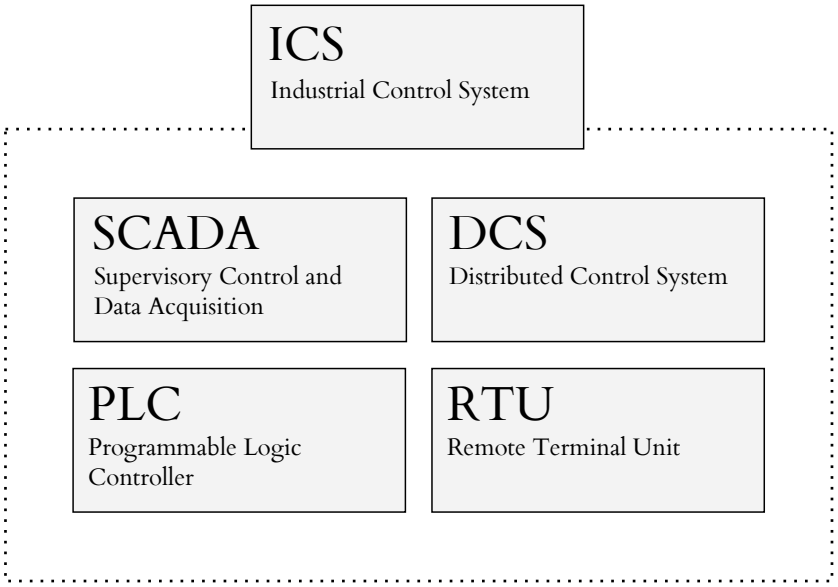

Figure 2.1: Industrial Control System act as general term, describing SCADA, DCS, RTU or PLC.

2.1.1 Supervisory Control and Data Acquisition (SCADA)

Supervision and *control* can be considered as the central keywords in SCADA. The purpose of SCADA systems is to supervise automation systems in order to provide data for decision makers, plant owners and production managers. SCADA systems are nowadays employed in many critical national infrastructures [42].

SCADA vendors increasingly integrate remote web-based access in their devices (e.g. Schneider Electric in their FactoryCast product line) in order to facilitate supervisory and control actions. Having access to real-time information enables decision makers to respond quicker to changes in production, to respond to safety incidents, to have a global overview of the system situation and finally to increase the productivity.

Additionally, SCADA systems increasingly employ commercial off-the-shelf (COTS) hardware and software. Consequently, SCADA devices are no longer exclusively employing proprietary protocols but rather standard and open protocols such as Ethernet. This results in improved interconnectivity of ICSs, but may also enable malicious attackers to gain access to a target system.

Distributed Control Systems (DCS) represent a subset of Industrial Control systems. In contrast to SCADA systems, they support autonomous decision processes [17].

2.1.2 Programmable Logic Controllers (PLC)

According to the original patent, Programmable Logic Controllers consist of a processor taking input from memory, modifying it by executing a control program and storing the result in an output image in memory [3]. Hereby, the PLC is processing data retrieved from sensors and actuators.

PLCs are commonly used within Industrial Control Systems in order to constantly operate in conformance to the pre-defined control sequences. Therefore, PLCs can be viewed as computers, optimized for an industrial use. Depending on the vendors, they may be more robust against low or high temperatures or other environmental forces and can be employed in places unsuitable for general-purpose hardware.

2.1.3 Remote Terminal Unit (RTU)

RTU and PLC are both applied and designed for similar purposes within ICSs: They both perform control tasks and acquire data on-site. According to Motorola [39], RTUs provide increased processing power, communication capabilities and flexibility compared to PLCs. RTUs are for instance extensible through a common backplane where additional modules can be plugged. PLCs from Schneider Electric, such as the Modicon M340 also support this functionality. Consequently, RTU and PLC offer similar features and their applications are overlapping.

2.1.4 ICSs architecture history

ICSs architectures have evolved during the last couple of years. The first ICSs architectures remained so-called *monolithic* architectures as can be seen on figure 2.2). Each remote terminal unit (RTU) was communicating with the SCADA server through a point-to-point connection. Furthermore, the communication was done using proprietary protocols and adapters [41].

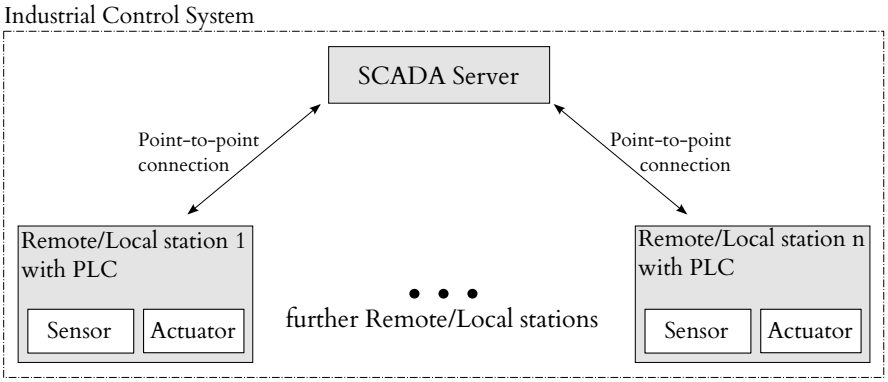

Figure 2.2: First generation of ICSs: Monolithic Architecture [41]

The second generation of ICSs, also called *distributed ICSs* [41], took advantage of the Local Area Network (LAN) development. Compared to the monolithic architecture, several general-purpose computers may be controlling and supervising the remote stations through a SCADA server. It is important to note that vendors used mostly proprietary LAN protocols specialized for real-time traffic. Therefore, all components of these systems have usually been from the same SCADA vendor.

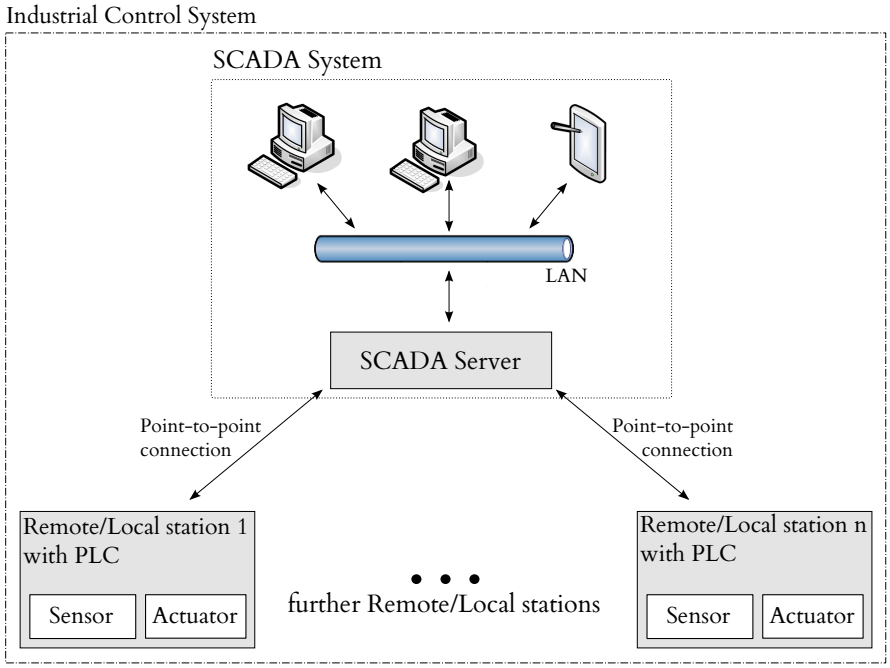

Figure 2.3: Second generation of ICSs: Distributed Architecture [41]

The third generation of ICSs, also called *networked ICSs* [41], represents the current generation. The use of open protocols in order to share SCADA functionality enables new ways of accessing the ICS, such as web-browser-based access. Ethernet and IP are nowadays well known in the automation industry, and ICS vendors are producing nearly every new device with Ethernet functionality. Automation system bus protocols (e.g. Modbus) communicate over TCP/IP and ICS devices employ Web, FTP and Telnet services. The FTP service is for instance used to update firmware and change production configurations. The following figure 2.4 illustrates a generic SCADA architecture.

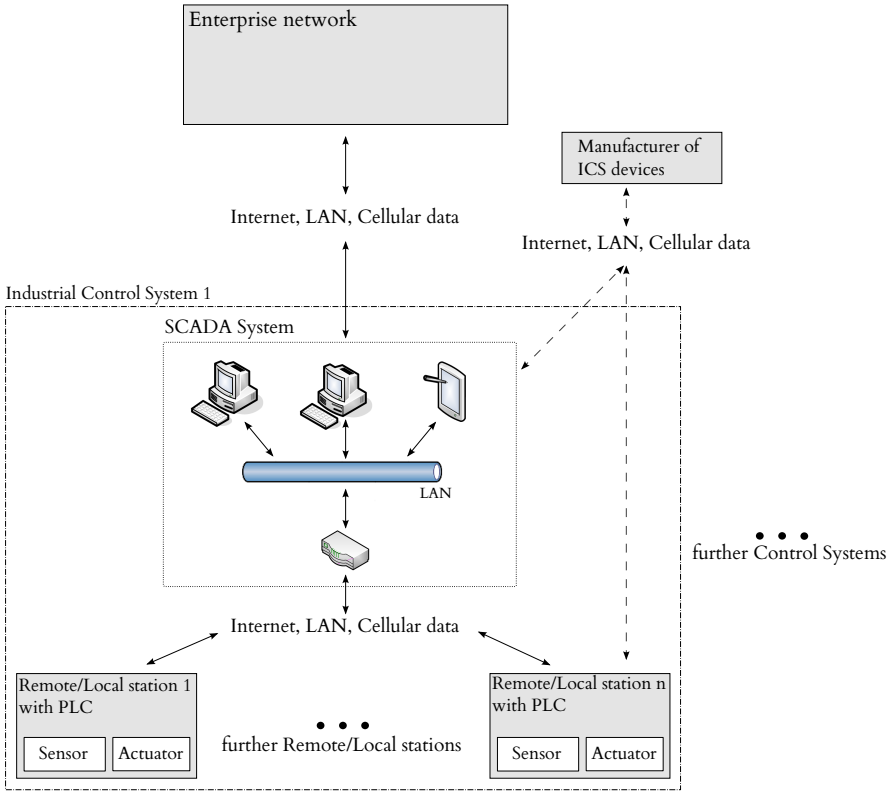

Figure 2.4: Current generation of Industrial Control Systems architecture [17]

2.2 Industrial Control Systems compared to IT systems

Taking into account the history of ICSs, the subsequent chapters will compare ICSs to Information Technology systems (IT).

Despite the fact that control systems are increasingly employing commercial off-the-shelf software they have fundamental differences compared to IT systems. These differences are not limited to technical specificities but also cover experience, education and awareness differences between IT and ICS engineers.

2.2.1 ICSs priorities

A fundamental difference between the IT and automation industry is the order of priorities. In the IT security industry, the CIA triad (confidentiality, integrity, availability) represents the most important security goals, and privacy could be added as a fourth priority. Therefore, in order to prevent attackers from listening sensitive information encryption is considered very important.

On the other hand, in the ICS industry, the SRA model (safety, reliability, availability) represents the most important priorities [16]. Therefore, the automation industry has achieved to produce products which are safe and offer strong robustness, but it has put little effort into designing secure products in the IT-industry sense of the word.

2.2.2 Awareness differences between ICS and IT engineers

Automation systems evolved from monolithic, proprietary systems to the networked and interconnected systems of today supporting a multitude of connections (see section 2.1.4). Recent ICSs communicate over IP and exchange data through various channels like HTTP, Email, Telnet, FTP or the Simple Network Management Protocol (SNMP)[1]. Therefore, ICSs and IT systems are increasingly merging.

2.2.2.1 EtherCAT example

The following example explains how EtherCAT works and how an ICS engineer perceives the security of this protocol.

EtherCAT is an open, real-time network protocol with an emphasis on performance [15]. As the name suggests, EtherCAT uses the Ethernet protocol with the Ethertype 0x88a4 in the Ethernet Header. In order to achieve high bandwidth utilization, EtherCAT pursues a *processing on the fly* [15] approach. An analogy would be to compare an EtherCAT-frame to a train following railways. The train starts at the master and goes through all the

[1] `https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol`

slaves. Finally, the EtherCAT-frame reaches the master again (figure 2.5)[2].

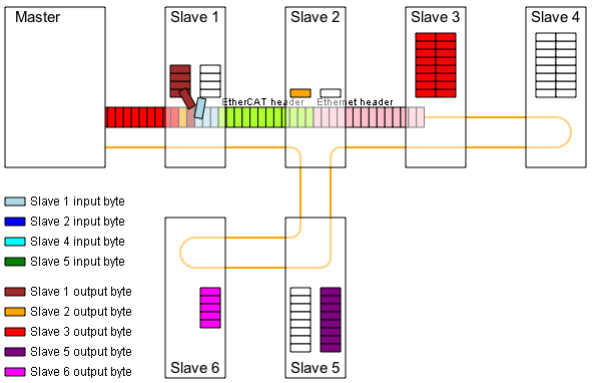

Figure 2.5: A typical EtherCAT installation

Each slave is supposed to only read or write in a specific area of the Ethernet packet (marked with the corresponding color in the figure). ICS engineers are usually not aware of the fact that, since the Ethernet packet is *physically* passing through the slave, the slave can have access to the entire content of the Ethernet frame. In conclusion, in order to cooperate effectively with automation engineers, IT engineers needs to understand such basic assumptions of ICS engineers.

2.2.2.2 Security assumption of ICS engineers

Extensive discussions conducted during this thesis project revealed [29] that ICS engineers assume that intruders needs highly advanced hardware in order to attack ICSs[3]. Therefore, they tend to count complex hardware (e.g. an optimized FPGA) as a security feature, because attackers need to be able to afford such hardware. Nevertheless, automation engineers should consider that an attacker can simply use the available hardware in ICSs by infiltrating it.

[2]Image taken from Wikipedia: `http://upload.wikimedia.org/wikipedia/commons/1/1f/EthercatOperatingPrinciple.svg`

[3]This knowledge is based on a work-experience with an ICS manufacturer

2.2.3 Risk management

The ICS industry has always considered safety as one of the most important characteristics of their products. It is thus appropriate to elaborate on the differences between safety and security. Safety means the protection from natural or unintentional man-made risks [6]. On the other hand, security protects against threats from malicious entities. Therefore, a risk related to safety can also arise from a security threat. Furthermore, the probabilities of security and safety related incidents are different. Malicious attackers who follow several steps in order to attack critical infrastructure behave differently from a natural disaster. Depending on the ICS, its location and importance, security risks might be significantly higher than safety risks.

2.2.4 Security priorities in IT and ICS architectures

In regular IT systems, IT security focuses on protecting IT-assets such as computers and databases. In an ICS, the assets are usually the end devices, PLC and actuators [42], for example the motors. Such differences may create a demand for security solutions that are specifically designed for ICSs.

2.2.5 Lifetime of IT and ICSs

Regular IT systems last approximately 3-5 years. New processors and functionality deprecate hardware and software. On the other hand, an automation device life cycle can be 10-20 years [42]. Additionally, as has already been explained, ICSs are not primarily built with security in mind [7]. Design mistakes are particularly difficult to repair and usually require a product change. Consequently, because of their substantial lifetime, ICS vulnerabilities have a longer-lasting impact than IT vulnerabilities.

2.2.6 Real-time requirements

In contrast to most IT systems, ICSs do have strict real-time requirements. Therefore, communication and security protocols need to be adapted to meet deadlines. Furthermore, encryption or passing information through firewalls consumes valuable time, increases the processing power and may be not suitable for every ICS.

Thus, in order to measure the impact of encryption or firewalls on communications, the delay needs to be measured. According to Kiuchi & Serizawa [30], in a simple ICS system simulation, the introduction of a firewall and encryption based on Advanced Encryption Standard introduces a latency of up to 1.3 ms compared to system without firewall and encryption.

Furthermore, another specialty of time-critical systems is the interaction between humans and the machine. In case of an emergency, an operator at a factory needs to quickly turn on or off a specific device. Therefore, standard IT security mechanisms, such as long passwords are often too burdensome.

2.2.7 Physical interaction

Regular ICSs have a physical interaction with their environment [42]. One example would be the AC drives in nuclear centrifuges. AC drives are designed to control the speed of an electric motor. Moreover, SCADA systems supervise and control the AC drives, which in turn control the motor speed. In a recent well-published incident, the Stuxnet malware manipulated AC drives in order to increase maliciously the speed of the motors [19]. Compared to ICSs, IT systems have little or no physical interaction with their surroundings.

2.2.8 Resource constraints

PLC are specialized hardware and have more limited resources than general-purpose hardware [42]. Therefore, implementations of regular IT security mechanisms might be difficult. On the other hand, ICSs are increasingly merging with regular IT systems. Consequently, recent PLCs are equipped with more computing power for web servers in order to enable the monitoring of production processes in real-time over a web browser.

2.2.9 Patch management

Because of the fact that availability is one of the top priorities in ICSs, patching ICS components is difficult. Patching usually involves thorough testing of new software and implies downtime for the ICS. Furthermore, patching may introduce incompatibilities between system components, putting the

production chain at risk. Consequently, ICS manufacturers needs to find a transparent and accessible way to patch ICS components.

In IT, patching is nowadays widely accepted as standard procedure. The chromium Browser[4] for instance installs *silent patches* without interaction with the user.

2.2.10 Vendor device support

The dependency on specific vendors is more pronounced in ICSs than in IT systems [42]. Although ICSs are using increasingly standardized hardware and software, the unique characteristics of vendor-specific ICS devices means that making any modifications to them requires support from the vendors.

2.2.11 Limited physical access to components

Industrial Control Systems can be distributed across hundreds of kilometers because they can be controller over the Internet and cellular data connections (figure 2.4). Sometimes, control equipment is situated on moving objects such as ships or trains. Therefore, physical access to these devices is limited and hardware or firmware updates involve expensive travel. Consequently, these devices can only be replaced in rare situations.

The following table 2.6 summarizes the former paragraphs explaining the main differences between IT and Industrial Control Systems.

[4]http://www.chromium.org/Home

Property	Industrial Control Systems	Information Technology Systems
Priorities	SRA (Safety, Reliability, Availability) represent the most important priorities in order to meet production requirements. Furthermore, the integrity of parameters sent to the PLC is important.	CIA (Confidentiality, Integrity, Availability) represent the top priorities in IT.
Awareness differences	ICS engineers usually do not reason correctly about the capabilities of malicious attackers. They often assume anything that the specifications do not allow to be impossible.	IT systems engineers are aware that, if an attacker gains physical access to a device, it is very difficult to protect it.
Risk management	Safety is the main concern, even if security issues can put the safety at risk.	Security starts to become integral part of the design process.
Security architecture priorities	End devices, such as PLC, need to be protected from malicious attackers.	Data assets needs to be protected.
Lifetime	10-20 years. The convergence of IT and ICS will probably change this lifetime.	3-5 years.
Real-time requirements	Automation devices have strict real-time requirements and consequently need to meet deadlines.	Usually no strict real-time requirements.
Physical interaction	Important physical interaction with devices, humans and processes.	Few physical interactions with the environment.
Resource constraints	End devices (PLC) have little processing power. Nevertheless, recent devices have significantly increased processing capabilities.	IT systems usually have important and sufficient processing power.
Patch management	Patching is difficult because of availability requirements. Tailored solutions need to be developed.	Patch management is nowadays a standard procedure. Silent updates perform installations in the background.
Vendor device support	Support from original vendor is usually needed.	Support from various sources possible.
Limited physical access to components	Distributed systems make access to devices difficult and expensive.	Access to most of the IT systems possible.

Figure 2.6: Industrial Control Systems compared to IT systems

2.3 Industrial Control Systems incidents

Numerous incidents have occurred over the last years in industrial control systems. Some of them occurred because of generic IT malware and some have been directed attacks. Therefore, it is not always straightforward to distinguish between unintentional and intentional attacks. Furthermore, due to the lack of intrusion detection systems (IDS) and Security Information and Event Management (SIEM), probably most of the actual attacks remain unknown and undetected.

2.3.1 Malware

For more than 26 years, malware has been targeting information systems [18]. The very early malware was written by hobbyists for fun in their spare time. Furthermore, such malware typically was visible to the user as it printed something on the screen. Nowadays, malware is increasingly professional and tries to hide from detection as long as possible.

One recent and well published malware targeting especially ICSs is called Stuxnet. In contrast to regular PC malware, Stuxnet is targeting Windows machines which are used in the scope of SCADA systems. These Windows machines are used in order to supervise and control the PLC, which itself is controlling the manufacturing process. Besides the fact that Stuxnet employed four zero-day vulnerabilities[5] to attack its target, Stuxnet detected if the computer was connected to a Siemens Simatic (Step 7) factory system [19]. If no such system was available, Stuxnet did nothing. When it detected an ICS with Siemens PLCs, it infected the these with specific PLC logic in order to alter the manufacturing process.

According to F-Secure's analysis [19], Stuxnet development had taken approximately 10 man years. It has probably been created in 2009 and the author is most likely a government agency[6]. Furthermore, because Stuxnet only operates if several environment requirements are met at once, it seems that Stuxnet has been created for a targeted attack. Stuxnet has been widely

[5] A zero-day vulnerability is a so-far unknown vulnerability which has not been patched by the affected vendor.

[6] `https://www.nytimes.com/2012/06/01/world/middleeast/obama-ordered-wave-of-cyberattacks-against-iran.html?_r=1&pagewanted=all`

analyzed and can now be seen as a reference implementation for advanced ICS malware.

2.4 Industrial Control Systems recommended practices and standards

Many institutions have become aware of the lack of security protection of Industrial Control Systems. Therefore, they have created a series of standards and best practices. These best practices should help asset owners to become aware of the potential risks and to be able to implement in an organized manner appropriate mitigations. The following table represents a summary of the most common and relevant standards and best practices.

Acronym	Institution and name	Comment
NIST SP 800-82 [42]	National Institute of Standards and Technology - Guide to Industrial Control Systems security	Detailed and complete documentation about ICS security threats and vulnerabilities. Compares ICSs to IT from a security point of view.
ENISA report [16]	European Network and Information Security Agency	Extensive compilation of current information about Industrial Control Systems security. Includes recommendations, survey results are included in the annexes.
ISO/IEC 27001 [24] and 27002 [25]	ISO - International Organization for Standardization, IEC - International Electrotechnical Commission	Very generic standards, not focused on ICS security.
DHS CSSP [10]	Department of Homeland Security National Cyber Security Division	Non-technical, defense-in-depth strategy, risk-model generation, security policies, recommendation of best-practices.
NERC CIP [43]	North American Electric Reliability Corporation Critical Infrastructure Protection	The electrical industry must comply to this standard for bulk-power systems. Reporting of incidents is necessary, asset needs to be defined and denied-by-default policy is enforced.
IEEE 1686 [23]	Institute of Electrical and Electronics Engineers Standard for Substation Intelligent Electronic Devices Cyber Security Capabilities	Practical standard focused on IED in order to complement NERC CIP.
ISA 99 [26]	International Society of Automation - Manufacturing and Control Systems Security	Focuses on establishing and operating a manufacturing and control systems security program.

Figure 2.7: ICS standards, best practices and committees

Chapter 3

Threat analysis of ICSs

3.1 Risks

The following section summarizes different risks related to Industrial Control Systems.

3.1.1 Insecure by design

ICSs have not been conceived with security in mind [42]. As in IT, the security awareness should be present throughout the life cycle of a product [32]. The later security is included into the system, the more difficult and thus more expensive security is. Moreover, the long life cycle of ICS devices renders design insecurities particular dangerous and expensive to mitigate.

3.1.2 Open protocols and commercial off-the-shelf (COTS) components

ICS manufacturers are increasingly employing open and standardized protocols [42]. Open protocols imply reduced costs and easier interconnectivity between industrial devices. Even though open and standardized protocols can be considered as good from a security perspective, they need to be properly implemented, configured and tested on security.

Furthermore, ICSs are employing increasingly commercial off-the-shelf

software (COTS) and general-purpose hardware [4]. Accordingly, these systems face the same risks as desktop devices and attackers can use their existing exploits and attack methods. Therefore, the initial cost of attacking critical infrastructures is substantially lowered.

Finally, the implemented protocol stacks have not been tested extensively for robustness. Penetration tests, even with the most basic scanning methods, can therefore lead to discovery of vulnerabilities. It has been shown that ICS devices can crash if TCP connections are established to specific ports. Furthermore, ping sweeps have caused devices to execute unwanted actions. Consequently, testing an ICS infrastructure requires extremely cautious test methods in order not to disrupt or damage a productive system [22].

3.1.3 Insufficient access control

Web-enabled ICS devices usually do not support strong authentication but require username and password. Furthermore, default passwords are weak and not always changed. Additionally there is usually no mechanism in place to enforce strong passwords. Therefore, the employed authentication methods cannot be considered sufficiently secure for accessing critical infrastructures.

Moreover, access control between corporate networks and control systems is usually minimal [42]. Consequently, an attacker needs *only* to gain access to the corporate intranet in order to compromise the control systems.

3.1.4 Insecure communication protocols

Industrial Control Systems employ a variety of different communication protocols or field buses, such as Modbus/TCP, DNP3[1], PROFINET or EtherCAT [48][15]. Modbus [37] has been specified in 1979 by Modicon and DNP3 [11] in 1993 by GE-Harris Canada. Furthermore, Modbus has initially been specified for serial line communication. Nowadays, Modbus/TCP implementations are widely used in ICSs. Modbus and DNP3 protocols currently do not support authentication, integrity checking, authorization or encryption. Consequently, design weaknesses in the core protocols render Industrial Control Systems insecure.

[1]IEC 62351 and the DNP User Group are currently developing a strong authentication method for the DNP3 protocol [42].

3.1.5 Public information about ICS devices

ICS vendors publish a significant amount of information about their devices online. Firmware and configuration software is readily available on the ICS vendor websites. Consequently, security researchers have access to the necessary binary software in order to find vulnerabilities [28]. Additionally, companies tend to disclose publicly which ICS vendor devices they bought. Therefore, attackers are able to gather sensitive data easily.

ICS can be found without difficulty by searching for specific HTTP headers [14] can find many ICS devices accessible through the Internet. Schneider web servers for instance answer to a HTTP HEAD request with the following information:

Listing 3.1: HTTP Header Response

```
HTTP/1.0 200 Ok
Server: Schneider-WEB/V2.1.4
```

Using a search engine that indexes protocol headers (e.g. HTTP Headers)[2], it is possible to find quickly ICS devices online by searching for specific keywords.

Because the awareness of insecure ICSs raises, security researchers are increasingly targeting automation devices. Consequently, entire security conferences are nowadays held about ICS security[3]. Furthermore, blogger are publicly disclosing weaknesses in various ICS devices[4] or software, often in full-disclosure, without notifying the vendors. Additionally, addresses of publicly available ICS devices are disclosed on the Internet via Twitter[5]. Therefore, the Computer Emergency Response Team Finland (CERT-FI) released a warning in the beginning of 2012.

The malware called Stuxnet has been one of the first malwares targeting industrial automation devices. Stuxnet has been thoroughly analyzed and can be considered as a public reference implementation [42] for highly

[2]Shodan `http://www.shodanhq.com`

[3]`http://www.digitalbond.com/s4/`

[4]`http://reversemode.com/index.php?option=com_content&task=view&id=80&Itemid=1`

[5]`http://www.cert.fi/tietoturvanyt/2012/01/ttn201201121500.html`

advanced ICS malware. Therefore, future malware developers will learn from the mistakes and success of Stuxnet.

3.2 Threats

Threats can be defined as "possible actions that can can be taken against a system" [4] or "a potential occurrence, malicious or otherwise, that might damage or compromise your assets" [33]. These possible actions are executed by different kind of persons, also referred to as threat agents. Please note that, in this thesis, the focus will be kept on security threats and not natural disasters or safety-failures.

3.2.1 Threat agents

According to Hyppönen [36], cyber criminals currently can be divided into three categories:

- Commercially motivated online criminals
- Protesting online groups (hacktivists)
- Nation-state governments

Commercially motivated online criminals are gaining millions of US Dollars every year [27]. They are abusing the online world in order to make a profit, typically by exploiting thousands of individuals. These criminals may also attack critical national infrastructures. It is not uncommon that cyber criminals conduct Distributed Denial of Service (DDoS) attacks against websites in order to extort money from the legitimate owner. Similar attacks can be conducted against online-accessible ICSs.

On the other hand, protesting online groups (e.g. Anonymous[6]) are conducting Denial of Service attacks and disclosing sensitive data from the targets. These groups are protesting against perceived unethical companies, individuals or governments and could also target a critical infrastructure if they deem the associated governments unethical. It is not really possible to

[6] `https://twitter.com/#!/AnonymousIRC`

forecast what these groups are attacking in their next steps. Therefore, the threat posed by them should not be underestimated.

The third, and probably most skilled category of cyber criminals are nation states. Stuxnet for example has been created by the United States [9] and is such an advanced and complicated malware, that it could only have been created by someone with significant resources [52] [50] [12]. The F-Secure Stuxnet analysis estimates that Stuxnet has been a 10-man-year project [19], and the necessary knowledge is probably only accessible to governments.

Last but not least, extreme hacktivists are successfully operating worldwide in several countries. Once they are able to alter critical infrastructure, they would have a new way to conduct effectively cyber terrorist activities.

3.2.2 Threat modeling

The objective of this section is to identify possible threats against ICSs. Therefore, in order not to forget eventual threats, it is crucial to follow a proven methodology. Additionally, ICSs and IT systems are converging and proven threat models for the IT systems exists. Thus, this study conducts an analysis on how to apply the Microsoft STRIDE model [34] [67] to ICSs and evaluates the possibility of applying IT threat models to ICSs. STRIDE has been developed by Microsoft primarily for conducting threat analysis on IT systems and particularly on software. STRIDE stands for **S**poofing, **T**ampering, **R**epudiation, **I**nformation disclosure, **D**enial of Service (DoS) and **E**levation of privilege.

The STRIDE model can be executed by following different steps:

1. Visualizing the system or software architecture
2. Identifying entry points, data flows, trust boundaries and the assets to be protected
3. Decomposing the architecture into subsystems and creating Data Flow Diagrams (DFD) of the subsystems
4. Identifying threats by analyzing the data flows, data stores and processes in the DFD
5. Prioritizing and mitigating the threats

The fictive company we are analyzing is using an ICS and wants to be able to increase interconnectivity and to conduct faster decision making. Therefore, the objective is to have a centralized supervision, coordination and decision making. The necessary information results from sensor readings of remote and local industrial stations.

For achieving this objective, the following goals are identified:

- Remote stations need to be supervised from any distance. They can be located on ships, trains or fixed positions.
- As in most ICSs architectures, the remote stations and database server represent the assets [42].

A detailed description of the fictive system architecture can be found in figure 3.1.

Figure 3.1 visualizes the Industrial Control Systems architecture adopted for the threat analysis. Comparing the generic ICS architecture (figure 2.4) with figure 3.1 highlights similarities and enables the reader to have a more concrete image of an ICS infrastructure. Furthermore, the remote station part of this architecture resembles the real ICS devices analyzed in chapter 5. Consequently, this threat analysis helps to prepare and focus a technical security analysis.

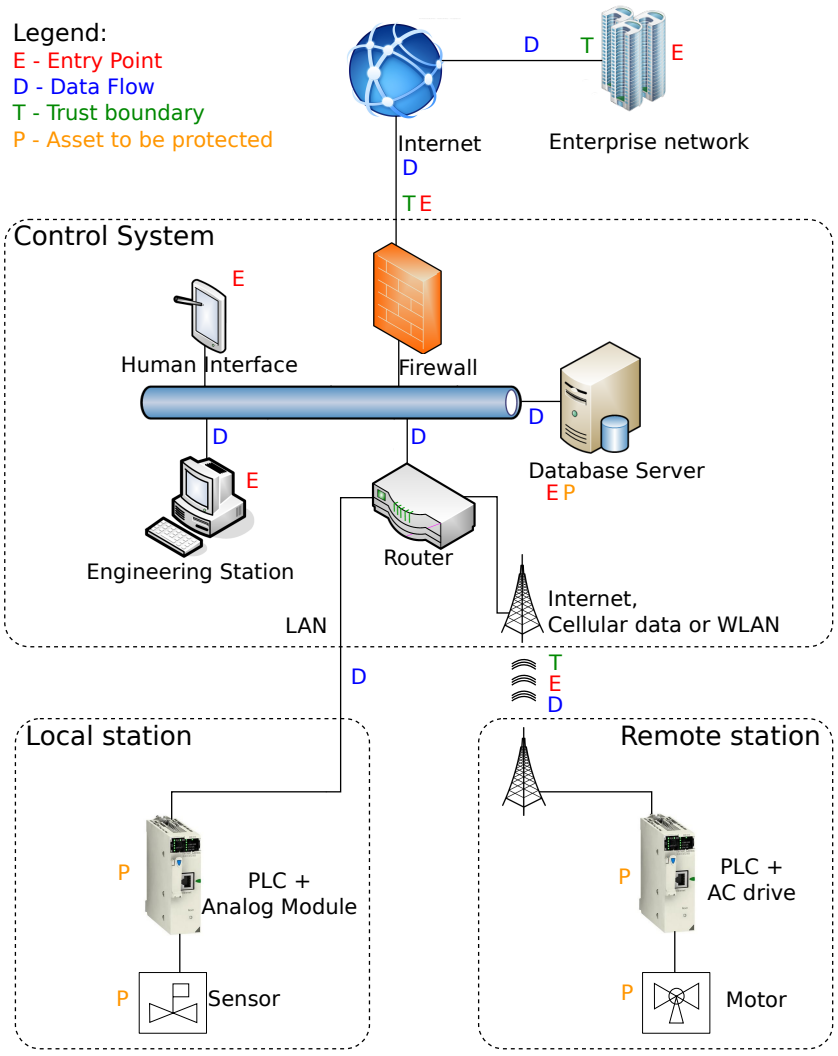

Figure 3.1: Example ICS infrastructure for threat analysis.

Identifying interfaces is the next step in the Microsoft STRIDE model and consist of identifying existing data flows, entry points, trust boundaries and to be protected assets. The following figure 3.2 explains these terms.

Interface name	Description
Trust boundary	Represents an attack surface and consists of a frontier between networks. In regular IT systems, a network interface or firewall can be considered as a trust boundary [34]. Since trust boundaries also exist in ICSs, the same reasoning can be applied.
Data flow	Place where data is exchanged.
Entry point	Places where data flows cross trust boundaries. For example, the entry points for a networked SCADA device might be: • Web server interface on port 80 • Modbus/TCP interface on port 502 • Serial line communication interface • USB port • SD card slot The risk posed by the entry points depends on which entities can gain access to them.
Protected asset	PLCs and RTUs and the equipment controlled by them as well as the data collected from sensors are the primary assets of Industrial Control Systems [42]. System data such as passwords and firmware may also be treated as assets in the threat model.

Figure 3.2: STRIDE interfaces

3.2.2.1 Architecture decomposition and data flow diagrams

In the following, we will focus on the remote station part of the system in figure 3.1 containing the AC drive, the motor and the communication with the control system. In order to understand how the AC drive is operating, the following use case will be considered:

The Motor, attached to the AC drive is turning at a given speed. The PLC controls the AC drive, which both sets and senses the motor rounds per minute (RPM). The RPM are set by and reported to the PLC. The PLC communicates with the control system. An engineer at the control system can

supervise and control the entire equipment.

Furthermore, in order to model the subsystem correctly, a syntax for the data flow diagram needs to be established. The Microsoft STRIDE model uses four different graphical items to visualize a DFD (see the legend in figure 3.3).

By analyzing the architecture and the use case description, a data flow diagram can be created. Thus, figure 3.3(a) represents the first version of the DFD diagram. In this diagram, the engineer communicates with a control process (here the engineering station) and the control process can send control commands over the trust boundary to the AC drive. Additionally, Microsoft recommends the following verification rules in order to validate data flow diagrams [34]:

- Data stores should have a reader and a writer.
- Processes need to read and write data.
- Related components (e.g. two consecutive processes) can be merged together.
- System components separated by trust boundaries need to be separated into different DFDs.
- The diagram should not represent the implementation but rather the major function of the system.

Having these rules in mind, the first DFD (figure 3.3(a)) can be verified: Each data store in the diagram has a reader and a writer. Furthermore, each process reads and writes data. Additionally, related components, such as the motor part (physical entity, motor process and motor voltage) can be merged.

Finally, the two sides of the trust boundary are represented on the same DFD. Since each side of the trust boundary cannot trust each other, the improved DFD (figure 3.3(b)) focuses on the left DFD.

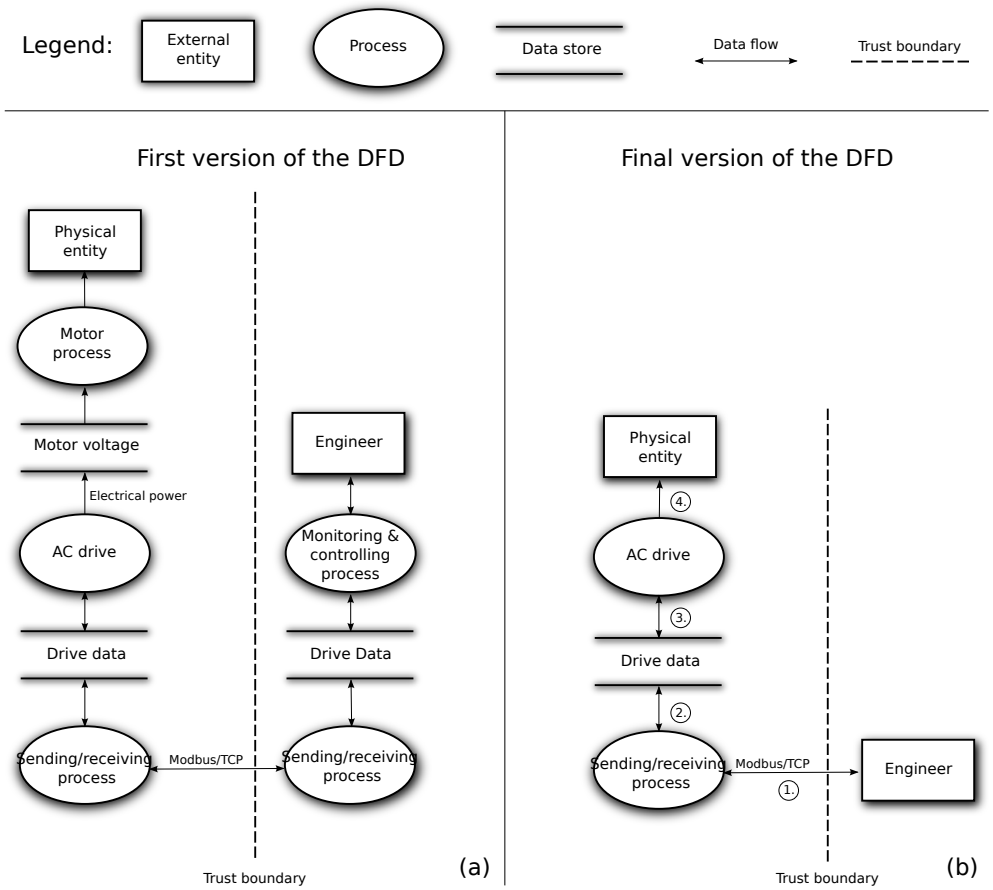

Figure 3.3: Two different data flow diagrams of the industrial control subsystem. On the left a non-optimized diagram and on the right an optimized version.

3.2.2.2 Identifying threats

Having a clear data flow diagram, it is now possible to identify the different threats. Each data flow diagram item is susceptible to a specific subset of the STRIDE threats [35] as shown in figure 3.4.

Item	**S**poof.	**T**amp.	**R**epud.	**I**nfo. discl.	**D**oS	**E**levat.
External Entity	X		X			
Process	X	X	X	X	X	X
Data store		X	X	X	X	
Data flow		X		X	X	

Figure 3.4: How the STRIDE threats affect the data flow diagram items

Identifying threats is the most important step in the threat analysis. Therefore, the analyst needs to look at the Industrial Control System with the mindset of an attacker. We will use figure 3.3(b) as an example for the threat identification and starting analyzing the data flows.

The data flow 1 between the engineer and the sending and receiving process is susceptible to data tampering especially when transmitting over the Internet and information disclosure when not properly encrypted. The data flow can also be dropped to conduct denial of service attacks. The same threats apply to the data flows 2, 3 and 4. Furthermore, proper authentication should be put into place for each data flow.

In addition to the threat identification, a risk analysis could be conducted. For example, data flows within a trust boundary represent a lower risk than flows that cross a boundary.

In the next step, the data stores are analyzed. Similar to the data flow, the data stores are susceptible to tampering, information disclosure and denial of service attacks. Additionally, the data store may be vulnerable to repudiation attacks. If no proper protection exist, it is not trivial to trace who has altered the data store, and therefore it is not possible to prove that the data store has been altered by the AC drive or the sending and receiving process. In conclusion, if no proper protection is in place, someone could maliciously alter the data store and it would not be possible to prove who it was.

Finally, according to the Microsoft STRIDE model, processes are vulnerable to all the STRIDE threats [35]. The sending and receiving process could for example be spoofed and wrong data could be sent to the AC drive from the PLC. For example, the malware Stuxnet modified PLC logic and sent malicious configurations to AC drives in uranium-enrichment facilities. Consequently, the drive configuration [19] has been manipulated. Moreover, the running processes on the PLC may have only limited access rights to all sensors. An attacker could tamper with a process and try to extend its rights, which can be referred to as elevation of privilege.

3.2.2.3 Mitigating the threats

For each identified threat, mitigations need to be established. The pertinent question is to ask which security mechanisms should be implemented. Usually, the best mitigations are well-known and widely used security protocols.

In order to prevent tampering and information leakage, the data flow 1 in figure 3.3(b) should be encrypted. Multiple encryption technologies exist and an appropriate solution should be found. If symmetric-key protection is used and an attacker is able to gain physical access to the device, he might be able to extract the symmetric key and decrypt and modify messages. Furthermore, proper key-management needs to be employed in order to be able to change the encryption keys if needed.

Another possibility to prevent tampering and information leakage of the dataflow is to use public key cryptography. The public key of the control system could be stored on the remote station. Therefore, the remote station could regularly generate a random symmetric key and sent it to the control system encrypted with the public key of the control system. Consequently the control system is able to decrypt the symmetric key and use it in subsequent messages. This solution does not provide authorization and needs to be developed further. In order to protect against key theft, the remote station should be protected against physical attacks or have mechanisms for detecting if it has been tampered.

In order to protect data stores from repudiation attacks, a central log management system could be employed. Before accessing a data store, the reader or writer should authenticate itself and the server needs to log the access in the log management system.

In conclusion, mitigating the threats is important in order to minimize the risks against Industrial Control Systems.

3.2.2.4 Conclusion

Another way to identify possible threats to an industrial Control System is by applying the experience of an security consultant and thinking about different attack vectors. The figure 3.5 visualizes for instance a very generic threat analysis conducted on an example enterprise network connected to an Industrial Control System.

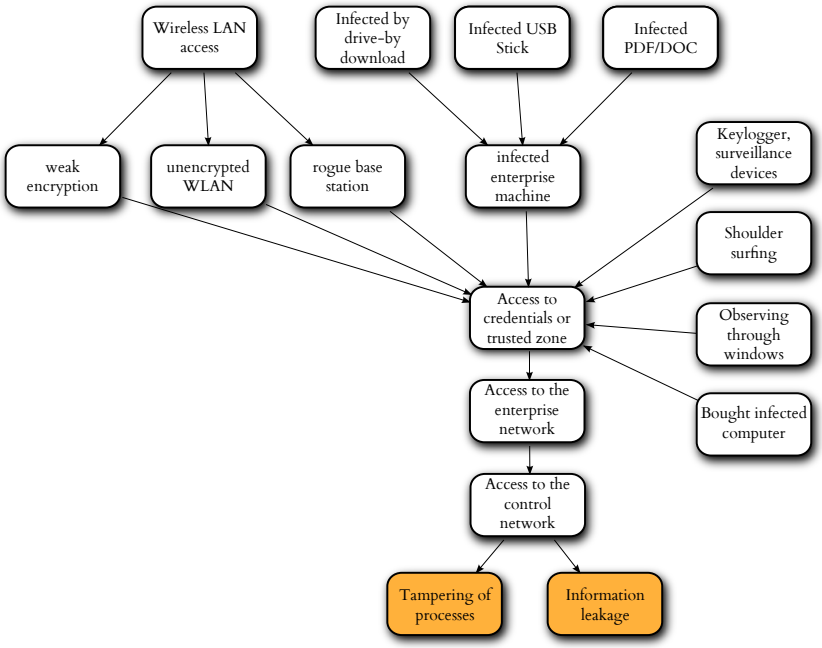

Figure 3.5: Possible threats to an enterprise network

The aim of this study was not to find mitigations for specific ICS threats but rather to evaluate the possibility of employing a well known threat model for IT to Industrial Control Systems. The study shows that the Microsoft STRIDE model can well be used for conducting a threat analysis on an existing ICS infrastructure.

Chapter 5 focuses on the technical security analysis of ICS devices. The threat analysis helped to focus the technical security analysis on specific parts of the analyzed devices. One result of the threat analysis is for example, that it is particularly important to analyze the communication between the Human-Machine Interface and the PLC. The security analysis in chapter 5 reveals vulnerabilities in this communication channel.

Chapter 4

Guidelines for the security analysis of ICS devices

Based on the previously described analysis, the present chapter provides guidelines on how to evaluate the security of ICS components such as PLCs.

4.1 Documentation

Before focusing on technical details, it is important to consult the documentation of the analyzed ICS devices. The documentation explains for example how the device is working, which communication capabilities exist, and mentions security related information such as default usernames and passwords. Users who do not change default login credentials are more likely exposed to security threats. Consequently, ICS vendors should enforce strong customized credentials for accessing ICS devices.

4.2 Attack vectors

Recent ICS devices offer numerous modern IT features, such as Ethernet communication and HTTP and FTP servers. Due to these different communication capabilities, different attack vectors exist and should be taken into account while evaluating the security of an ICS device. The following sections enumerate some of the main attack vectors specific to ICS.

4.2.1 Robustness testing

Fuzzing or arbitrary data injection [31] [66] [21] [1] [56] is the process of testing the robustness of an implementation by injecting malicious or arbitrarily chosen data in the implementation under test and analyzing the outcome.

Depending on the robustness of the tested application the following events might occur:

- The application under test ignores the injected data and continues working as expected.
- The application under test ignores the injected data but consumes increased resources. This could lead to resource exhaustion, also called Denial of Service.
- The application under test executes part of the injected data. This leads in most cases to a crash of the application under test as it might allow execution of arbitrary code.

Basically, every service accepting input data can be tested for its robustness. Therefore, typical IT services such as HTTP, FTP or Ethernet should be fuzzed. Furthermore, ICS protocol implementations like Modbus/TCP (see chapter 4.3) can be tested from this point of view.

4.2.1.1 Types of fuzzers

The following types of fuzzers can be implemented:

- Random fuzzer
- Generation-based fuzzer
- Mutation-based fuzzer

A *random fuzzer* arbitrarily generates and sends data to the target. Although the tested application may reject the random data in most cases, this technique is able to find vulnerabilities in the software [31].

The second category, *generation-based fuzzer*, implements parts of the tested protocol. Therefore, such a fuzzer generates more *valid* data than a

random fuzzer [31]. Moreover, semi-valid data is more likely to pass sanity checks on the tested application than random data. Consequently, the probability of triggering problems with generation-based fuzzers is higher than with random fuzzers. Nevertheless, since generation-based fuzzer are more advanced, they are also more complicated and time-consuming to implement.

The last category, *mutation-based fuzzer*, can be explained by taking advantage of an example: when a Modbus client sends a request to the Modbus server, an intermediate proxy (the fuzzer) is altering arbitrarily parts of the message. For implementing a mutation-based fuzzer, the tested protocol does not need to be studied. Furthermore, the fuzzer is generating semi-valid data. One drawback of the mutation-based fuzzer is that an active connection between the Modbus client and server needs to be maintained [31]. Additionally, the mutation-based fuzzer might not cover all parts of the tested protocol.

Consequently, the effectiveness of robustness testing can be increased by using different types of fuzzers. A series of public fuzzing software such as Brute force Exploit Detector (BED)[1] are available on the market and are ready to be used. However, for advanced fuzzing, dedicated and customized software should be written and employed.

4.2.2 Web application security

Some ICS devices such as Schneider PLCs, nowadays support HTTP server and dynamic web applications. Therefore, they are also vulnerable to standard web application security issues. Assessing web applications in IT is a well-known topic and, therefore, resources such as the *OWASP Top 10*[2] [49] can be taken as references for assessing them. Web application assessments should be done with a mix of automated and manual testing in order to maximize the coverage and effectiveness.

4.2.3 Firmware analysis

Commonly, the firmware of ICS devices can be downloaded from the ICS vendor's website. If the firmware is not encrypted, it can be reverse engineered with medium effort. Hereby, it is possible to find security related

[1] Protocol fuzzer included in the Linux penetration testing Distribution Backtrack
[2] `https://www.owasp.org/index.php/Top_10_2010-Main`

issues, without even having access to the physical ICS device. Especially hard-coded usernames and passwords can be found with this method. In some cases, even undocumented access methods such as backdoors can be identified [55].

4.2.4 Ladder logic upload and download

Ladder logic [13] is commonly used to refer to the software which is running on the PLC firmware. This software decides, depending on the values of sensors and actuators, what the PLC executes. Because this is business-critical software, only highly authorized persons should be allowed to create and upload this software to the PLC. That is, an attacker should not be able to download from or upload to the PLC custom ladder logic. However, as we will see later, this is not the case for some of the nowadays ICS devices.

4.2.5 ICS related problems

While evaluating the security of an ICS device, it is important to identify especially ICS related problems such as elaborated later. Giving concrete guidelines is not trivial, but the software should be analyzed from the perspective of an attacker. One test scenario could be: "If it is possible to alter the ladder logic remotely, can the HMI still be considered as a reliable source of information?". If the HMI is not reliable, this can be considered as a problem.

4.2.6 Control software security

ICS devices such as PLCs are usually configured, updated and operated through control or programming software, running on a connected Windows machine. This control and HMI software may contain security flaws, putting at risk the entire control system[3]. Furthermore, control software needs to be updated and the update mechanism needs to be secure and trustworthy.

[3]100 bugs in 100 days, DerbyCon 2011 Talk `https://www.youtube.com/watch?feature=player_embedded&v=29S_Beg71dA`

4.2.7 Embedded operating system security

PLCs nowadays often use COTS operating systems such as VxWorks[4]. VxWorks is powering more than 1 billion embedded systems[4] and critical vulnerabilities have been discovered in earlier versions [64]. Therefore, the security of the PLC operating system should be taken into account. Additionally, the process of updating the operating system, often conducted a as firmware update, needs to be tested (see section 4.2.9).

4.2.8 Undocumented features

PLCs might offer undocumented features such as activated debugging functionality [63]. Network port scans (covering all 65535 TCP and UDP ports) and physical Joint Test Action Group (JTAG) connectors can reveal existing debugging ports which could be exploited by an attacker. Vendor-implemented backdoors such as those found on the RuggedCom devices[5] may be discovered.

4.2.9 Firmware update procedure

Modern PLCs have operating systems and need to be updated by adding new functionality and fixing security issues. Conducting a firmware update of the ICS device is a critical procedure because the device should only accept legitimate firmware updates. An attacker should not be able to craft customized firmware and upload it to the PLC. Therefore, the firmware update procedure needs to be analyzed: One should study how is firmware loaded onto the device or whether it is possible to upload customized firmware.

Besides the firmware update procedure, one should also look into the firmware distribution channel (e.g. the website of the vendor). This channel is responsible for providing clean and verified firmware images. Therefore, it should be regularly audited and the integrity of provided firmware updates needs to be verified.

[4] `http://www.windriver.com/products/vxworks/`

[5] Default username "factory" and password retrieved from MAC address allow complete administrative control over the ROS devices, `http://www.us-cert.gov/control_systems/pdf/ICS-ALERT-12-116-01.pdf`

4.2.10 Further observations

In addition to these guidelines, it is important to inspect all available information which can be retrieved from the PLC. The HTTP server may for instance offer additional services via XML/SOAP requests, or the FTP server may store confidential information for accessing further services of the PLC.

4.3 Modbus Protocol

Over 70 companies are members of the Modbus organization [38] since Modbus is widely supported by ICS devices. For this reason, Modbus [37] [8]is a concerning protocol, especially from a security point of view. The following section explains how the Modbus protocol works. Moreover, a Modbus/TCP specific fuzzer is presented.

The Modbus protocol has first been specified in 1979 by Modicon [37]. Its purpose and design are focused on controlling and monitoring Industrial Control Systems. Furthermore, the first Modbus specification and implementations were designed to work over serial communication. Later, communication over-IP became necessary and Modbus/TCP was specified [37]. Modbus/TCP is built on the top of TCP and can therefore be used over Local Area Networks (LAN) or the Internet.

Modbus/TCP is a binary protocol and composed of two elements: the ADU (the header) and the PDU (the payload).

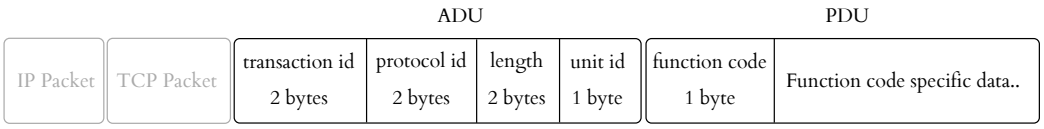

Figure 4.1: Modbus/TCP packet with a header (ADU) and payload (PDU)

In the following, the different fields of the ADU are explained:

- Transaction id - a counter which is incremented by one for each exchanged Modbus/TCP packet.
- Protocol id - needs to be 0x0000 for Modbus
- Length - the length of the total Modbus/TCP data minus 6 bytes

- Unit id - should be either 0x00 or 0xFF for Modbus/TCP

The PDU structure depends on the function code. Modbus supports by default up to 127 function codes. Function codes 65 to 72 and 100 to 110 are specified as *user defined*. The remaining represent *public* and *documented* function codes. Function code 1, for instance, corresponds to the *Read Coils* function. Modbus was first specified for over serial communication to turn relays (coils) on or off. Basically, function code 1 enables the Modbus client to read x bits, beginning at start address y on the Modbus server.

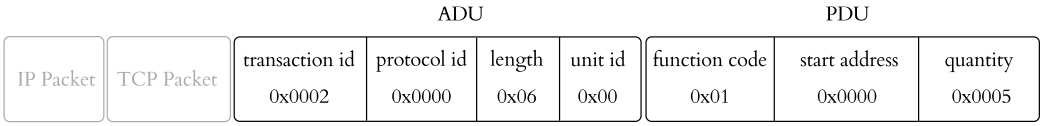

Figure 4.2: The Modbus/TCP read coils request. The Modbus client requests to read 5 bits, starting at the address 0x0000.

4.3.1 Modbus/TCP exchange

Modbus is a stateless protocol and does not support authentication or encryption [37]. Therefore, the Modbus server will interpret and possibly answer every Modbus request it receives.

The Modbus client (also called master) sends a request to the server (also called slave) and the server sends either an answer, exception or nothing back (figure 4.3).

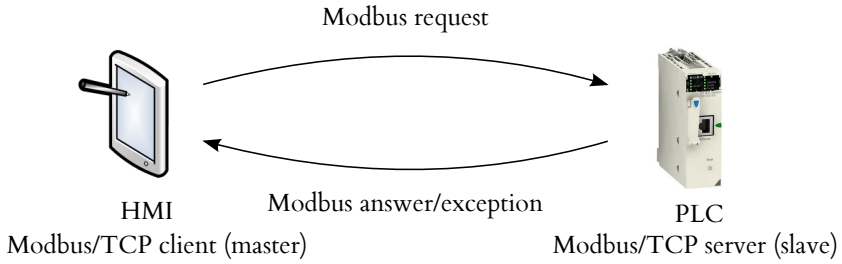

Figure 4.3: The Modbus/TCP communication consists of sending a Modbus request to the Modbus server. The server then responds to the Modbus client with an answer.

An answer to a read coils request is visualized in figure 4.4. Output 1 corresponds to the least-significant bit (LSB) and output 5 to the fifth most

significant bit (MSB). Because the request only requested 5 bits of data, three bits of the output (bits 6,7,8) are unused.

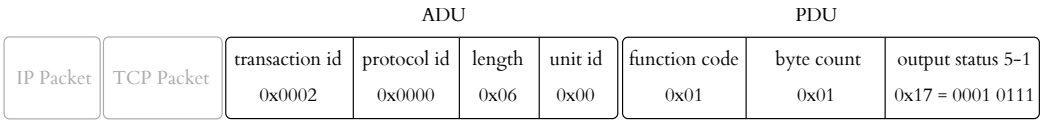

Figure 4.4: The Modbus/TCP communication consists of sending a Modbus request to the Modbus server. The server then responds to the Modbus client with an answer.

When the server receives an erroneous request or is not able to handle the request, an exception message is returned. The function code of an exception message corresponds to the original function code of the request + 0x80. Figure 4.5 displays an exception message in response to a read coils request.

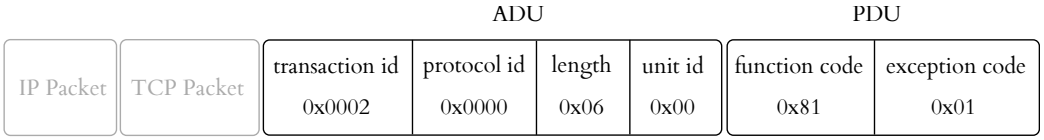

Figure 4.5: Modbus/TCP exception for read coils (function code 1). Exception code 1 corresponds to *invalid function code*.

4.3.2 Modbus/TCP packet crafting library

In order to test the robustness of Modbus/TCP implementations we developed a Modbus/TCP packet crafting library.

The library can be used as a Modbus fuzzer to send arbitrary and unexpected data to the Modbus server. Furthermore, the library is able to monitor the reaction of the Modbus server. If the Modbus server stops responding, it is assumed that the server stopped working properly. Besides packet generation, sent network packets needs to be recorded in order to repeat exceptions. The recording can be done by using a variety of specialized software such as Wireshark [65].

4.3.2.1 Scapy Modbus fuzzer

The Modbus/TCP library has been implemented with the help of Scapy [47], a python packet manipulation framework.

Scapy is extensible and supports numerous protocols out-of-the-box. Nevertheless, Modbus/TCP is not supported by default. Therefore, in order to extend Scapy with the Modbus/TCP protocol, we created an additional library. After loading the library into Scapy, creating a Modbus packet can be done as documented in listing 4.1.

Listing 4.1: Create a Modbus/TCP packet for reading coils

```
>>> (ModbusADU()/ModbusPDU01_Read_Coils()).show2()
###[ ModbusADU ]###
  transId= 0x1
  protoId= 0x0
  len= 0x6
  unitId= 0x0
###[ Read Coils Request ]###
     funcCode= 0x1
     startAddr= 0x0
     quantity= 0x1
```

Besides the read coils function code, the Scapy Modbus library supports the following function codes:

- 01 (0x01) Read Coils
- 02 (0x02) Read Discrete Inputs
- 03 (0x03) Read Holding Registers
- 04 (0x04) Read Input Registers
- 05 (0x05) Write Single Coil
- 06 (0x06) Write Single Holding Register
- 07 (0x07) Read Exception Status (Serial Line only)
- 15 (0x0F) Write Multiple Coils
- 16 (0x10) Write Multiple Holding Registers
- 17 (0x11) Report Slave ID (Serial Line only)

In addition to these function codes, specific answer and exception codes

associated to each function code are supported. Since different devices support different function codes, the Modbus Scapy library has a method for identifying the supported function codes (listing 4.2). The following example illustrates a Modbus slave, supporting in total 13 Modbus/TCP function codes. With the exception for function code 90, most of these function codes are documented [37].

Listing 4.2: Using the Modbus/TCP library for finding supported function codes of a Modbus/TCP server

```
>>> connection = connectToTarget("169.254.0.2")
>>> getSupportedFunctionCodes(connection)
Looking for supported function codes..
Function Code 1 is supported.
Function Code 2 is supported.
Function Code 3 is supported.
Function Code 4 is supported.
Function Code 5 is supported.
Function Code 6 is supported.
Function Code 8 is supported.
Function Code 15 is supported.
Function Code 16 is supported.
Function Code 22 is supported.
Function Code 23 is supported.
Function Code 43 is supported.
Function Code 90 is supported.
[1, 2, 3, 4, 5, 6, 8, 15, 16, 22, 23, 43, 90]
```

In order to test a Modbus/TCP server, several predefined functions have been added to the packet-manipulation library. Furthermore, the functionality of Scapy can be extended and customized robustness testing is accomplished with few lines of code (listing 4.3). The following example creates 65535 Modbus/TCP read coils requests and sends them to the target server. For each packet, the quantity field is incremented by one.

Listing 4.3: Using the Modbus/TCP library for fuzzing

```
>>> connection = connectToTarget("169.254.0.2")
>>> for p in ModbusADU()/ModbusPDU01_Read_Coils(quantity=(1,65536)): \
        connection.send(p)
...
<ModbusADU  |<ModbusPDU01_Read_Coils  quantity=0x1 |>>
<ModbusADU  |<ModbusPDU01_Read_Coils  quantity=0x2 |>>
<ModbusADU  |<ModbusPDU01_Read_Coils  quantity=0x3 |>>
<ModbusADU  |<ModbusPDU01_Read_Coils  quantity=0x4 |>>
...
```

Chapter 5

Security analysis of Schneider Electric ICS devices

Note: Some critical vulnerabilities were found and the vendor has requested us to withhold the information until September 2013. This section represents the less serious vulnerabilities. Nevertheless, it should give a good idea of the range of security issues encountered in ICS devices. The ICS-CERT has been informed about the vulnerabilities.

Taking into account the guidelines presented in the previous chapter, a technical security evaluation of ICS devices has been conducted. The security analysis was made possible and supported by Schneider Electric. Schneider provided their current mid-range PLCs for testing purposes. Similar issues were found in other ICS devices from other vendors.

5.1 Description of the ICS devices

The ICS devices are built into a portable case and used for demonstration purposes at conferences and exhibitions. The first step of the security analysis consists of reading the documentation (see 4.1).

Figure 5.1 visualizes the technical components of the tested ICS devices. A picture of the devices can be found in annex A.1. The devices are:

- A Programmable Logic Controller (Modicon M340 PLC) with:

  - PLC (P34 20302) with Ethernet and USB connectivity
  - CANopen network card
  - Optical sensor card
  - Analog input card
  - Ethernet module (BMX NOE 0100)
- Human-Machine Interface, HMI (Magelis XBT GT) connected via Ethernet
- Ethernet switch
- Optical sensors (Osiris XUAH0515)
- Servo Motor Drive (Lexium 05)
- Motor
- Temperature probe (PT100)

The PLC can be considered the central element of the architecture. It can be programmed with ladder logic and it interfaces the communication between the Motor, the HMI and the other components. In addition to the hardware, Schneider provided the Windows control software necessary for programming and controlling the PLC.

Furthermore, the ICS devices were shipped with pre-installed software for demonstration purposes. In total eight programs were installed on the PLC, showing different capabilities of the devices. One example program called *Speed control* (see picture A.2 in the Annex) controls the speed of the motor with the built-in potentiometer.

5.1.0.2 Documentation findings

The documentation of the ICS devices reveals that the default username and password for the web interface are USER and USER. Therefore, after the first login attempt, a password change should be required from the user. A note in the device-manual cannot be considered as sufficient to enforce this.

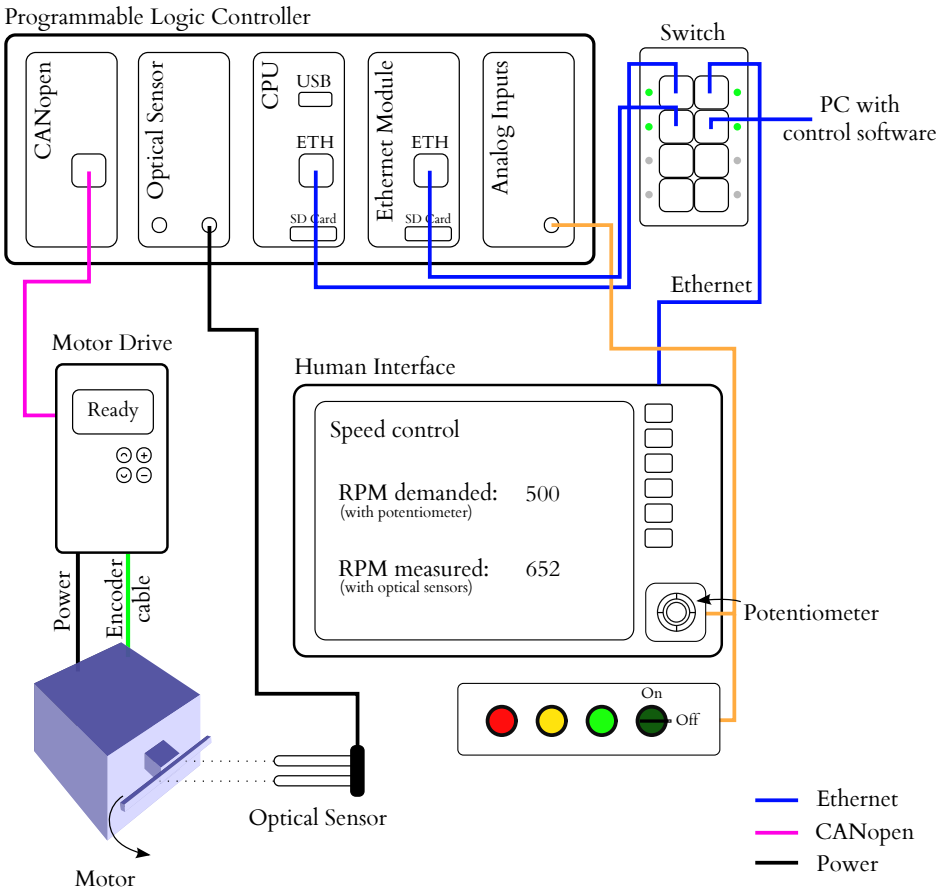

Figure 5.1: ICS devices from Schneider Electric. The components correspond to the current ICS devices (May 2012) from Schneider.

5.2 Technical analysis

In the following section, guidelines for the security analysis of ICS devices from chapter 4 are applied to the ICS devices.

5.2.1 Robustness testing

The subsequent chapters describe the robustness testing conducted on the different IT services of the ICS devices.

5.2.1.1 HTTP

In order to test the robustness of a web server, existing fuzzers such as BED can be used. BED is written in Perl and is included in the Linux distribution Backtrack 5[1]. BED issues thousands of unusual HTTP requests towards the target web server in order to trigger buffer overflows and other vulnerabilities. An example HTTP request issued by BED can be seen in figure 5.2.

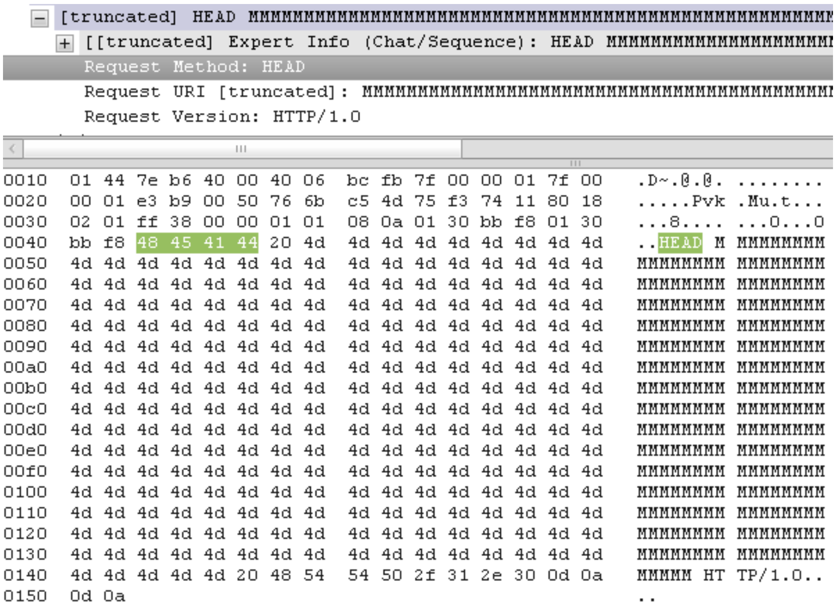

Figure 5.2: HTTP HEAD request with a long filename.

BED uses by default the 'A' character for long filenames. Since some implementations filter long 'A' sequences to avoid buffer overflow testing [31], the letter has been changed to 'M'.

BED has been run against the PLC web server (Schneider-WEB/V2.1.3) and the BMX NOE 0100 web server (Schneider-WEB/V2.2.0) without success. Nevertheless, Reid Wightman from Digitalbond has analyzed the Schneider Modicon Quantum device and was able to find buffer overflows with BED affecting the HTTP (Schneider-WEB/V2.1.4) and FTP service [53].

[1] Penetration testing Linux Distribution Backtrack (`http://www.backtrack-linux.org/`)

5.2.1.2 FTP

The PLC is offering access to an FTP server. The server stores, for example, web server files. Therefore, BED was also used to test the robustness of the FTP server, but no reproducible bugs were identified.

Nevertheless, while downloading with FileZilla all files from the FTP server, the PLC crashed in approximately 50% of the cases. Moreover, the PLC displayed a *CAN error message*. Fortunately, this instability issue has been fixed in the current firmware version.

5.2.1.3 Modbus/TCP

IBM recommends testing industrial protocol implementations thoroughly [22]. Therefore, the robustness of the Modbus/TCP implementation was tested by using the developed Scapy library presented in section 4.3.2.1.

No buffer overflow or similar vulnerability was discovered during the research on the PLC. Nevertheless, while sending generated Modbus/TCP packets to the PLC, the motor started to turn. This can be explained with the fact that the motor can be controlled through the HMI which is communicating over Modbus/TCP with the PLC. Since Modbus/TCP does not support authentication, replaying or forging valid packets is trivial.

5.2.1.4 Canape

Besides the generation-based Scapy fuzzer, another network testing software called Canape[2] was also employed. Canape is a binary network-protocol testing tool which is acting as a proxy between the client and the server. Therefore, it is suitable to be used as an automated mutation-based fuzzer.

Canape was used to randomly alter the communication between the control software and the PLC. A *Net Graph* has been designed (see figure 5.3) which describes what should be done with the packet, depending the state of the connection:

1. The control software first sends approximately 50 packets to the PLC

[2]Canape has been presented at the Blackhat Europe 2012 and is available at `http://www.contextis.de/research/tools/canape/`

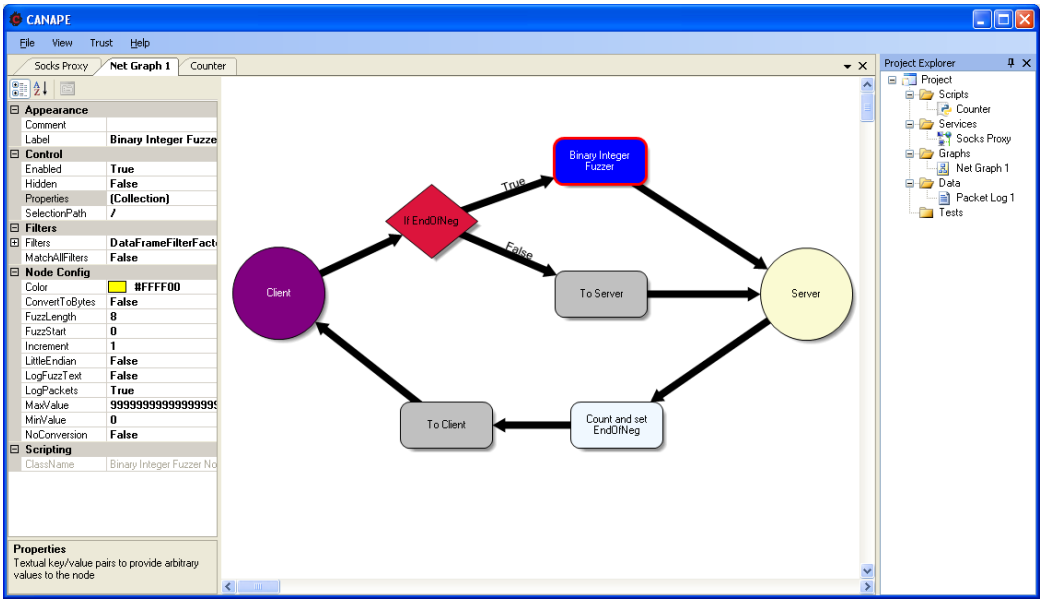

Figure 5.3: Canape - a binary network protocol testing tool acting as a proxy. The grey nodes are packet logging nodes.

in order to initiate the connection.

2. On the server–client path, the node *Count and set EndOfNeg* counts the number of packets.

3. Once 50 packets are counted, the *EndOfNeg* (end of negotiation) variable is set.

4. On the client–server path, an if statement checks if the negotiation is finished.

5. Once the negotiation has terminated, the packets sent by the client are altered by the *Binary Integer Fuzzer* node.

In order to automate the testing, a virtual machine has been set up with the AutoHotkey[3] software. Connection and disconnection attempts are regularly simulated and Canape modifies the packets in transit. Besides various error messages on the client side, no vulnerability has been identified.

[3] `http://www.autohotkey.com/`

5.2.2 Web application security

The web interface of the tested PLC is divided into two sections: one section where no authentication is necessary and one where the user needs to authenticate via basic HTTP authentication[4]. The communication is conducted over plain HTTP and not encrypted with TLS.

5.2.2.1 Missing authentication in web services

The web server of the Ethernet module includes Modbus/TCP capabilities. Furthermore, a dedicated web service accepts SOAP requests in order to issue read and write commands to the Modbus interface of the PLC. The web service does not support authentication and consequently leaves devices unprotected. A potential attacker can issue HTTP Post requests to the relative URL /ws/ModbusXmlDa with an appropriate SOAP message. Figure 5.4 displays how this can be used in order to read device identification.

5.2.2.2 Mitigation of web application issues

In order to secure the web service, several points need to be taken into consideration:

- The connection needs to be authenticated for each request. Therefore, the client needs to authenticate to the server. Authentication can take place with credentials, such as username and password. A better method of authentication would be via asymmetric key authentication. Mutual authentication, where the server (the PLC) is also authenticating to the client side is considered as essential.
- The connection needs to be confidential. Therefore, the exchanged messages should be encrypted for example with Transport Layer Security (TLS) [60]. For securing SOAP communication, a security mechanism called Web Service Security (WS-Security) has been presented by OASIS [44].
- The integrity of the exchanged messages needs to be verified with TLS or WS-Securtiy.

[4]HTTP Authentication: Basic and Digest Access Authentication: `https://www.ietf.org/rfc/rfc2617.txt`

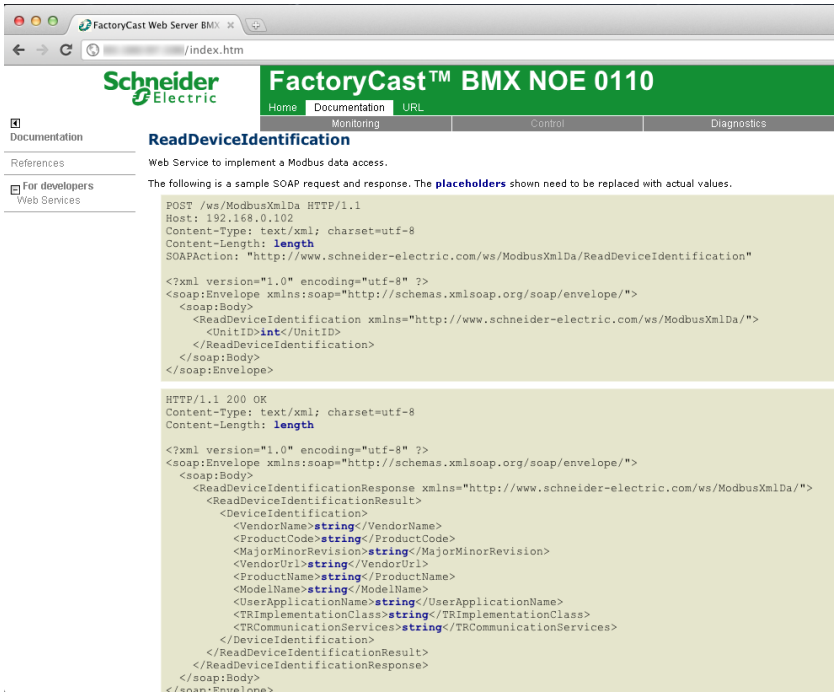

Figure 5.4: The BMX NOE 0110 web services Modbus Data Access does not require authentication.

5.2.2.3 Cross-Site Request Forgery for changing the password

Once a user is authenticated to the PLC web server, the username and password of the web application can be changed without knowledge of the old password. Furthermore, the web application does not have Cross-Site Request Forgery (CSRF) protection[5]. The password can be changed via a HTTP GET request. An attacker knowing the IP address of the PLC, can send to an *authenticated user* a link which will change the password to *NEWPASSWORD*:

`http://IP_OF_PLC/secure/embedded/builtin?Language=English&user=USER&passwd=NEWPASSWORD&cnfpasswd=NEWPASSWORD&subhttppwd=Change+Password`.

Once the password has been changed, the attacker can access the PLC completely.

For mitigating these kind of attacks, one might consider the following:

[5] `https://www.owasp.org/index.php/Cross-Site_Request_Forgery_%28CSRF%29`

1. In order to set a new password, the current password must be provided.
2. Cross-Site Request Forgery tokens need to be put into place, to avoid CSRF attacks.
3. The proper HTTP verb for modifying requests should be used. Not GET but POST is appropriate for changing a password.
4. IP-based access control could be added. Such a restriction cannot be seen as a strong security feature but it improves the defence.

Besides technical solutions, ICS engineers needs to be aware of potential security risks while clicking on links. The control computers should for example not be used for surfing on the Internet or reading emails.

The vulnerability cannot be exploited without prior knowledge. The attacker needs to be sufficiently sophisticated to trick an ICS engineer into clicking a link and the engineer needs to be authenticated at the same time to the PLC. Therefore, the impact of this vulnerability is limited.

5.2.3 Firmware analysis

Firmware updates for the PLC are available on the Schneider website. The downloads are provided without a hash or signature and, therefore, their integrity cannot be verified.

The downloaded firmware can be unzipped and contain several directories and files. The most important files are:

- /Firmware/WebServer.out (the web server image)
- /VxWorks_*.bin (the VxWorks operating system)
- /Web/wwwroot/classes/*.jar (Java applets for the web server)

5.2.3.1 Webserver

The web server file is an ELF binary [61] and seems to be based on the GoAhead web server. By analyzing the binary file with a disassembler, a lot of code is specific to the PLC usage. The binary contains debug information and function names can be recovered.

5.2.3.2 VxWorks Image

The VxWorks binary represents a customized VxWorks image and is based on VxWorks 6.4. The binary is not obfuscated or encrypted. Debug and version information can be found in the binary. Section 5.2.7 discusses security implications of the employed VxWorks version.

5.2.3.3 Java Applets

The web server supplies several Java applets which can be used to monitor the PLC through the web application. These Java applets are not obfuscated and existing decompilers are able to properly reverse-compile Java byte-code. Therefore, reverse engineering is easily accomplished with decompiler such as JD-GUI[6].

5.2.3.4 Static username and passwords

Ruben Santamarta performed extensive reverse engineering on the Schneider NOE 771 firmware and documented this in his blog [55]. The outcome of his work revealed, that several static usernames and passwords can be found in the firmware.

Nevertheless, although the available PLC is different from NOE 771, similar access credentials were found. For example the Java applets also contain the static username *sysdiag* and password *factorycast@schneider*. These credentials might give an attacker information about the system diagnosis.

5.2.4 Ladder logic upload and download

As explained in section 4.2.4, ladder logic is software executing on the operating system of a PLC. The available PLC can be programmed and controlled via a USB or Ethernet connection from a Windows control software. The communication is conducted over Modbus/TCP and mostly uses the undocumented Modbus function code 90.

[6] `http://java.decompiler.free.fr/?q=jdgui`

5.2.4.1 Programming of the PLC

Modbus does not support authentication and the control software does not require any authentication for programming the PLC. Therefore, an attacker with network access can reprogram without difficulty the PLC. Furthermore, the control software supports the functionality of transferring existing ladder logic from the PLC to the PC. Consequently, an attacker equipped with the control software can:

1. Transfer the ladder logic from the PLC to the PC
2. Alter the ladder logic
3. Upload the ladder logic back to the PLC

Reid Wightman from Digitalbond has published at the beginning of April 2012 a Metasploit module for transferring ladder logic between a PC and Modicon devices[7]. We were able to slightly modify the Metasploit module to make it functional for the Modicon M340.

5.2.4.2 Reliability of Human-Machine Interface (HMI)

The PLC is programmed with a specific programming language and displays informations on the HMI. In order to program the PLC, different languages can be employed:

- Structured text (ST): similar to C
- Function block diagram (FBD): graphical programming language where inputs, outputs and blocks are connected with lines
- Ladder diagram (LD): corresponding graphical diagram with relay logic hardware

Depending on the requirements of the PLC application, the corresponding language can be chosen. Furthermore, the available ICS devices are

[7] `http://dev.metasploit.com/redmine/projects/framework/repository/entry/modules/auxiliary/admin/scada/modicon_stux_transfer.rb`

shipped with eight different applications taking advantage of the available programming languages.

One application for instance is called *Speed control*. The application can be started by selecting the corresponding button on the HMI. Then, the motor starts rotating at a speed given by the potentiometer (figure in Annex A.2). Moreover, the rotation speed is limited from 0 to maximum 4000 rounds per minute (RPM). In order to measure the speed of the motor, two optical sensors are employed. One is measuring the rotation speed and the other is counting the rotations made by the motor.

If an attacker is able to program the PLC, he has complete control over the PLC. Therefore, the attacker would be able to alter the values reported to the HMI. Consequently, *the HMI cannot be considered a reliable source of information* either for the speed of the motor or for other values.

For testing purposes, we altered the speed control application in order to demonstrate this problem. Consequently, when executing the Speed control application, the motor turned at 2100 RPM instead of 100 RPM as displayed on the HMI.

5.2.4.3 HMI design improvements

With the current design choices of the PLC and programming environment, the HMI cannot be seen as a reliable source of information. In order to mitigate the risks associated with this issue, several options are possible:

- In order to upload the ladder logic to a PLC, authentication must be required. Furthermore, the person programming the ladder logic needs to be fully trusted. Moreover, the Windows control machine used to program the PLC needs to be fully trusted.
- If possible, the option of transferring ladder logic from the PLC to the PC should be disabled.
- For controlling purposes, a separate supervisory mechanism should be installed. A solution could be, for instance, an encoder measuring reliably the speed of a motor. In order to avoid tampering, this mechanism needs to be decoupled from the regular PLC architecture.

  Because of scaling issues, a redundant logging system might only cover parts of a complex ICS.

- An administrator could verify PLC programs and digitally sign these before they can be executed in production environments.
- If possible, the separation of user-privileges for PLC applications is recommended. A *superuser* would have full rights over the entire PLC functionality. On the other hand, a standard user might only be able to use parts of the PLC functionality.

Furthermore, the software system security principle called *least privilege* proclaims that each entity (a user or process) should have the most restrictive privileges which are needed in order to perform a given task [58]. This principle should be applied within ICSs.

5.2.5 Further ICS related issues

The following sections will focus on further ICS specific issues.

5.2.5.1 HMI communication

The communication between the PLC and the HMI is conducted over Modbus/TCP. Therefore, both the PLC and the HMI are connected via Ethernet to a switch.

The available PLC contains an application called *Slope Generation*. When this program is activated, the motor periodically accelerates until a given maximum speed, then decelerates to zero, and accelerates again (figure 5.5).

Figure 5.5: Slope generator periodically accelerates and decelerates the motor

Furthermore, a user can manually increase the speed of the motor instantly by pressing the +100 button of the HMI (indicated with an arrow in figure 5.5). When pressing this button, a Modbus/TCP request is sent to the PLC. Additionally, since Modbus does not support authentication, the PLC is interpreting every Modbus/TCP message it receives. Consequently, it is possible to replay packets issued by the HMI.

By using the Modbus/TCP Scapy library presented in chapter 4 Modbus/TCP packets can also be forged. The following listing (5.1) shows how to activate the slope control program.

Listing 5.1: Activating the slope control program

```
>>> connection = connectToTarget("169.254.0.2")
>>> connection.sr1(ModbusADU()/ModbusPDU10_Write_Multiple_Registers( \
        startingAddr=0x0000, quantityRegisters=0x0002, byteCount=0x04, \
        outputsValue=[0x00,0x01,0x00,0x00]))
```

Furthermore it is possible to simulate the pressing of the +100 RPM button. Additionally, it is possible to issue speed requests which are higher than +100 RPM. The following lines of code (listing 5.2) dictate an instantaneous motor speed of 4000 RPM. The output value of 0xfa0 corresponds to 4000 RPM.

Listing 5.2: Forged Modbus/TCP packet

```
>>> connection.sr1(ModbusADU()/ModbusPDU10_Write_Multiple_Registers( \
        startingAddr=0x0008, quantityRegisters=0x0001, byteCount=0x02, \
        outputsValue=[0xf,0xa0]))
```

Figure 5.6: Forged Modbus/TCP packet provokes that the motor to turn at a high speed (4000 RPM)

Besides lack of authentication, the PLC application is not properly sanitizing input. The request of listing 5.2 can be used to request a rotation-speed of 10000 RPM (0x2710). The Modbus request was accepted by the PLC and the motor accelerated. In order to not harm the device, the power was cut of during this test. Therefore, it has not been tested if the motor drive may limit the maximum rotation speed.

By issuing constantly zero-speed commands (listing 5.3) to the PLC, an attacker could also trigger a Denial of Service (DoS) attack.

Listing 5.3: Denial of Service by issuing zero-speed Modbus packets

```
>>> while 1: \
connection.sr1(ModbusADU()/ModbusPDU10_Write_Multiple_Registers( \
startingAddr=0x0008, quantityRegisters=0x0001, byteCount=0x02, \
outputsValue=[0x00,0x00]))
```

In order to mitigate these problems, proper input validation needs to be done in the PLC application.

5.2.5.2 Input Process Image and debugging features

Some PLC registers, collectively called the input process image, can be used to access the values that the PLC is reading from sensors. An example would be the temperature the PLC is reading from its temperature module. For the current device, the temperature can be read from a specific register which is part of the input process image (in Schneider products referred to with %I).

For debugging purposes, the Windows control software can alter the input values which the PLC reads from its sensors. Therefore, it is possible to force the PLC to report specific values back to the HMI. One example is to force, i.e. misrepresent, the temperature measured by the thermometer node. In figure 5.7, the temperature displayed on the HMI is 42 °C, whereas the actual temperature in the office is around 25 °C. Such functionality might be necessary in development environments, but should be disabled when the PLC is in productive use.

Figure 5.7: The temperature of 42 °C has been forced on the PLC. The actual ambient temperature is about 25 °C.

5.2.6 Control software security

PLCs can be programmed, monitored and operated through dedicated Windows control software. Within the scope of this thesis, the control software has not been analyzed.

5.2.7 Embedded operating system security

According to the FTP service banner and the firmware image, the operating system of the PLC is based on VxWorks 6.4. The current version of VxWorks 6.9 was released in February 2011. Furthermore, common vulnerability entries (CVE) exist for VxWorks 6.4:

1. CVE-2010-2967: *The loginDefaultEncrypt algorithm in loginLib in Wind River VxWorks before 6.9 does not properly support a large set of distinct possible passwords, which makes it easier for remote attackers to obtain access via a (1) telnet, (2) rlogin, or (3) FTP session. [64]*

Consequently it is recommended to update VxWorks to the latest version.

5.2.8 Undocumented features

Except the Modbus function code 90, no undocumented features or backdoors were identified on the PLC.

5.2.9 Firmware update procedure

In order to perform a firmware update the Windows control software can be used. The PLC can be updated either via USB or Ethernet connection.

To avoid an attacker reverse engineering the firmware, the firmware should be encrypted. The symmetric key for encryption needs to be stored securely somewhere in the device which is not an easy task. A trusted platform module could be used. Furthermore, having a shared master-key in all devices is not a good solution. Finally, the first firmware installed on the device needs to support the decryption routine.

5.2.9.1 Comments about the update procedure

The available PLC was shipped with an outdated firmware from 2008. In order to update to the latest firmware version two steps were necessary: The first was to upgrade the CPU to a new CPU version with a software update. Second was to apply the latest firmware update.

PLC software compiled for a specific CPU model is not compatible with other CPU models. Therefore, the PLC software needed to be reconfigured and recompiled manually, taking about a day of configuration work. Consequently, if the update procedure requires a CPU model change, the update needs to be planned carefully in advance. Otherwise, the Industrial Control System might not be operable for several hours.

5.2.9.2 Firmware distribution channel

Customers can download new firmware from the Schneider Electric web site. However, no hash or digital signature of the firmware is provided. Therefore, it is not possible to verify that a genuine firmware has been downloaded. Furthermore, an attacker might be able to compromise the Schneider website and distribute malicious firmware to customers. Consequently, the distribution channel for new firmware needs to be secured with HTTPS and a hash or digital signature of the firmwares should be provided.

5.2.10 Further observations

In the following section, further security related observations concerning the analysed ICS devices are presented.

Username and password for the web interface are stored on the FTP server of the PLC. The default username and password are stored in /SDCA/Web/rdt/password.rde. Moreover, once the default password is changed, the new password is stored in /SDCA/Web/userlist.dat. Reid Wightman published a Metasploit module which can be used to recover the password for Schneider Modicon PLCs [53].

Chapter 6

Existing security solutions for ICSs

The following paragraphs will describe briefly two existing security mechanisms in order to secure Industrial Control Systems. These solutions focus on securing the connection between the PLC and the controlling PC (the SCADA part of an ICS). Therefore, they protect against several threats, but cannot be seen as the ultimate solution.

Within the scope of the thesis, the focus is kept on HIP (Host Identity Protocol) and OPC UA (OLE for process control Unified Architecture).

6.1 Host Identity Protocol (HIP)

It is not the aim of this chapter to explain the details of HIP [40] but rather to give a short introduction and then analyze its usefulness within ICSs.

In the current Internet architecture, location and identity of the communicating parties are both based on the Internet Protocol (IP) Address. The Host-Identity protocol (HIP) proposes a different approach by introducing Host Identifiers (HIs) [51]. Technically, HIs are the public key of a private and public key-pair.

The principal idea behind HIP is to separate the location from the identity: The HI identifies a network participant uniquely the IP address represents the location. Furthermore, HIP introduces a supplementary *Host Identity* layer in the Network Stack. Benefits of HIP are for instance mobility support [2], transparent security or multi-homing.

6.1.1 Boeing SCADAnet

One concrete example where HIP is employed in production facilities is at Boeing. In order to meet production requirements and to ensure a specific production capacity Boeing adopted the *moving factory line* for their airplane factories. Therefore, the plane is constantly moving during production. Consequently, the different equipment and production units cannot be attached over wired connections and need to communicate over a wireless communication.

Wireless communications can be eavesdropped easily if the communication is not encrypted. Thus, the airplane crafting company implemented the so-called SCADAnet factory (see figure 6.1) with the help of HIP and the Secure Mobile Architecture (SMA) [54]. The aim was to especially find a solution to secure mobile wireless communications, based on the current Internet protocol suite.

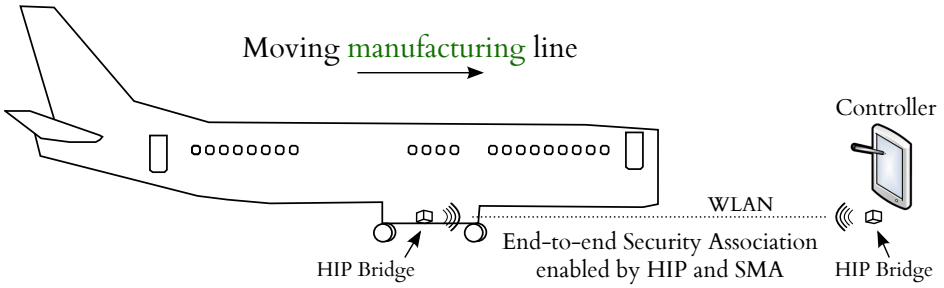

Figure 6.1: Boeing moving factory line for airplanes. The wireless communication is secured through HIP.

As can be seen in figure 6.1, one mobile HIP bridge is installed in the airplane in the factory, and one HIP bridge on the controlling side. The HIP bridges create an overlay network which can be used to communicate securely. Each bridge contains a SIM chip in order to perform mutual authentication. Furthermore, the bridges are able to react in real-time to events such as a production failure or reaching a physical position [54].

In addition to strong encryption and authentication, the Boeing HIP brides are not difficult to deploy. Therefore, untrained employees can plug in one bridge on an Ethernet port in the PLC near the plane and one bridge at the Ethernet port of the corresponding controller.

Besides 802.11 networks, the bridges can communicate over any IP-based network. Consequently, the HIP bridges can be used for securing ICS con-

nections through the Internet.

6.1.2 Tofino

Tofino is a brand-name of the Byres Security Inc. and offers specialized hardware to improve the security of ICSs. Tofino's solution has end-boxes comparable to the bridges of Boeing. They are based on OpenHIP and implement a Virtual Private Network (VPN) in order to secure the communication between the PLC and the controller. Similar to Boeing, a Tofino end-box also supports mutual authentication through smart cards [62] and is proclaimed to be deployed with Plug-n-Protect^{TM} capabilities.

Tofino advertises that their products can mitigate threats like Stuxnet. Since Stuxnet has spread over the network [59] in order to infect further machines, this is partially true. Nevertheless, Tofino devices, or VPN end-boxes in general cannot mitigate every threat that Stuxnet exposes them to. If a controlling machine uses one network interface to communicate securely over HIP to the PLC and another network interface to communicate within the control system network, Stuxnet can still infect the control machine. Furthermore the control machine can be infected through USB devices. Once the control machine is infected, even though the communication to the PLC is secured, Stuxnet can maliciously alter the PLC.

6.2 OLE for process control (OPC)

OPC by the OPC Foundation[45] *a series of standards specifications* which has been mainly implemented on Windows devices. The first OPC standard, called Data Access Specification (formerly known as OLE for process control) has been established by ICS manufacturers and Microsoft. The specification provides interfaces and methods which can be employed in ICSs.

According to the OPC Foundation, the most suitable comparison in order to understand OPC DA, are Microsoft Windows printer drivers [45]: Before Microsoft implemented the concept of system-wide printer drivers, each application developer had to implement for each program individually the necessary printer drivers. Nowadays, the printer drivers are designed and implemented by the printer manufacturer. Once installed in Windows, every application is able to use these drivers in order to send print commands to the

printer. OCP Data Access is introducing in a similar manner a framework which can be employed by all industrial control manufacturers. Therefore, it is easier to conduct interconnectivity between different vendors and software vendors can build their industrial control software on the top of OPC.

The OPC Data Access is mainly used in order to transfer real-time data from PLCs to control devices such as HMI [45].

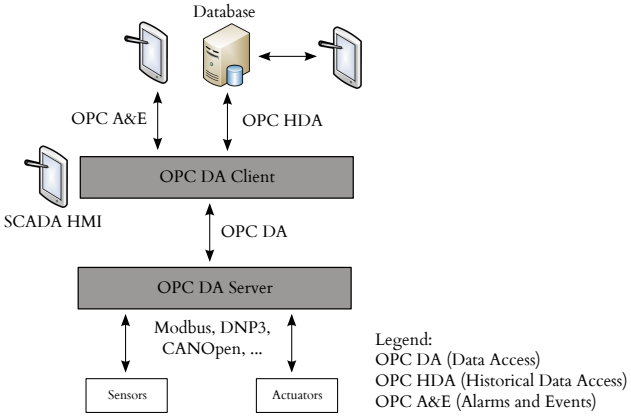

Figure 6.2: Classical OPC architecture

The first OPC Data Access specification was released in 1999[1]. Moreover, security was not one of the main concerns at that time. Nevertheless, nowadays the demand for security is constantly rising and ICS vendors demanded for Microsoft-independent OPC implementations. Consequently, the OPC Foundation has created a new set of specifications called OPC UA (Unified Architecture) which no longer relies on Microsoft COM technology.

6.2.1 OPC Unified Architecture (UA)

In OPC UA [20], security is one of the main built-in components. The OPC UA security model offers user authentication and authorization on the application layer; application authentication, message integrity and confidentiality on the communication layer; and optionally also confidentiality and integrity on the transport layer.

The OPC UA services use for instance a Public-Key Infrastructure (PKI)

[1] `http://www.opcfoundation.org/DownloadFile.aspx?CM=3&RI=68&CN=KEY&CI=274&CU=16`

in order to exchange and validate certificates[2]. Furthermore, OPC UA can be operated in two modes: through a binary protocol (opc.tcp://) and through a web service (http://). The binary mode is faster than the web service mode and should be used for performance critical applications [46].

OPC UA is built with security in mind and can properly secure the connection between the PLCs and the HMI/SCADA devices. It represents an important step towards more secured ICSs. Nevertheless, OPC UA only protects ICSs against a subset of the existing threats. Malware like Stuxnet may infect computers in the control rooms of automation systems. Once these control stations are infected with the malware, the OPC UA architecture protection for the PLCs is limited.

Moreover, OPC has been released in 2006 and, until to date, the Aalto Automation Department estimates that only about 10% of the worldwide OPC architectures have been upgraded to OPC UA. Furthermore, OPC UA can be operated without activated security. Consequently, ICS engineers need to be aware of the necessity of updating the existing architecture and should also enable the security features.

[2] `http://www.opcfoundation.org/DownloadFile.aspx?CM=3&RI=457&CN=KEY&CI=287&CU=59`

Chapter 7

Discussion

The presented security analysis of Industrial Control System devices shows that ICSs are not sufficiently secure against malicious intruders. This chapter elaborates the implications of insecure Industrial Control Systems. Furthermore, the presented ICS insecurities cannot be fixed without difficulty. Replacing insecure ICS protocols such as Modbus/TCP implies significant changes to many different devices and softwares. Moreover, the long life cycle of ICSs increases the significance of present vulnerabilities.

This chapter is divided into three sections: First, the implications of the vulnerable ICSs will be explained. Secondly, the accuracy of the analysis guidelines will be discussed. Finally, a brief evaluation of the methodology employed in this thesis will be performed.

7.1 Implications of insecure Industrial Control Systems

The presented weaknesses of ICS devices are real and urgent. Vulnerable devices can be found without difficulty[1]. Nearly 200 Schneider Electric PLCs can be found through HTTP Header indexing search engines. Unfortunately, some vendors take a year or more to release patches to known vulnerabilities. In some cases, the vendors decide to not patch their systems because the life

[1]By scanning the Internet or using services like shodanhq.com

cycle of affected devices has ceased[2]. On the other hand, sometimes vendors respond quickly: In December 2012, Schneider Electric responded within 9 days to an ICS-CERT alert with a patch for their products.

Nonetheless, the majority of Industrial Control Systems stays insecure. The principal reasons of this situation are constituted by wrong design decisions taken several years ago. Furthermore, ICS vendors were and are ignoring security issues by proclaiming that devices not connected to the Internet are secure. Once an attacker has access to a local area network inside a control system, installing backdoors is possible. Malware such as Stuxnet have effectively demonstrated this issue [19].

During the year 2010 the ICS-CERT published about 39 alerts, in 2011 around 145, and in 2012 there are currently 76 alerts disclosed[3], affecting more than 50 vendors. According to McBrides [57], 215 ICS vulnerabilities have been disclosed within 2011. Taking into account the number of vulnerabilities published in the recent months, a series of implications can be deduced. The first implication is that independent security researchers are increasingly conducting security analysis of ICS components such as firmware, software and hardware. Second, not only one vendor is affected by the security problems but rather the whole ICS industry. Third, the number of unreported vulnerabilities is unknown and probably higher.

Mitigating these problems will take several product cycles and, therefore, depending on the effort of the ICS vendors, may take years. However, since the ICS industry is increasingly introducing IT technology into the ICSs, IT and ICS are merging technologies. Consequently, because IT life cycles are relatively short, it is probable that the life cycle of ICS devices will be shortened significantly.

7.2 Accuracy of analysis guidelines

Chapter 4 presented a set of generic analysis guidelines for ICS devices such as PLCs. These guidelines are not tied to any specific vendor and can be applicable to any ICS device. However, since every ICS device has its specific features, in some cases the guidelines may need to be adapted in order to

[2]ABB will not patch products affected in this alert, although they might allow remote code execution (`https://www.us-cert.gov/control_systems/pdf/ICSA-12-095-01.pdf`)

[3]18 June 2012

take into account individual characteristics of ICS devices.

Furthermore, the guidelines focus on the industrial protocol Modbus/TCP. Modbus/TCP is one of the most widely employed industrial protocols. Nevertheless, other protocols exist: PROFINET, Ethernet IP, the Common Industrial Protocol (CIP) or even CAN, which are not covered by the presented guidelines. More extensive analysis guidelines could cover these protocols.

7.3 Evaluation of the methodology

The presented security analysis of ICS has first been conducted on a very generic level in the background part (chapter 2) and threat analysis (chapter 3). Especially the guidelines for the security analysis of ICS devices (chapter 4) have been kept generic. This choice of methodology can be justified since the analysis methodology could be applied to different Industrial Control Systems and devices regardless their specific features.

The threat analysis covered only a small part of a real Industrial Control System. A complete threat analysis would have taken a tremendous amount of time and effort and was not in the scope of the thesis. Other threat evaluation methodologies might be more appropriate, depending on the complexity of the ICS. Furthermore, by analyzing more ICS devices, it might come out that the analysis guidelines need to be extended. However, the presented guidelines enable an engineer to get a quick overview of the security of the analyzed ICS and can reveal vital security issues.

Finally, the combination of high- and low-level ICS analysis gives an extended view of ICS security on different levels. If the analysis had been focused only on the high-level problems, the evaluation results might have been less credible.

Chapter 8

Conclusion and further work

Fortunately, the awareness of ICS insecurity is rising. The recent report from the European Network and Information Security Agency (ENISA) [16] is only one example of this. Nevertheless, ICS vendors are still reacting too slow and need to take security more seriously.

Several aspects of Industrial Control Systems have been addressed within this thesis. The main contributions of this thesis are:

1. A background part explaining ICSs to IT engineers.
2. A high-level threat analysis of an ICS architecture by employing the Microsoft STRIDE threat modeling methodology from software engineering.
3. Development of an Modbus/TCP Scapy library for testing ICS devices and infrastructure.
4. Security analysis guidelines for ICS devices.
5. Security analysis of ICS devices.
6. Proposed mitigation techniques for the discussed security problems and existing industrial security solutions.

Engineers who are not familiar with ICSs can learn from the background part of the present thesis the typical characteristics of Industrial Control Systems. Besides explaining terminology, a brief overview of ICS history

helps to understand their fragility especially once connected to open networks such as the Internet. The most notable problem regarding ICS architectures consists of installed backdoors by the manufacturers, which unfortunately seems to be common practice in ICS. The background part does not require any former knowledge of ICSs and is addressed to IT engineers.

The following threat analysis elaborates on the different risk factors for Industrial Control Systems. The use of commercial off-the-shelf products is for instance significantly increasing the threat surface of ICSs. We provide a threat analysis with the STRIDE model. The threat analysis can help ICS engineers to assess their infrastructure from a high-level perspective.

Before conducting a security analysis of a real ICS devices, the thesis presents guidelines for how to assess an ICS devices from a security perspective. The guidelines are kept on a general level and can be applied to different devices from different vendors. In the year 2011, alerts from the ICS-CERT covered 50 ICS vendors and, therefore, it can be expected that many ICS products have security issues. Many such vulnerabilities could be found by following the presented guidelines.

We developed a Modbus/TCP library for Scapy a framework used by professional security testers. It can be considered a powerful library for packet manipulation. Consequently, it can be reemployed for robustness or custom network testing of Industrial Control Systems. The security analysis we performed has revealed several weaknesses of the ICS devices in the experiments. For the reason of responsible disclosure, not all the details have been presented in this thesis.

An overview of existing security solutions shows examples on how to protect ICS. We also presented several mitigations against the found security problems, but those were not the focus of this thesis.

In conclusion, the ICS industry has only recently become aware of security-related risks. ICS customers will demand the ICS manufacturers to improve their security. Furthermore, security is now a critical sales point when choosing an ICS manufacturer. Finally, the safety of Industrial Control Systems cannot be guaranteed without properly designed and evaluated security measures.

8.1 Future work

For further studies, the analysis of existing ICS architectures in production should be considered. Furthermore, different devices from several vendors could be compared. Besides searching for security vulnerabilities, mitigation techniques and recommendations need to be specified. Further studies should implement and verify the effectiveness of the proposed mitigations.

One interesting study could be analyzing whether the public Modbus/TCP function codes can be used to implement authentication subsequent Modbus requests.

Finally, the Modbus/TCP Scapy library presented in this thesis can be extended to improve its functionality.

Bibliography

[1] A. Biyani, G. Sharma, J. Aghav, P. Waradpande, P. Savaji, M. Gautam. Extension of SPIKE for Encrypted Protocol Fuzzing. Third International Conference on Multimedia Information Networking and Security (MINES), 2011.

[2] A. Leonardo, H. Chaouchi. Host Identity Protocol Proactive Mobility Management Experimentation. International Conference on Telecommunications (AICT), 2010 Sixth Advanced.

[3] Allen-Bradley Company. Programmable logic controller, United States Patent, March 1976. Patent 3,942,158 `http://www.patents.com/us-3942158.html`.

[4] American National Standard (ANSI) and International Society of Automation (ISA). ANSI/ISA99.00.012007 Security for Industrial Automation and Control Systems, 2007. Part 1 Terminology, Concepts, and Models.

[5] B. Atlagic, D. Milinkov, M. Sagi, B. Bogovac. High-Performance Networked SCADA Architecture for Safety-Critical Systems. 2nd Eastern European Regional Conference on the Engineering of Computer Based Systems (ECBS-EERC), 2011.

[6] Charles G. Oakes, PhD. Safety versus security in fire protection planning. Patent 3,942,158 `http://www.aia.org/practicing/groups/kc/AIAB079791`.

[7] Cohen, F. Automated Control System Security. Security and Privacy, IEEE.

[8] Dao-gang Peng, Hao Zhang, Li Yang, Hui Li. Design and Realization of Modbus Protocol Based on Embedded Linux System. In-

ternational Conference on Embedded Software and Systems Symposia, 2008. ICESS Symposia '08.

[9] David E. Sanger. Obama Order Sped Up Wave of Cyberattacks Against Iran. New York Times, http://www.nytimes.com/2012/06/01/world/middleeast/obama-ordered-wave-of-cyberattacks-against-iran.html?pagewanted=2&_r=1&seid=auto&smid=tw-nytimespolitics&pagewanted=all, Accessed 01 June 2012.

[10] Department of Homeland Security, USA. Recommended practice: Improving industrial control systems cybersecurity with defense-in-depth strategies. http://www.us-cert.gov/control_systems/practices/documents/Defense_in_Depth_Oct09.pdf.

[11] DNP Users Group. 1815-2010 IEEE Standard for Electric Power Systems Communications – Distributed Network Protocol (DNP3). http://ieeexplore.ieee.org/servlet/opac?punumber=5518535, Accessed 31 Jan 201.

[12] D.P. Fidler. Was Stuxnet an Act of War? Decoding a Cyberattack. Security and Privacy, IEEE.

[13] Edward W. Kamen. Industrial Controls and Manufacturing. Academic Press, 1999, ISBN 0123948509, Chapter 8 Ladder Logic Diagrams and PLC Implementations.

[14] Eireann P. Leverett. Quantitatively Assessing and Visualising Industrial System Attack Surfaces, June 2011. http://www.cl.cam.ac.uk/~fms27/papers/2011-Leverett-industrial.pdf.

[15] EtherCAT Technology Group. Ethercat - ethernet for control automation technology. http://www.ethercat.org/en/ethercat.html, Accessed 12 Jan 2012.

[16] European Network and Information Security Agency (ENSIA). Protecting Industrial Control Systems, December 2011. http://www.enisa.europa.eu/activities/res/other-areas/ics-scada/protecting-industrial-control-systems.-recommendations-for-europe-and-member-states.

[17] European Network and Information Security Agency (ENSIA). Protecting industrial control systems. annex 1: Desktop research

results, December 2011. http://www.enisa.europa.eu/activities/res/other-areas/ics-scada/annex-i.

[18] F-Secure. Virus:boot/brain. http://www.f-secure.com/v-descs/brain.shtml, Accessed 31 Jan 2012.

[19] F-Secure. Stuxnet redux: Questions and answers, 2010. http://www.f-secure.com/weblog/archives/00002066.html, Accessed 22 Jan 2012.

[20] Huang Renjie, Liu Feng, Pan Dongbo. Research on OPC UA security. The 5th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2010.

[21] Hyoungchun Kim, Younghan Choi, Dohoon Lee, Donghoon Lee. Practical Security Testing using File Fuzzing. Advanced Communication Technology, 2008. ICACT 2008.

[22] IBM Global Services. A strategic approach to protecting scada and process control systems, 2007.

[23] IEEE. 1686-2007 IEEE Standard for Substation Intelligent Electronic Devices (IEDs) Cyber Security Capabilities. http://ieeexplore.ieee.org/servlet/opac?punumber=4453837.

[24] International Organization for Standardization. Information technology – security techniques – information security management systems – requirements. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=42103, Accessed 31 Jan 2012.

[25] International Organization for Standardization. Information technology – security techniques – information security management systems – requirements. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=50297, Accessed 31 Jan 2012.

[26] ISA99 Committee. ISA99 Committee on Industrial Automation and Control Systems Security. http://isa99.isa.org/ISA99%20Wiki/Home.aspx, Accessed 31 Jan 2012.

[27] Kanich, Weaver McCoy et al. Show Me the Money: Characterizing Spam-advertised Revenue. http://cseweb.ucsd.edu/~savage/papers/UsenixSec11-SMTM.pdf.

[28] Luigi Auriemma. A strategic approach to protecting scada and process control systems, 2007. http://aluigi.altervista.org/, Accessed 12 Jan 2012.

[29] Magnus Sundell, Janne Kuivalainen, Juhani Mäkelä, Arthur Gervais, Jouko Orava, Mikko H. Hyppönen. White paper on industrial automation security in fieldbus and field device level, December 2011. http://www.vacon.com/Vacon-White-Paper-On-Industrial-Automation-Security-In-Fieldbus-And-Field-Device-pdf.

[30] Mai Kiuchi and Yoshizumi Serizawa. Security Technologies, Usage and Guidelines in SCADA System Networks, 2009. ICROS-SICE International Joint Conference 2009, August 18-21, 2009, Fukuoka International Congress Center, Japan, Part 1 Terminology, Concepts, and Models.

[31] Michael Sutton, Adam Greene, Pedram Amini. Fuzzing: Brute Force Vulnerability Discovery. ISBN-10: 0321446119, http://www.fuzzing.org, Accessed 15 Feb 2012.

[32] Microsoft. Microsoft security development lifecycle. http://www.microsoft.com/security/sdl/default.aspx, Accessed 31 Jan 2012.

[33] Microsoft. Improving web application security: Chapter 3 threat modeling, 2003. http://msdn.microsoft.com/en-us/library/ff648644.aspx, Accessed 24 Jan 2012.

[34] Microsoft. Threat modeling - Uncover Security Design Flaws Using The STRIDE Approach, 2006. http://msdn.microsoft.com/en-us/magazine/cc163519.aspx, Accessed 24 Jan 2012.

[35] Microsoft. IT Infrastructure Threat Modeling Guide, 2009. http://technet.microsoft.com/en-us/library/dd941826.aspx, Accessed 24 Jan 2012.

[36] Mikko Hypponen. Mikko hypponen: Three types of online attack. http://www.ted.com/talks/mikko_hypponen_three_types_of_online_attack.html, Accessed 31 Jan 2012.

[37] Modbus Organization. Modbus application protocol specification v1.1b. http://modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf, Accessed 31 Jan 2012.

[38] Modbus Organization. Modbus Organization Members. http://modbus.org/about.php.

[39] Motorola. SCADA Systems. http://www.motorola.com/web/Business/Products/SCADA%20Products/_Documents/Static%20Files/SCADA_Sys_Wht_Ppr-2a_New.pdf, Accessed 18 May 2012.

[40] Muslam, M. Muhana, H. Chan, Anthony, Magagula, A. Linoh, Ventura, Neco. Network-based mobility and Host Identity Protocol. Wireless Communications and Networking Conference (WCNC), 2012 IEEE.

[41] National Communications System, USA. Supervisory Control and Data Acquisition (SCADA) Systems, October 2004. http://www.ncs.gov/library/tech_bulletins/2004/tib_04-1.pdf.

[42] National Institute of Standards and Technology (NIST). NIST SP 800-82: Guide to Industrial Control Systems (ICS) Security, 2011. csrc.nist.gov/publications/nistpubs/800-82/SP800-82-final.pdf.

[43] NERC - North American Electric Reliability Corporation. CIP - Critical Infrastructure Protection. http://www.nerc.com/page.php?cid=2%7C20, Accessed 31 Jan 2012.

[44] OASIS - Organization for the Advancement of Structured Information Standards. Web Services Security: 3 SOAP Message Security. Working Draft 13, Thursday, 01 May 2003, http://www.oasis-open.org/committees/download.php/2314/WSS-SOAPMessageSecurity-13-050103-merged.pdf.

[45] OPC Foundation. What is OPC, 2011. http://www.opcfoundation.org/Default.aspx/01_about/01_whatis.asp?MID=AboutOPC, Accessed 31 Jan 2012.

[46] OPC Foundation. What is OPC UA, 2011. http://www.opcfoundation.org/Default.aspx/01_about/UA.asp?MID=AboutOPC, Accessed 31 Jan 2012.

[47] Philippe Biondi. Scapy, python interactive packet manipulation framework. http://www.secdev.org/projects/scapy/, Accessed 31 Jan 2012.

[48] Profibus International (PI). Profibus overview. http://www.profibus.com/technology/profibus/, http://www.kuebler.com/PDFs/Feldbus_Multiturn/specification_DP.pdf, Accessed 31 Jan 2012.

[49] R. Johari, P. Sharma. A Survey on Web Application Vulnerabilities (SQLIA, XSS) Exploitation and Security Engine for SQL Injection. Conference on Communication Systems and Network Technologies (CSNT), 2012 International.

[50] R. Langner. Stuxnet: Dissecting a Cyberwarfare Weapon. Security and Privacy, IEEE.

[51] R. Moskowitz and P. Nikander. RFC 4423: Host identity protocol (hip) architecture, May 2006. ICSA Labs, a division of Cybertrust, Inc. and Ericsson Research Nomadic Lab, Status: INFORMATIONAL, https://tools.ietf.org/html/rfc4423.

[52] Ralph Langner. Cracking Stuxnet, a 21st-century cyber weapon. http://www.ted.com/talks/lang/en/ralph_langner_cracking_stuxnet_a_21st_century_cyberweapon.html, Accessed 31 Jan 2012.

[53] Reid Wightman. Schneider modicon quantum. http://www.digitalbond.com/tools/basecamp/schneider-modicon-quantum/, Accessed 12 Apr 2012.

[54] Richard Paine. Beyond HIP. The end to hackers as we know it, 2009. ISBN: 1-4392-5604-7.

[55] Ruben Santamarta. Reversing Industrial firmware for fun and backdoors I. http://www.reversemode.com/index.php?option=com_content&task=view&id=80&Itemid=1, Accessed 01 June 2012.

[56] S. Bekrar, C. Bekrar, R. Groz, L. Mounier. Finding Software Vulnerabilities by Smart Fuzzing. IEEE Fourth International Conference on Software Testing, Verification and Validation (ICST), 2011.

[57] Sean McBrides. Documenting The Lost Decade ICS Vuln Analysis. http://www.digitalbond.com/2012/01/30/documenting-the-lost-decade-ics-vuln-analysis/.

[58] Software Assurance Workforce Education and Training Group. Software Assurance: A Curriculum Guide to the Common Body of Knowledge to Produce, Acquire and Sustain Secure Software, 2007. US Department of Homeland Security.

[59] Symantec: Nicolas Falliere, Liam O Murchu, Eric Chien. W32.stuxnet dossier, February 2011. Version 1.4, `http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf`.

[60] T. Dierks, C. Allen, Certicom. RFC 2246: The TLS Protocol, January 1999. Network Working Group, `http://www.ietf.org/rfc/rfc2246`.

[61] TIS Committee. Tool Interface Standard (TIS) Executable and Linking Format (ELF) Specification, May 1995. `http://refspecs.freestandards.org/elf/elf.pdf`.

[62] Tofino. Tofino endbox, pre-release documentation, December 2009.

[63] US-CERT Vulnerability Note VU Nb.362332. Wind River Systems VxWorks debug service enabled by default. `http://www.kb.cert.org/vuls/id/362332`.

[64] US-CERT Vulnerability Note VU Nb.840249. Wind River Systems VxWorks weak default hashing algorithm in standard authentication API (loginLib). `http://www.kb.cert.org/vuls/id/840249`.

[65] Wireshark Foundation. Wireshark. `http://www.wireshark.org/`.

[66] Xiao-Song Zhang, Lin Shao, Jiong Zheng. A Novel Method of Software Vulnerability Detection based on Fuzzing Technique. Apperceiving Computing and Intelligence Analysis, 2008. ICACIA 2008.

[67] Zhimin Yang, Zengguang Zhang. The Study on Resolutions of STRIDE Threat Model. First IEEE International Symposium on Information Technologies and Applications in Education, 2007. ISITAE 07.

Appendix A

First appendix

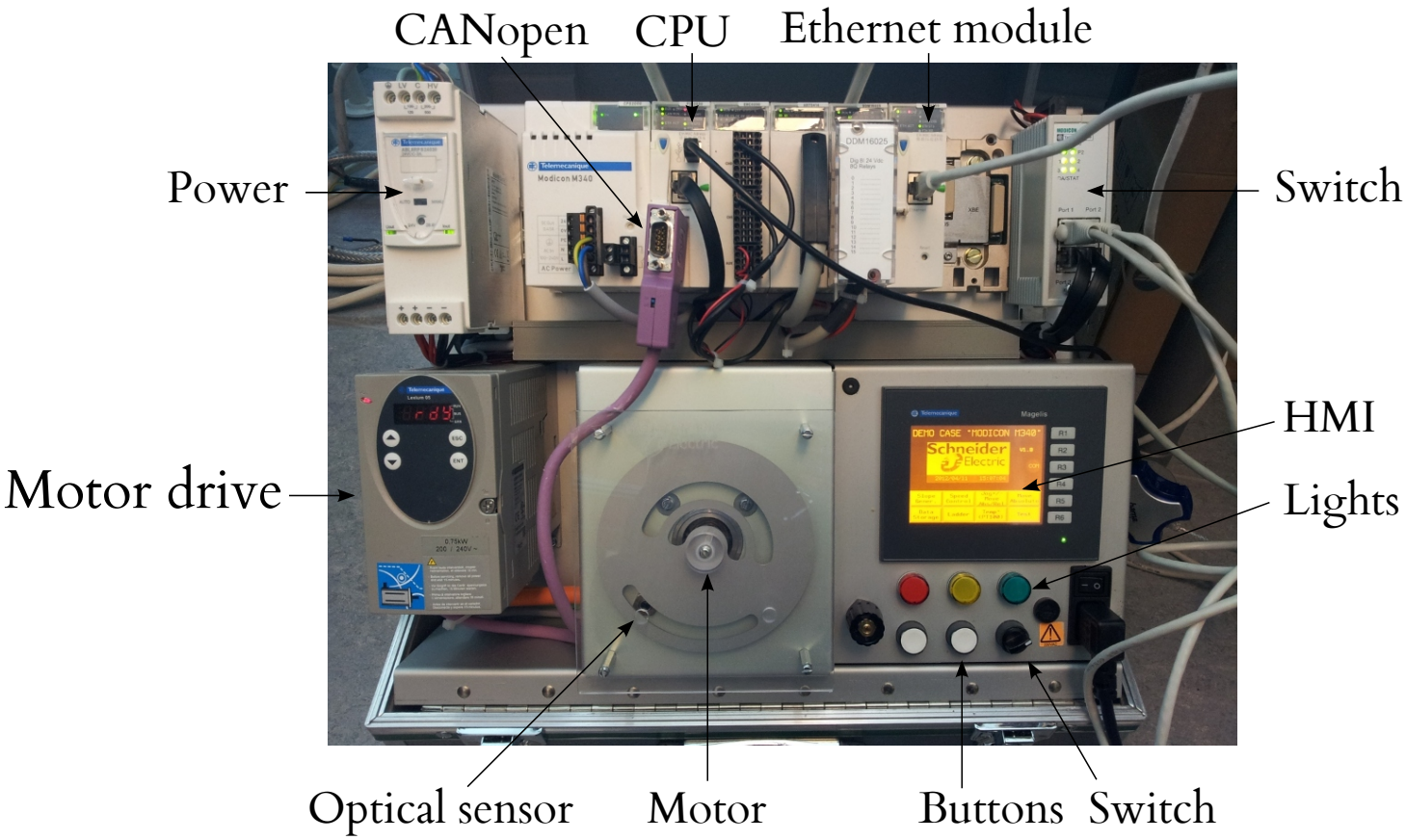

Figure A.1: ICS devices provided by Schneider Electric. With Modicon M340, P342030 CPU and BMX NOE 0100

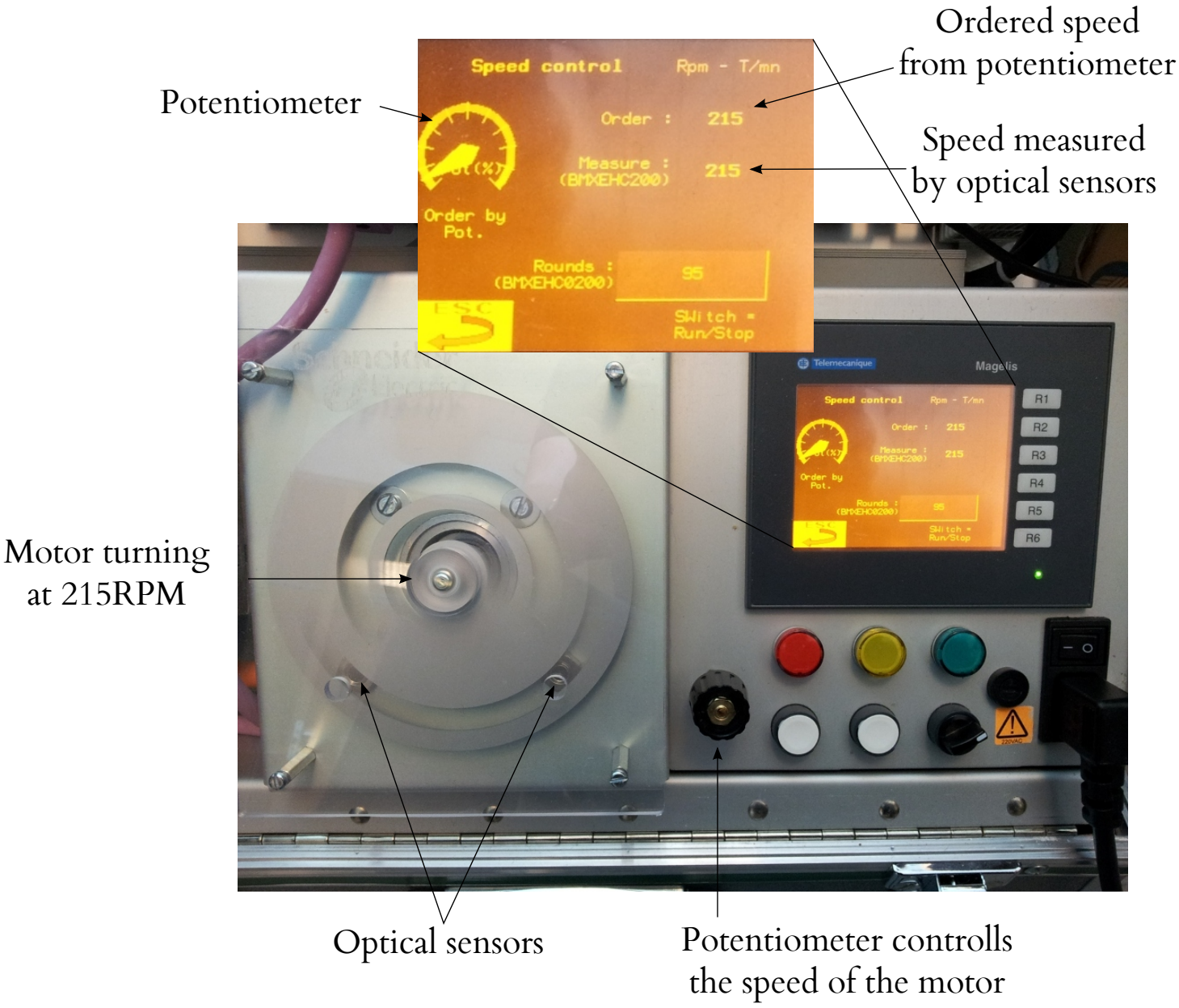

Figure A.2: "Speed control" program running. The speed of the motor can be controlled with the potentiometer.