

# WHITE PAPER ON INDUSTRIAL AUTOMATION SECURITY IN FIELDBUS AND FIELD DEVICE LEVEL

**Authors:**

Magnus Sundell, Vacon Plc, [magnus.sundell@vacon.com](mailto:magnus.sundell@vacon.com)

Janne Kuivalainen, Vacon Plc, [janne.kuivalainen@vacon.com](mailto:janne.kuivalainen@vacon.com)

Juhani Mäkelä, Nixu Ltd, [juhani.makela@nixu.com](mailto:juhani.makela@nixu.com)

Arthur Gervais, Nixu Ltd, [arthur.gervais@nixu.com](mailto:arthur.gervais@nixu.com)

Jouko Orava, Vacon Plc, [jouko.orava@vacon.com](mailto:jouko.orava@vacon.com)

Mikko H. Hyppönen, F-Secure Corporation, [mikko.hypponen@f-secure.com](mailto:mikko.hypponen@f-secure.com)

## **Abstract**

There has been a lot of discussion about malware and security in industrial automation systems after Stuxnet. This white paper is based on material from the public domain and focuses on presenting a generic overview about security in industrial automation on the fieldbus and device level.

The level of standardization in the information security field is presented, comparing the status of ICT systems' security standardization to that of industrial automation.

Security aspects of traditional fieldbuses, Ethernet-based networks and wireless communication technologies are presented. Challenges regarding data security in the field of industrial automation are discussed. The properties of industrial automation devices are described with a focus on security, tampering possibilities, and risk mitigation methods.

Index terms – security, industrial automation, fieldbus, industrial Ethernet, wireless communication, embedded devices, standardization, Stuxnet

## Table of Contents

1	Introduction and scope.....	6
1.1	Industrial automation systems overview .....	6
1.2	Types of malware.....	6
1.3	Current malware status concerning the industrial automation sector.....	7
1.4	Standardization and related organizations .....	8
1.4.1	ICT security standards .....	8
1.4.2	Industrial automation standards .....	9
1.4.3	Other industrial automation security related organizations and standards.....	11
1.4.4	Standardization summary .....	11
2	Generic security considerations.....	11
2.1	Attacks and scenarios .....	12
2.2	Security program.....	12
3	Security in communication between devices .....	13
3.1	Purpose of communication .....	13
3.2	Security threats and issues .....	14
3.2.1	Reconnaissance activity.....	15
3.2.2	Attacks on communication .....	16
3.3	Traditional fieldbuses .....	17
3.3.1	Modbus RTU.....	18
3.3.2	PROFIBUS DP.....	19
3.3.3	CANopen .....	22
3.3.4	DeviceNet .....	23
3.4	Ethernet networks .....	23
3.4.1	Ethernet physical layer .....	23
3.4.2	Ethernet data link layer.....	24

3.4.3	Internet Protocol.....	24
3.4.4	Transport layer .....	25
3.4.5	Network configuration.....	25
3.4.6	Network topology.....	26
3.4.7	Industrial Ethernet protocols .....	27
3.5	Recommendations for enhancing security.....	31
4	Security in field devices .....	32
4.1	Security threats and issues .....	32
4.1.1	Information leakage.....	32
4.1.2	Tampering risks.....	32
4.2	Simple field devices .....	33
4.3	Embedded devices with real-time operating systems.....	33
4.4	Embedded devices with general-purpose operating systems.....	34
4.4.1	Operating system vulnerabilities .....	34
4.4.2	Open-source systems .....	34
4.4.3	General-purpose operating systems.....	34
4.5	Recommendations for enhancing security in devices .....	35
4.5.1	Debugging interfaces .....	35
4.5.2	Communication interfaces .....	35
4.5.3	Firmware protection .....	36
4.5.4	Device parameters and configuration .....	37
4.5.5	Firmware updating .....	37
4.5.6	Superfluous information .....	37
5	Security in wireless communications.....	38
5.1	Security of wireless technology.....	38
5.1.1	IEEE 802.15.4.....	39
5.1.2	Wireless LAN .....	39

5.1.3	Bluetooth.....	39
5.2	Recommendations for improving wireless network security.....	40
6	Summary .....	41
7	References .....	42

## 1 Introduction and scope

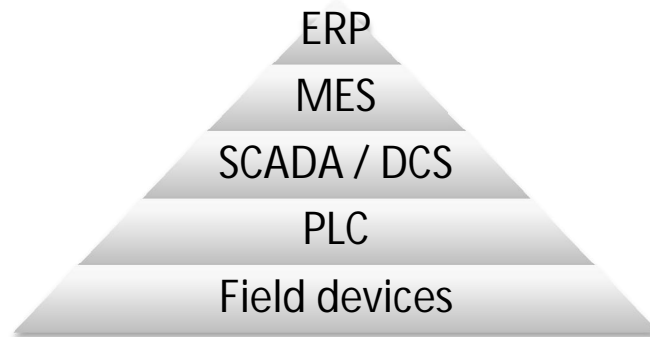
This white paper presents an overview of industrial automation systems and the role of malware and information security in this field. The topics discussed herein are of interest not only to industrial applications, but are also relevant for e.g. automation of municipal services.

The scope of the white paper is limited to considering the lower two layers of the automation pyramid (see below) and touching the third SCADA/DCS layer. The paper attempts to describe the field of information security in industrial automation from a generic point of view. Data and information in this paper, on which conclusions are based, are available in the public domain.

The chapters 1 and 2 provide an overview of information security in general and in industrial automation. Chapter 3 discusses the security of communication between devices and systems in industrial automation, while chapter 4 focuses on the security of individual automation devices. Wireless communications interfaces are briefly considered in chapter 5, followed by a summary of the paper in chapter 6.

### 1.1 Industrial automation systems overview

The different layers of an industrial automation system are commonly illustrated using the automation pyramid, which is presented in Figure 1.



**Figure 1. The automation pyramid and its five layers.**

The top of the pyramid consists of high-level systems such as Enterprise Resource Planning (ERP) solutions, which integrate IT systems across an entire organization. ERPs rely on the services of Manufacturing Execution Systems (MES) for managing individual plants or factories. According to the third layer of the pyramid, a plant is generally operated using Supervisory Control and Data Acquisition (SCADA) or Distributed Control System (DCS), which are industrial control systems used for monitoring and control of processes. Furthermore, SCADA may control and monitor devices such as Programmable Logic Controllers (PLC) or Remote Terminal Units (RTU). In the end, PLCs interact with sensors and actuators on the field device level, performing real-time control as necessary in the process.

### 1.2 Types of malware

The F-Secure Corporation terminology [1] defines malware as programs and files (e.g. viruses, worms and Trojan horses) which are created and spread in order to cause harm. The term malware is obtained by combining the words ‘malicious’ and ‘software’.

Three different types of malware are explained briefly: Worms, Viruses, and Trojan horses. Firstly, worms are self-replicating programs which uses security flaws to spread, often without the user of the machine being aware of this. Worms need not harm the machines,

however they quite often consume bandwidth and cause harm due to increased network traffic hindering important information exchange. Worms with payload (i.e. code designed to perform actions on the infected system) can cause damage to the infected machine and its system.

Viruses are also designed to self-replicate and spread to new machines. The F-Secure terminology [1] mentions a key characteristic of viruses being the replication mechanism. The terminology further notes that viruses commonly infect certain files, such as EXE or COM files on PC systems, or the Master Boot Record of hard drives and similar.

Trojan horses are, according to the F-Secure terminology [1] a program which appears to perform some action which may be desired by the user, but in reality performs some other (often undesired) action without the user knowing. Essentially, the function of the program is to make the user allow it inside the safe boundaries of the system, before silently beginning to execute malicious actions.

### **1.3 Current malware status concerning the industrial automation sector**

For more than 25 years malware has targeted the IT world. In the beginning, malware was easy to detect since it modified the visible content of the screen. Nowadays, malware tries to hide itself as much as possible which makes it difficult to detect. Furthermore, financially motivated cybercriminals are exploiting hundreds of thousands of PC's in order to make money. Until recently the industrial automation sector has not been touched by malware.

Stuxnet, probably created in 2009, has shown like no other former malware that security issues do not only reside in the regular IT-world, but

also in the industrial automation sector. Since general purpose operating systems like Windows are used in the scope of SCADA, vulnerabilities affecting the latter operating systems can also affect the industrial automation sector.

Stuxnet is a worm which is capable of spreading via USB-Sticks from Windows machine to Windows machine. Therefore, an infected machine does not necessarily need to be connected to the Internet. Stuxnet is using zero-day vulnerabilities (vulnerabilities which have not been known) and therefore, it is very difficult to protect against Stuxnet infections even with an up-to-date and patched Windows system.

Once Stuxnet has successfully installed itself on a Windows machine, it is capable of searching for automation systems. Moreover, it is looking for Siemens' Simatic factory systems, the so called SCADA systems. If Stuxnet cannot find any SCADA systems, it will remain silent and does not pursue any activity. On the other hand, if an automation system is found and more specifically high-frequency converter drives, Stuxnet tries to alter its functioning.

The reason why Stuxnet is so special is that it is very complex software and seems to be part of a targeted attack. Simply the size of the binary, 1.5 Mb, is unusually big for malware. Furthermore, it employs 5 exploits, 4 of them being zero-day vulnerabilities. A single zero-day vulnerability costs about \$50 000 to \$500 000, which makes Stuxnet a very expensive malware. Finally, in order to operate as silently as possible, Stuxnet has been signed with a stolen certificate.

All these three facts already make it clear that this malware has been created by a highly sophisticated attacker with a considerable amount of resources. Stuxnet was found in June 2010 and according to different sources it was created during 2009. This means those

professional attackers are able to target industrial automation systems and remain undetected for more than one year. Furthermore, Stuxnet has been installed on many computers worldwide. Would have been a targeted malware installed on 15 computers be detected anytime?

On October 18<sup>th</sup> of 2011 a new malware called Duqu which is very similar to Stuxnet was discovered. Compile times of this new malware could indicate that it has been created in the beginning of 2010. Duqu's intention is not to alter any functioning of industrial automation systems, but rather to collect sensitive information and send it to a remote server. Therefore, it can be considered more as a kind of Trojan Spy.

Although the maturity of malware and the rate of occurrence in the industrial automation sector are still quite low it is foreseeable that attacks may become more frequent and severe in the future. Potential scenarios might include vandalism or sabotaging of industrial plants, municipal services or critical infrastructure just for fun (by everyday hackers) or possibly the hijacking and/or blackmailing of entire plants.

In conclusion, there exist highly sophisticated and financially well-established malware creators targeting industrial automation systems. Nevertheless, there are signs that vandalism cases are occurring in the industrial automation sector, with similarities to vandalism caused by hackers in the ICT sector. Therefore it is crucial to analyze the risks and create appropriate defenses.

## **1.4 Standardization and related organizations**

Generally, standardization aims to provide commonly approved methods and practices to enable transparency in defined areas. With the help of standardization people can do e.g. internet banking safely and securely or use their

mobile phones. Also industrial standards facilitate global trade, protect human life (safety) and lately to drive more and more so called "green values". ICT has met the challenges of security for a relatively long period. This can also be seen when standardization activities in security sector are briefly introduced.

### **1.4.1 ICT security standards**

International telecommunication union (ITU) has group "ITU-T Study Group 17 – Security" which operates and covers a wide spectrum of application areas for security. It has published over seventy standards (ITU-T Recommendations) focusing on security. One key reference is X.509 which has enabled electronic authentication over public networks being an enabler for the rise of e-business. SG 17 is active in standard development and in coordination between applications specific groups (e.g. SmartGrid security) and other organizations. ITU standards are typically an underlying technology in industrial automation security or are linked to industry requirements via other organizations. [2]

Standardization work in IT security is also done by the International Organization for Standardization (ISO). Committee ISO/IEC JTC 1/SC 6 Telecommunications and Information Exchange Between Systems is developing telecommunication standardization for the exchange of information between open systems. This standardization includes both the lower layers that, as well as the upper layers that support the application protocols and services. [3]

Responsible technical committee for the security is JTC 1/SC 27 IT Security techniques. There are 98 published standards and all of them are ISO/IEC versions. The corresponding technical committee in IEC organization is ISO/IEC JTC 1/SC 27 IT security techniques.

The first standard has been published as early as 1998. The focus is on protection of information and ICT. This includes generic methods, techniques and guidelines to address both security and privacy aspects. [4]

#### **1.4.2 Industrial automation standards**

Industrial security standardization work under the International Electrotechnical Commission (IEC) is a relatively new area, when compared to IT activities under ISO. Usage of standard IT technologies and open systems in process control has increased the risk of security threats

in the industry. Connectivity to business/IT networks is also more and more common today. Also cyber attacks are more and more advanced today. This all means that there is a clear need for industry specific standards/ specifications and references.

Technical committee IEC/TC65 [5] and its four sub-committees prepare standards for industrial automation as well as process industry specific standards including security aspects. TC65 has published four generic security standards to this day and there are six standards under construction work. The focus of these standards is on the network and system level.

**Table 1. A list of published IEC standards on industrial automation security (status 11/2011).**

Standard	Notes
IEC/TS 62443-1-1 Ed1.0 (2009-07-30) Industrial communication networks - Network and system security - Part 1-1: Terminology, concepts and models	Technical specification defines according to its name definitions around the topic
IEC 62443-2-1 Ed1.0 (2010-11-10) Industrial communication networks - Network and system security - Part 2-1: Establishing an industrial automation and control system security program	Standard has a concept for a cyber-security management system (CSMS) for industrial automation and control systems (IACS) including risk analysis, addressing risks with CSMS and monitoring and improving CSMS.  ISO/IEC 17799 and ISO/IEC 27001 are corresponding standards for business/information technology systems. This standard has focus in specialties in IACS as failures can have impacts on health, safety and environment (HSE)
IEC/PAS 62443-3 Ed1.0 (2008-01-22) Security for industrial process measurement and control - Network and system security	This part is published as a publicly available specification/pre-standard for industrial control system (ICS) security policy.  ICS requirements for plant operation can differ from business/IT systems (e.g. response times) and these aspects are taken into account when setting specifications for industry.
IEC/TR 62443-3-1 Ed1.0 (2009-07-30) Industrial communication networks - Network and system security - Part 3-1: Security technologies for industrial automation and control systems	Technical report IEC/TR 62443-3-1 helps to evaluate technologies and countermeasures to build security for IACS. Topic is divided into categories: authentication, access control, data encryption & validation, management, IACS SW and physical security. Different measures are introduced, evaluated and recommended for each category. This standard also includes recommendations for device level.
IEC/TR 62541-2 Ed1.0 (2010-02) OPC Unified Architecture – Part 2: Security Model	OPC Unified Architecture (OPC UA) security model focus on securing the data exchange between applications. Security model describes the security threats of the physical, hardware and software environments for OPC UA use.

Preparation of industrial automation security IEC62443-standards is done based on the International Society of Automation (ISA) work. ISA99 committee “Industrial Automation and Control Systems Security” has originally started this activity, which is now utilized by IEC. [6]

A brief idea in this standardization is to divide process or plant in security zones connected by conduits and determine security by security assurance levels (SAL’s) (alike with safety integrity levels in functional safety). Detailed definition work based on this approach for the security system is ongoing.

### **1.4.3 Other industrial automation security related organizations and standards**

IEC technical committee TC 57 Power systems management and associated information exchange is responsible for international standards for power systems control equipment and related systems and associated information exchange. TC 57 launched the first standard about data and communication security in 2003 and there is a new series IEC 62351-1...8 for “Power systems management and associated information exchange - Data and communications security” available. The availability of electric power systems is vital for today’s infrastructure and as control of these systems is based on digital communications today and they are geographically wide systems, the security challenge is addressed with industry specific standard. Communications protocols for substation automation like IEC 61850 are included.

ISO/IEC joint technical committee JTC 1 SC37 is preparing standardization of generic biometric technologies pertaining to human beings to support interoperability and data interchange among applications and systems. Personal identification & ID cards with biometrics and

biometric data protections techniques, biometric security testing are excluded.

The Internet Engineering Task Force (IETF, <http://www.ietf.org>) is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. One of the operating areas is security, which has multiple working groups around different topics.

### **1.4.4 Standardization summary**

Standardization and related activities has strong position in ICT security. Nature of standardization fits well in security as it offers transparent and open platform for development. Technical solutions are reviewed by experts globally and the results are available in the public domain for use in industry. Security of automation is newer topic. However, there has been active work by society and wider IEC standardization is ongoing. Industry should adopt present automation security standardization and prepare for the forthcoming outcome. There are proven security methods available; it’s more an industry task to apply them in proper and relevant extent. Some limitations will apply in security realization due to historical reasons, but the current IEC standardization approach gives a good starting point towards better automation security.

## **2 Generic security considerations**

Generally known, the three most important elements of information security are confidentiality, integrity, and availability (CIA). In the scope of automation systems the CIA triad may shift to AIC, availability being the most important characteristic of an industrial automation system.

Information needs to be available if requested. Denials of Service attacks for instance are attacks against the availability of information. Especially in industrial automation systems it is crucial that the machines are working and that information can be retrieved from them. Availability can be achieved by creating robust systems with multiple layers of redundancy. Protection against Denial of Service attacks have to be put into place.

Integrity refers to protecting information against unauthorized modification. It is necessary to be able to detect if the given information is qualitatively valuable or not. If an attacker is able to alter information this represents an important threat which can be, depending on the case, even worse than deletion of information. Integrity therefore guarantees that if the information has been altered, then this can be detected. Different approaches can achieve integrity, depending on the need. A simple hash function can be used to calculate the hash over information. In other cases asymmetric cryptography might be employed in order to sign data with a private key. In both cases the legitimate receiver will have the certainty that the information has or has not been altered during transit.

Confidentiality means that information should be protected against unauthorized access. This can for instance be achieved by encrypting the information. An attacker who is able to receive the encrypted information is not able to disclose the content of the information (if properly encrypted and the secret keys are kept secret). Therefore the confidentiality is guaranteed.

A further concept which could be added to the CIA triad is called accountability. The intention of accountability is to be able to attribute a given action to a known actor. Furthermore, it might be necessary to know the time and activity performed by the actor.

## 2.1 Attacks and scenarios

Concerning industrial automation systems, essentially two types of attacks seem to be the most important: Information leaking attacks and tampering attacks. First, information leaking can have an important impact on advantages for competitors and may also affect the trust of the customers. Second, tampering can directly affect a customer and therefore represents an equally critical threat. If an attacker is able to successfully alter an industrial automation system, the consequences will inevitably damage the manufacturer. Sections 3.2 and 4.1 of this whitepaper will explain the detailed consequences.

## 2.2 Security program

Unless a security program already exists in an organization, it is of high importance that one is established. A security program commonly defines the objectives, policies, and guidelines regarding information security, and is also concerned with the practices used to analyze, implement, and maintain security in an organization and its systems. The security program should concern IT systems, industrial control systems, as well as the links between these two.

The security objectives, policies, and guidelines are important tools for employees and partner companies for understanding why information security is important and how security is achieved in the organization. By helping people to understand their role in creating information security, it is easier for them to act securely in their daily tasks.

A crucial aspect of the security program is the continuous assessment of threats and risks, prevention and countermeasures, and constant monitoring and improvement. The security program must be viewed as a continuous process

which maintains the information security in the organization at the required level.

In the field of industrial systems, the IEC 62443 standard presents terminology, models, and guidelines for establishing a security program.

### 3 Security in communication between devices

This section of the white paper describes security issues in the communication between devices, e.g. SCADA to PLC or field device, PLC to field device, or between field devices.

#### 3.1 Purpose of communication

Communication between devices in an industrial control system enables real-time monitoring and control of the target system and devices. Additionally, auxiliary functions such as parameterization and configuration, asset management, and potentially firmware upgrading may take place in the communication.

Higher level systems such as SCADA can be considered to have more of a “coordinating” role, acquiring data from the PLC and field device level and utilizing this information to supervise,

control and optimize the overall functionality of the system. Basic, non-real-time control executed by the SCADA might include changing or overriding setpoint values. Acquired data is often illustrated in a graphical user interface.

When discussing communication protocols and links, the OSI reference model is commonly used to represent the layers of abstraction provided by different protocols. The figure below illustrates the OSI model, in which a communication relationship between devices A and B is viewed as consisting of multiple layers. The application layer on top has the highest level of abstraction, providing functionality which is related to the main functionality of the device.

It is interesting to note that in the OSI model, which was introduced in the late 1970s and early 1980s, no layer explicitly considers the need for any security. Although this can be (and is) implemented inside layers in different protocols, the below illustration as commonly presented does not detail the need for security functionality. If, in some systems, intermittent layers do not address security specifically, then this commonly has to be implemented in the application layer.

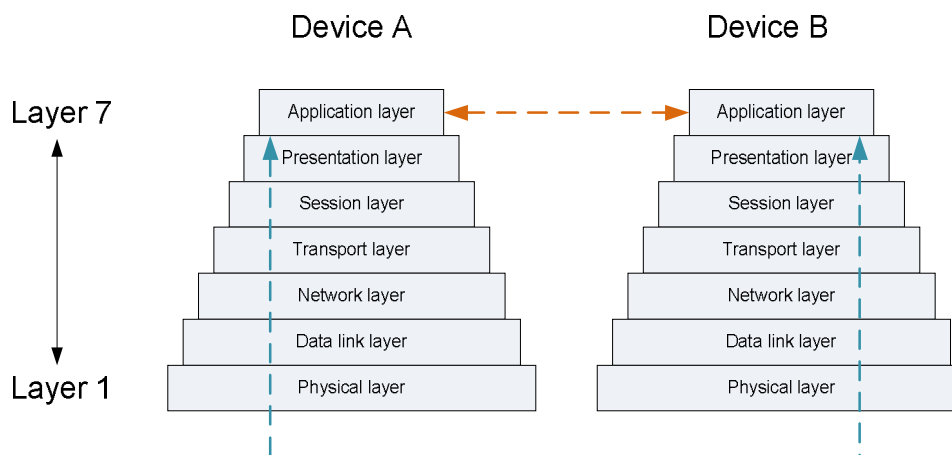


Figure 2 The OSI seven-layer reference model.

Depending on the communication link, not all layers specified in the model are used. However, the purpose of the model is to illustrate that the layering of protocols creates a kind of transparency; the layering of protocols means that communication follows the path of the blue dashed arrow, but to the application layer it seems like it is communicating directly to the application layer of another device, as illustrated by the orange dashed arrow.

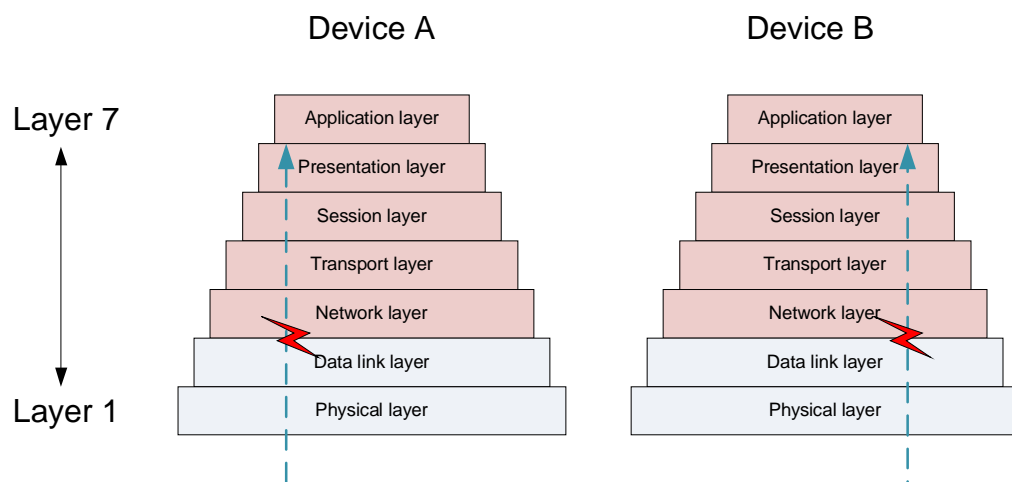
### 3.2 Security threats and issues

The use of communication networks has allowed installations to reduce the amount of cabling required, compared to e.g. wired I/O control and monitoring. The reduced cabling results in reduced costs and generally also a more manageable installation. Communication also better utilizes the capabilities of modern electronics. However, the available digital communication interfaces in both systems and devices pose security risks unless they are correctly addressed.

The word ‘attack’ commonly means the deliberate realization of a threat against a system,

with the purpose of evading or circumventing security measures and violating security policies. Attacks may be directed from outside an organization or plant, but they may also be initiated from within. It is also possible that attacks are initiated due to a suitable opportunity arising, perhaps without significant planning effort. There are also attacks which are highly deliberate and thoroughly planned, perhaps with such an important goal that large amounts of money, resources and time are used in implementing the attack. Furthermore, security threats and issues may be associated with either intentional or unintentional actions of people, e.g. deliberate incorrect operation of equipment vs. incorrect operation due to ignorance or lack of understanding of security policies.

When viewing communication between devices, the security threats can essentially be viewed to target either one or both of the devices, or the communication link and data. Before attacks are executed, it is common that the attacker performs some surveillance of the target system.



**Figure 3. An illustration of the effects of one layer in the OSI model being compromised.**

Considering the OSI reference model, the threats to a communication link may exist at any layer which is used. As an example, if the information of a message is attacked on the data link layer, and the attacker succeeds in e.g. modifying the source address of a message such that it passes through security defenses, then the layers on top of the data link layer are also compromised. The layers above the data link layer assume that the frame checking on the data link layer is reliable and has indicated a positive match, meaning that the encapsulated data shall be processed by the next higher layer.

It is not uncommon that an organization or its employees needs to have remote access to information and systems, commonly in the form of VPNs (Virtual Private Network). This functionality is enabled by tunneling protocols using cryptography to communicate information securely over an untrusted network. Often it is wrongly believed either by an organization or its employees that the VPN system is immune to intrusion or attacks. However, it is well-known that VPNs are commonly used for communicating sensitive information, and in many cases a VPN allows access to an organization's IT networks and applications. In other words, VPN connections form an attractive target for attackers, and therefore emphasis needs to be placed on analyzing the security of remote connections.

Threats to the security of VPN solutions are not merely technology-based, but are sometimes also due to human ignorance or error. As an example, even if the technical security issues of a VPN system are resolved, an employee may mistakenly or intentionally leak a username and password to a third party, effectively compromising the security of the entire system. If sent in plain-text using e.g. unencrypted email, it is possible for a hacker performing traffic sniffing to detect this information.

Over the last years, the use of remote connections for accessing industrial control systems has increased. The connections are made either to a SCADA or DCS system, however sometimes such connections are established directly to PLCs or even to field devices. It is essential to acknowledge that in connecting these devices to a network with internet access, the industrial control system and its devices are automatically exposed to threats.

### **3.2.1 Reconnaissance activity**

Learning to know the target which is to be attacked is commonly the first step in the plan. This prestudy may involve physically visiting the target if possible, obtaining information by observing and potentially stealing information.

For determining the structure or topology of the communication network(s) in the targeted system, various approaches can be considered. Information obtained through physical presence, as mentioned above, may include schematics/blueprints or documentation on the structure of an electrical system. Alternatively, configuration or project files used in control systems such as PLCs or SCADA systems may provide significant information regarding the layout and operation of a system. It is important to recognize risks such as this kind of document leaking during the commissioning phase of a plant, e.g. in the interaction with suppliers and subcontractors.

If access to a communication bus can be obtained, it may be possible for an intruder to listen to the communication activity in the concerned bus or network. Apart from the communication which occurs frequently (e.g. monitoring and control commands in the case of an industrial communication bus); infrequently or irregularly communicated information may be of interest to an intruder. Such information may include passwords or other sensitive information, but also e.g. proprietary protocols may be

observed with the purpose of reverse engineering. It may also be possible to determine some structure of the bus or network based on the activity log.

The way in which the targeted devices are physically installed or located in the plant affects the possibilities to study the communication system and/or gain access to it. If the devices are located inside cabinets or electrical rooms, most likely there is only a communication cable entering and exiting the enclosure. On the other hand, if the devices are distributed across the plant in the vicinity of the equipment or process controlled, this kind of enclosure need not be present which means that the device may be more exposed to a potential intruder.

If an attacker is successful in determining the type of devices in a system, additional information about individual devices may often be found online. Such information includes user manuals and data sheets, and in the terms of industrial communication often device description files and examples on the kinds of messages to use for interacting with the device.

### **3.2.2 Attacks on communication**

The communication between devices can be attacked in different ways.

An example of an attack on a communication bus is a man-in-the-middle attack, in which e.g. a gateway, switch or server is compromised. In this case, the information which is intended to flow through the intermediate component may be read, modified and/or forwarded to a third-party before it is sent to its original destination. This behavior may occur silently, avoiding detection while gathering information about the system. Theoretically, an intrusive device might provide incorrect commands or data to the legitimate devices, causing behavior which differs from that which is intended and expected of the system. Depending on the system and

circumstances, this may cause harm to the system, the equipment or the process which is controlled.

Another approach to compromising a communication bus is by overloading the bus, essentially equivalent to a denial-of-service (DoS) attack. This kind of attack will likely be detected once it is executed. Overloading of the bus or exhaustion of resources in the bus or some device connected to it, may prevent the system from performing the functions that are expected of it. The inability to execute functionality may mean that services are denied e.g. due to certificates or authorization not being communicated properly, which may prevent an operator of the system from exercising control. Also the inability to control setpoint and/or monitor actual values might cause the process control to malfunction, potentially leading to equipment damage, risk of personal injury, and/or financial loss.

Additionally the spoofing of information such as source or destination addresses forms another kind of attack. Protocols and systems which are not able to authenticate the source or destination address are vulnerable to spoofing, and would generally need precautions to be taken by e.g. the application layer to authenticate source and destination devices. This spoofing may be utilized to make intruding devices act like legitimate masters and attempt to control slave devices in a potentially harmful way.

It is important to protect the master devices in the communication buses or networks of the control system, regardless of which protocol or bus/network is in question. If a master device gets compromised, the attacker can issue commands appearing “legitimate” to the slaves.

It is important to note that if functional-safety-related data is communicated over fieldbuses using the various functional-safety extensions to

protocols, then e.g. man-in-the-middle attacks may compromise the safety of the process. Such a situation may pose a significant risk to human safety and also carries financial risk. This is especially true in systems where functional safety is entirely implemented by the automation system.

### 3.3 Traditional fieldbuses

In this white paper, the term “traditional fieldbus” is used to refer to fieldbuses using a non-Ethernet medium, for example (but not limited to) CAN- or EIA485-based fieldbuses. Communication protocols used in industrial and building automation systems include e.g. Modbus RTU, PROFIBUS DP, CANopen, DeviceNet, BACnet MS/TP and LON.

Many of these field buses are based on a master-slave interaction in which a master device commands and issues requests to the slave devices. Such commands and information returned by the slave devices is generally communicated cyclically at an update rate ranging from a matter of milliseconds to seconds. In many field buses, the master device is responsible for handling the start of the system, configuring the slave devices and ensuring that the system is operating correctly. A slave is not allowed to send messages to the bus unless requested by the master device, or if the slave has the token in a “token-passing” system.

One beneficial factor of traditional fieldbuses as compared to Ethernet networks is the restricted access to the bus. When a plant is commissioned, a specific set of devices are usually connected to the bus. For an attacker to gain access to the fieldbus, this would require either attaching an unfamiliar device to the bus, or obtaining access through an existing device.

Connecting a new device to the bus likely requires a physical presence at the bus; there may be a bus stub available for e.g. servicing or

diagnostics use, otherwise the intruder would need to connect such a stub. Obtaining access through an existing device likely involves hijacking or manipulation of the firmware in a bus node.

Another factor possibly limiting the attractiveness of targeting a traditional fieldbus may be the restricted openness and familiarity of the bus protocols. Although information about the protocols can be obtained with sufficient effort, specifications about IP-based protocols (e.g. UDP, TCP and FTP) are easily accessed. However, this “security by obscurity” cannot be viewed to increase system security considerably, because these legacy protocols are increasingly being replaced by standardized, well-documented protocols. Additionally, an attacker may cause problems even by disrupting the physical layer (electrical signals) and need not know the communication protocols used in the system.

Logging and analysis of the communication on a traditional fieldbus requires an access point to the bus, as well as a tool for capturing the frames being communicated. As mentioned earlier, this kind of eavesdropping or monitoring would require a direct access point with an intrusive device, or obtaining this information through an existing device e.g. by hijacking or using other methods.

**Table 2. An overview of a few fieldbus protocols and their properties.**

	<b>Modbus RTU</b>	<b>PROFIBUS DP</b>	<b>CANopen</b>	<b>DeviceNet</b>
<b>Speed</b>	9,6 – 19,2 kbit/s, or higher	up to 12 Mbit/s	up to 1 Mbit/s	125, 250 and 500 kbit/s
<b>Communication scheme</b>	Master-Slave	Master-Slave (multiple Master possible)	Master-Slave, Client-Server and Producer-Consumer	Master-Slave (multiple Master possible) or Peer-to-Peer
<b>Authentication of devices?</b>	No authentication	Device number	Optional, e.g. vendor ID	Optional, e.g. vendor ID
<b>Spoofing of data packets possible?</b>	Yes	Yes	Yes	Yes
<b>Remarks</b>	Master node has no specific address → Slave devices cannot know the identity of the master node.	Implementation is done preferably in Hardware.  Implements a class 2 Master which is used for e.g. diagnostics purposes.		

In conclusion, only some buses use device authentication, which is not especially spoof-resistant. Once a malicious attacker is able to get access to the different buses, he can conduct Denial of Service attacks and modify requests.

### 3.3.1 Modbus RTU

Modbus RTU is a mapping of the Modbus application layer protocol on the EIA-485 serial line. This protocol is a master-slave communication bus featuring a single master. In terms of communication, the master co-ordinates all transactions by issuing a request and then awaiting a response from the correct slave. The mandatory supported bitrates are 9600 and

19200 bits per second but others may also be supported. The response timeout, i.e. the time in which a slave must respond to a request by the master, may be several seconds and is application-dependent.

In the Modbus frames which are communicated, the master node has no specific device address; only slave devices are assigned an address. Additionally, the master can also use the address 0 for broadcast messages, which are processed by all slave nodes.

Because a Modbus request frame does not contain a master address, the slave devices cannot verify whether the device issuing the

request is the legitimate master, or an intruding device. If the frame is valid and the slave is in a suitable state, the request may very likely be processed.

The Modbus over serial line specification does not specify any mandatory requirement for the master node to indicate whether it has detected e.g. the presence of a second master on the communication bus. Essentially, when a master node in a Modbus-over-serial-line system is in the idle state, it does not receive any data from the bus. It might be the situation that the bus receiver electronics is disabled. In this case, it need not hear that another device is acting as a master node.

Since the bitrate of the system can be low, and the response timeout of the system is relatively long, an intruder does not need advanced hardware to access the bus. Commonly, the bitrates in a Modbus RTU bus are compatible with those of a PC serial port, meaning that a person with a portable computer and suitable EIA-485 hardware and software can easily connect to the bus and possibly enter the communication loop.

An intrusive device which has access to the bus can perform a denial-of-service attack e.g. by placing its transmitter in the “active” state, as opposed to the “inactive” state which allows the bus to be idle. Thus this rogue device could reserve the bus without releasing it, effectively preventing other devices from communicating normally. The intruding device could also activate its transmitter intermittently, resulting in what might seem like random errors in the communication on the bus.

### 3.3.2 PROFIBUS DP

PROFIBUS DP is a master-slave communication bus which supports the presence of multiple masters in the network. The masters share the EIA-485 medium using a special

telegram, in which one master passes the token to another master. Only the node which has the token can communicate. The PROFIBUS DP system can be configured to operate at up to 12 Mbit per second. The PROFIBUS DP protocol features a watchdog functionality, which is used by DP slaves to monitor that the master device sends updated I/O data frequently. The watchdog time is determined in the master setup and can be configured from a few milliseconds up to several minutes, or it can be completely disabled.

It is very common that the PROFIBUS DP protocol of a device is implemented in hardware, either by using a single-chip solution which includes the protocol and an MCU core in the same IC, or by using a separate MCU and a separate ASIC which handles the DP protocol. It is also possible to implement PROFIBUS DP in software but it is rather uncommon. Due to the high bitrates and thus tight timings of the protocol, a potential attacker likely needs to purchase suitable hardware for accessing the bus.

There are two classes of masters in the PROFIBUS DP system; class 1 which is cyclically commanding a number of slaves, and class 2 which is e.g. a laptop or programming console which can be used for configuration, maintenance or diagnostics. A class 2 master can communicate with other masters and their slaves; however it may only briefly perform actual I/O control. This is a managed feature which first requires stopping the data exchange with the primary master.

In each PROFIBUS DP message, two fields are reserved for the addresses of the sending and receiving devices. A PROFIBUS DP slave device remembers the address number of the master which initialized it; thus an attacker needs to detect the address of the master controlling a slave, in order to attempt to command the slave using a spoofed address.

When the bus system starts, masters which perform cyclical data exchange initialize the slaves which are assigned to them. This initialization consists of a parameterization and a configuration step, in which the PROFIBUS DP parameters as well as possible vendor- or device-specific parameters are set in the slave. Additionally, the device number is verified against that which the master expects, to ensure that the correct type of device exists at the correct address. In the configuration step, the length and structure of the periodically transmitted I/O data is set, as defined in the master setup. Both the parameterization and configuration procedures can be accepted or rejected by the slave. Each slave keeps track of which master address configured them for data exchange.

To avoid the conflicting situation in which different masters try to command the same slave, it is possible to lock slaves to a single class 1 master (the master which performs the initialization procedure described above). It is, however, also possible that a slave is not locked, which would mean that a slave could be claimed by different masters. This is nevertheless more an issue of network management, i.e. ensuring that slaves are locked to their primary master.

The PROFIBUS DP-V1 extension specifies an acyclic communication which can be used as needed to e.g. read or write variables or parameters of a device, if it implements some device profile such as PROFIdrive or encoder profile. A master class 2 can perform read and/or write operations targeted at a PROFIBUS DP slave device independent of the slave's relationship to its primary master. This presents risk in case the master class 2 can modify parameters which affect the operation of the device.

If a device gains access to the bus, and is able to perform DP-V1 functionality, then it can attempt

to read the Identification & Maintenance (I&M) information from bus devices. The I&M is a structure of device identification information which at a minimum includes the I&M0 information, but optionally also other I&M fields (see the table below) This kind of information about a device (which can be identified using its slave number in the bus) can reveal what the device is doing, the device type and give clues as to how its behavior could be compromised. The I&M information may be changeable by the owner or operator of a device, so in case this information is not write-protected, an intruder may attempt to change the information, e.g. change the text describing function, task, location or installation date so that identification of the device is tampered. Table 3. The I&M information in PROFIBUS and PROFINET devices.

<b>I&amp;M</b>	<b>Field</b>	<b>Information</b>	<b>Description</b>
<b>I&amp;M0</b>	Manufacturer ID	Number code	Reveals the device vendor.
	Order ID	Text string	Reveals the order number of the device.
	Serial Number	Text string	Reveals the serial number of the device.
	Hardware Revision	Version Number	Reveals the hardware revision of the device.
	Software Revision	Version Number	Reveals the software revision of the device.
	Revision Counter	Number	Change counter.
	Profile ID	Number	Reveals implemented device profile.
	Profile-specific type	Number	Possible profile-specific code.
	I&M version	Version Number	Reveals implemented I&M version.
	I&Ms supported	Bitmask	Reveals which I&Ms are supported.
<b>I&amp;M1</b>	Function Tag	Text string	Describes the function of the device.
	Location Tag	Text string	Describes the location of the device.
<b>I&amp;M2</b>	Installation Date	Text string	Reveals the installation date of the device.
<b>I&amp;M3</b>	Descriptor	Text string	Freely assignable comment/annotation.
<b>I&amp;M4</b>	Signature	Text string	Can be used as signature for tools.

As with e.g. Modbus RTU, an intruding device could issue a denial-of-service attack on the PROFIBUS DP bus by transmitting data either in bursts or continuously. This would compromise the normal communication in the system resulting in partial or complete loss of functionality.

### 3.3.3 CANopen

CANopen is a higher-layer protocol which is based on the CAN data link layer protocol. CANopen supports bitrates up to 1 Mbit per second. The network is managed by a single NMT master that controls the devices on the network. Different types of communication are defined as “protocols” in CANopen, e.g. Process Data Object (PDO), Service Data Object (SDO) and Network Management (NMT) which provide different sets of functionality to the system.

The PDO protocol can be configured to transmit in either synchronous or asynchronous mode. In the synchronous mode, devices communicate their input/output data within a specified time window of receiving a special SYNC command from a synchronizing application. Asynchronous data is transmitted without any relation to a SYNC command. The triggering of messages can be event-driven, timer-driven or remotely requested. The asynchronous PDO protocol is vulnerable to an intruding device sending data to the bus e.g. with spoofed addresses.

The CANopen protocol supports a node guarding protocol, using which the NMT master monitors that slaves respond to a guarding request within a specified time window. If such timely responses are not provided by a slave, or its NMT communication status changes then the NMT master should react on this event. Life guarding is essentially allows the slave to monitor the guarding performed by the master, and allowing the slave to react to not being guarded in a timely manner. Additionally,

CANopen features a heartbeat mechanism in which a heartbeat producer cyclically transmits a message to heartbeat consumers. Missing heartbeats indicate that a slave has gone offline and allows for reaction. It is mandatory to implement either guarding or heartbeat.

The CAN messages is shared by means of an arbitration of the CAN message priority. The priority is determined based on a bit field which is mandatory in all CAN frames. The hardware of every CAN device is required and able to detect when another device is transmitting a message with a higher priority. In this case, the “losing” device backs off and cancels its transmission, instead listening to the “winning” device.

A device which gains access to the bus can silently monitor the bus and observe the communication. Additionally, a denial-of-service attack could be launched utilizing the automatic, hardware-based arbitration of the CAN frame. If an intrusive device repeatedly transmits messages with the highest of priorities, no “normal” CAN frames will be communicated on the bus as these will losing the priority arbitration. Effectively, the bus is overloaded and system functionality which relies upon communication will suffer. Guarding or heartbeat protocols will trigger events after a specified time, but communication is not possible as long as the attack proceeds.

In CANopen, the highest priority CAN frame is an NMT message for network management. The NMT message contains one octet of data which indicates the requested state of the targeted node, and another octet which specifies the node which should change its state when receiving the NMT message. If the node number is 0, all nodes receiving the message shall change their state. Thus, an intrusive device could control the state of the CANopen devices on the bus.

It is also possible that an intrusive device which has access to the bus can transmit either valid or invalid data to the bus, corrupting some of the normal communication in the system. In the case of transmitting valid data, the intruding device can act as a master, commanding slave devices as it desires.

CANopen specifies mandatory objects which identify the devices on the bus. An example is the 1000h “Device Type” object, which contains the device profile number which is implemented by the device. Another mandatory object is the 1018h “Identity” object containing information such as the vendor ID (mandatory) and possible the product code and revision number. This readily available information may allow an intruder to study the device configuration in the network, learning which kinds of devices (based on device profiles) and whose devices (vendor information) are present.

CAN-in-Automation (CiA), which is the user organization for CANopen, has initiated a working group on the topic of communication security in CANopen and the reliable encryption and decryption of CAN frames.

### 3.3.4 DeviceNet

DeviceNet is a connection-based network protocol which is based on the CAN data link layer. The DeviceNet protocol supports three bitrates; 125, 250 and 500 kbit/s. Master-slave and peer-to-peer communication is supported by DeviceNet, still the majority of installations follow the master-slave scheme. There may be multiple masters on the same network.

Like many other protocols, DeviceNet specifies a mandatory “Identity” object which contains information such as vendor ID, device type, product code, product name and revision information. This information can provide an intruder with clues regarding the devices which are installed in the system.

The master in a DeviceNet bus may scan the network at startup with the purpose of verifying that the actual network corresponds to that which is configured. It can use information such as the Identity object for checking vendor ID numbers and product codes. DeviceNet features an optional heartbeat functionality which is used to monitor the status of devices on the bus. The heartbeat interval can be configured to an integer number of seconds, defining the intervals at which the slave device sends a heartbeat message to the master. If the heartbeat from a slave stops, the master interprets this as a slave going offline and can react on this event.

The DeviceNet data link layer is the same as used in e.g. CANopen or other CAN-based protocols, using priority arbitration in hardware. Because of this similarity, these protocols are all vulnerable to the same kinds of attacks which are targeted at the CAN data link layer or the physical layer. An intruding device may monitor the bus, learning how the primary master opens connections to its slaves. The intruder may impersonate the primary master and send incorrect commands to the slave devices.

## 3.4 Ethernet networks

Ethernet is a de facto standard medium in communication, with a multitude of communication protocols and applications. The Ethernet specification itself covers the two lowest layers of the OSI reference model (Physical Layer and Data Link Layer), while the Internet Protocol (IP) suite and its core protocols provide Transport Layer (e.g. TCP, UDP) and Application Layer (e.g. HTTP, FTP or TLS/SSL) functionality.

### 3.4.1 Ethernet physical layer

Except for the lowest layers in the OSI reference model, an increased level of abstraction as provided by higher layers also invites to a greater risk for intrusion, vulnerabilities and

malicious activity. The Physical Layer is concerned with issues such as electrical characteristics and the encoding of data in the medium, and in general does not implement any security features.

### 3.4.2 Ethernet data link layer

On the next layer is presented the concept of the Ethernet frame (of which there exist a few different types) containing amongst other things the “MAC addresses” of the source and destination device. These six-octet MAC addresses were intended to be a permanent and globally unique numerical identifier for each network device. However, in most modern hardware it is possible to change the MAC address, which may be maliciously used in e.g. MAC spoofing. An example of a MAC address is 00:21:99:00:2D:A9.

In an Ethernet MAC address, the first three octets form an OUI (Organizationally Unique Identifier) which is purchased by a device vendor from the IEEE registration authority. These first three octets can be used to determine the vendor of the device which is the sender or receiver of an Ethernet frame. In other words, an attack which is intended to target the equipment of a specific vendor theoretically only needs the MAC address OUI to detect potential targets.

The MAC address identifies a single device, meaning that a receiver cannot determine from the MAC address of an incoming frame whether the sender is installed on the same network segment (link) or on another segment which is bridged to the receiving device’s segment. In other words, MAC address filtering cannot be used to create security barriers based on network topology.

MAC address filtering can be used to prevent access to a network or prevent processing of a frame, but it can be circumvented by an intruder who knows how to spoof his or her MAC

address. If the intruder is able to find a MAC address which is not filtered, it is possible that the intruder gains access to a network (depending on whether other protective measures are in place) or that the frame is processed by the receiving device.

### 3.4.3 Internet Protocol

Inside an Ethernet frame, there may be enveloped an Internet Protocol (IP) datagram. Currently the IPv4 standard is most widely used but IPv6 is in deployment. The IP protocol is a connectionless protocol, meaning that messages can be sent from one device on the network to another without requiring any prior arrangements e.g. handshaking. The sender might believe that the receiver is on the network and capable of receiving data, when in fact it is not. The IP protocol does not guarantee delivery success or order of delivery, nor can it guarantee that a transmitted message will be received only once by the recipient, as network conditions may cause loss, duplication or out-of-order delivery of IP packets. This kind of reliability and security requirements are enabled by the use of higher-layer protocols e.g. TCP.

The IP protocol makes use of IP addresses (in IPv4 these are four-octet addresses) and subnet masks which allow the subdivision of a network into subnetworks. An example of an IP address is 192.168.1.0, with a subnet mask of 255.255.255.0. This example means that the subnetwork has the network prefix “192.168.1” and the last eight bits of the IP address is used for identifying individual devices in that subnetwork.

The Address Resolution Protocol (ARP) is designed with the purpose of providing a way for devices and networking equipment to resolve the MAC address corresponding to an IP address (i.e. learning the device identification number of a specific network device). Sometimes there is also the need for the opposite conversion, i.e.

learning the IP address for a specific MAC address. One protocol intended for that use is the Dynamic Host Configuration Protocol (DHCP).

IP filtering is the process of allowing certain IP datagrams access to a network, or allowing frames to be processed in a device. Different fields of the datagram may be subject to filtering, e.g. protocol type, datagram type, the source or destination IP address. If the filter rejects a frame based on its IP datagram, the frame is discarded as if it had never been received.

If an intruder is able to configure the contents of an IP datagram suitably, so that it passes the security settings of a network device, it is possible that he or she gains access to a network or that a frame is processed by a receiving device.

### 3.4.4 Transport layer

The Transport layer builds upon the services of e.g. the Internet Protocol. This layer enables the detection of missing or out-of-order frames, or retransmission of frames which have not been acknowledged by the receiver. Additionally, the transport layer may provide a concept of connections between network devices, so that a handshaking is performed before data exchange can commence between two devices. In the Internet Protocol suite, typical transport layer protocols include the User Datagram Protocol (UDP) and the Transmission Control Protocol (TCP).

In addition to the aforementioned improvements, the transport layer protocols TCP and UDP introduce the use of ports in communication endpoints. Ports are associated with a network device IP address so that the two in combination form the complete source or destination address for a communication connection. Some port numbers are predefined for commonly used services, while some port numbers may be used for custom purposes. As an example, the

Modbus protocol over TCP uses the port number 502.

Firewalls are commonly configured to check the port numbers in Ethernet frames and allow or disallow certain traffic into or out of a network. This functionality is referred to as port forwarding. An example of the use of port forwarding could be allowing computers on the Internet to perform an HTTP access to a web server within a private LAN, by allowing connections on the port 80 (which is reserved for HTTP traffic).

When preparing for an attack, an intruder may try to connect to a range of ports in sequence on a specific network device. This activity is commonly referred to as port scanning, the purpose of which is to detect any open ports which may be used as an entry point into the device. Another type of scanning is called portsweep, in which connection attempts to a specific port number is made to multiple network devices.

### 3.4.5 Network configuration

The networking equipment must be correctly configured and appropriate security features must be enabled. Because different equipment supports different options for configuration, it is difficult to provide a comprehensive list of things to address. At a minimum, however, the default username and password for administration of the settings must always be changed, so that it is not trivial to change the settings of equipment. Passwords for configuring network equipment must be selected with good strength, i.e. having different kinds of characters (numbers, upper- and lower-case characters, special characters) and with sufficient length.

IP address filtering in networking equipment may prevent attackers with basic skills from accessing the network. Similarly, MAC address

filtering is another barrier for preventing simple attacks. URL filtering may be applied in order to prevent persons inside the safe network zones from accessing known, insecure content in the Internet. Application-level firewalls or ones supporting stateful packet inspection (SPI) can be used to further increase the level of security.

Firewalls shall be configured in such a way that only the required functionality is open and enabled. Firewalls shall be used where needed, e.g. as required by the network security zones which are setup for an organization. Firewalls from different vendors may be used in order to provide some security due to diversification.

Appropriate encryptions need to be used e.g. for wireless connections. As an example, many Wireless LAN (WLAN) routers support WEP, WPA and WPA2 encryptions. Of these alternatives, WEP should not be used, WPA can be used but WPA2 provides the best level of security. Furthermore, in order to increase the level of security, WPA should be used in the enterprise mode (known as WPA-Enterprise or WPA-802.1X mode).

Manufacturers' recommendations regarding which equipment works well together should be followed. This is especially true if the recommendation is based on security functionality.

### 3.4.6 Network topology

It is important to consider how the Ethernet network is constructed, in terms of topology. The bridging between networks of different security, e.g. between an Industrial Ethernet network and an office- or IT-network, should be carefully considered and configured.

There is also a risk regarding physical security if there are unused ports in Ethernet equipment which can be used by an intruder to gain access to the network. Sometimes, unused ports in

networking equipment are used for port forwarding, which means that the traffic through e.g. a switch or similar piece of equipment is forwarded to a certain, unused port. This port can be used for logging and traffic analysis purposes by e.g. connecting a computer with suitable capture software.

Because Ethernet has become more popular in different automation systems, there have also appeared a number of gateways and bridges which allows connection of Ethernet to traditional fieldbuses. These devices present an access point from an Ethernet network to field buses, which were originally designed to be closed networks. The features of such gateways and bridges, such as integrated web interfaces for configuration or monitoring with the purpose of allowing simple configuration possibly from remote locations, may encourage the loosening of security configurations. As an example, the browsing of a web interface generally requires the port 80, which is reserved for HTTP communications, to be open. However, access to the web server in a device from outside means that the HTTP port is also exposed to non-intended users, which may target attacks on it.

Although this kind of web interface is commonly protected by a username-password combination, the default value is often listed in the device manual which is available online. Although the password may be changed, it is not uncommon that the new password has poor strength, due to user/operator ignorance and/or inadequate instructions to choose the password cleverly. It is worth mentioning, that if an intruder manages to determine the password for a gateway (or some other networking device) it may be possible for the intruder to change the password, security settings or other functionality related to the system.

### 3.4.6.1 Ethernet hub

An Ethernet hub is a layer 1 (physical layer) device which does not examine or manage the traffic through it, mainly rebroadcasting entering packets to all the other ports. Because hubs are generally not concerned or even aware of frames or packets, they tend to operate mostly on raw data. A consequence of this is that hubs do not integrate any security functionality and thus should not be used in industrial automation networks.

### 3.4.6.2 Ethernet switch

By an Ethernet switch it is generally meant a layer 2 (data link layer) device which routes data based on MAC addresses; however there are also switches in higher layers. There are generally two kinds of switches; unmanaged ones which have no configuration interface or options, and managed ones which can be configured by the user.

A switch discovers the devices which are connected to its ports and maintains a list of such devices and their addresses. When an Ethernet frame enters to the switch, the switch checks its table to determine which port (if any) contains the device with a MAC address matching that of the frame. If a match is found, the frame is forwarded only to the concerned port.

### 3.4.6.3 Ethernet router

An Ethernet router is a layer 3 (network layer) device which routes data based on IP addresses, thus enabling the construction of subnets. It checks the incoming frames for their address information and determines how to forward the frame, based on a routing table or routing policy. The routing table can be dynamically achieved using e.g. routing protocols; however an operator may also configured static routing rules manually.

### 3.4.7 Industrial Ethernet protocols

There exist a large number of communication protocols which are based on the Ethernet medium; some examples of industrial protocols include PROFINET IO, EtherCAT, Modbus TCP, and EtherNet/IP. Other areas of automation include BACnet/IP in building automation or GOOSE messaging as defined by IEC 61850 in power systems communication.

**Table 4. An overview of a few industrial Ethernet protocols and their properties.**

	<b>Modbus TCP</b> <b>Modbus UDP</b>	<b>PROFINET IO</b>	<b>EtherCAT</b>	<b>EtherNet/IP</b>
<b>Communication scheme</b>	Master-Slave	Master-Slave (multiple Master possible)	Master-Slave	Master-Slave (multiple Master possible) or Peer-to-Peer
<b>Authentication of devices?</b>	No authentication	Initialization, vendor ID and device ID	Optional, e.g. vendor ID, product code, revision number, serial number	Optional, e.g. vendor ID
<b>Spoofing of data packets possible?</b>	Yes	Yes. If an existing device cannot be compromised, specialized hardware is needed	Yes. If an existing device cannot be compromised, specialized hardware is needed	Yes
<b>Remarks</b>		Sometimes implemented in hardware, especially in the class Isochronous Real-Time	Hardware implementation in slaves	

### 3.4.7.1 Modbus TCP and UDP

Modbus TCP is a mapping of the Modbus application layer protocol onto the TCP/IP transport layer protocol. Another variant is Modbus UDP which maps the same application layer protocol onto the UDP protocol. These mappings permit the use of Modbus on the Ethernet medium.

In Modbus TCP/UDP, the IP addresses are used to identify devices. A Modbus TCP/UDP device may include functionality for both slave and master modes. The message frames communicated in this network do not separately identify a device as a master, which is analogous to the master not having its own address in the

Modbus RTU protocol. Devices cannot authenticate a device as a “legitimate” master.

The difference between using TCP and UDP lies mainly in the consideration that Modbus on TCP ensures that messages are reliably delivered, in order, while potentially reducing the timeliness of delivery. Using UDP, there is no guarantee that messages are delivered in the correct order (or delivered at all), but the logic of missing requests/responses and retry is moved to the Modbus application layer. Thus, timing considerations are different from the TCP case in which retransmissions are handled by the transport layer. Data such as setpoint and/or actual values are relevant only for a short span of time after “sampling”, thus there is little sense in trying to retransmit these packets numerous

times just because TCP says so. When using UDP, retransmission of old data can be avoided in case one cycle of data is missed; the next cycle with up-to-date data is sent instead.

The bridging between Modbus TCP or Modbus UDP into the serial line variants, e.g. Modbus RTU, has become more common as Industrial Ethernet also increases in popularity.

### 3.4.7.2 PROFINET IO

PROFINET IO is an Ethernet-based protocol designed for real-time communication. The experience gained from the PROFIBUS fieldbus was integrated with the Industrial Ethernet technology to create PROFINET. The master device in a PROFINET IO system is called the “Controller”, while slaves are referred to as “Devices”. Cyclic data exchange in PROFINET IO takes place directly in the Ethernet layer 2, not involving any transport protocols such as UDP or TCP; the messages are addressed using the MAC addresses of the PROFINET IO devices. Acyclic data is exchanged using the UDP protocol.

The cyclic data exchange connections are monitored using a watchdog time, which is configured as a multiple of the update (cycle) time of the network. As an example, if the update time is 8 milliseconds and the watchdog time multiplier is 3, then the watchdog time will be 24 milliseconds. If the communication is idle for longer than this period of time, the device monitoring the watchdog will detect this event and execute some reaction; however this is device- or user-specific.

Because the PROFINET IO real-time data exchange frames are communicated in the Ethernet layer 2, these frames contain only MAC addresses. This means that the real-time frames cannot be communicated outside a subnet which is delimited by a router, because routers form

subnets based on IP addresses as defined in layer 3.

A hardware implementation is required for the most deterministic class of PROFINET IO devices, known as IRT (Isochronous Real Time). This is commonly an ASIC with integrated switch and other functionality needed for the PROFINET IO IRT protocol.

A PROFINET IO Controller always needs to connect to a PROFINET IO Device using an explicit “Connect” message. After this the Controller downloads startup parameters to the device, following a handshake verifying that the startup is successful and complete. An intruder wishing to establish a connection to a device, i.e. to act as a second master, has to know how the device is structured and how to initialize it properly at startup. Based on vendor ID and device ID it may be possible to find the GSDML description file for the device, however if the device is modular then knowing the true configuration likely requires physical access or documentation about the system.

The PROFINET IO protocol furthermore requires the same I&M functionality as described for PROFIBUS DP earlier. PROFINET IO devices thus expose the same information to anyone who can access it.

At startup the PROFINET IO controller provides the vendor ID and product ID that is configured for the targeted IO device. The device checks the master’s expected information against its own data and aborts the connection request if a mismatch occurs.

Two potential methods of attacking the communication in a PROFINET IO system are described in [7]. The authors propose that it may be possible to modify the outputs of a PROFINET IO Device without being detected by either the Device or the Controller.

### 3.4.7.3 EtherCAT

EtherCAT is a real-time Ethernet-based protocol in which the Ethernet frame moves similar to a train along rails from the master, through all slave devices and back to the master device without stopping. Slave devices process the Ethernet frame “on-the-fly”, causing only a tiny delay in each slave device. Devices take data from and put data to different sections of the EtherCAT frame, depending on how the EtherCAT master has configured the slaves at startup. The “on-the-fly” processing requires a specialized hardware in the slave devices, while the master implementation can utilize virtually any Ethernet network interface such as PC networking cards. As a result, compromised slaves could spoof at least data in the parts of the EtherCAT frame which they are configured to process. A compromised master, or one which is the subject of a man-in-the-middle attack, could result in spoofed data and altered configuration of the slaves.

Depending on the implementation, the EtherCAT master can be configured to check various aspects of the slave device, e.g. vendor ID, product code, revision numbers and serial number at startup. These features are commonly optional and can be disabled.

EtherCAT supports different subprotocols which are tunneled in EtherCAT frames; examples of such are CAN-over-EtherCAT (CoE) and Ethernet-over-EtherCAT (EoE). The CoE protocol is an integral part of EtherCAT and is used to identify, configure and control the slaves. The EoE protocol is optional and allows non-EtherCAT devices to be added to the system using switchports, which “de-tunnel” the Ethernet frames from the EtherCAT frames. It is possible to tunnel Ethernet frames through the master PLC if such a feature is implemented and enabled.

Logging of the EtherCAT traffic may be possible in case there is a switch in the network which is configured to forward messages to an unused port. If the switch is not a real-time switch, the real-time attributes of the EtherCAT communication may not be evident from the log. It may also be possible to log the EtherCAT communication via the EtherCAT master, which may e.g. be PC-based running Windows.

The EtherCAT protocol diagnoses the network and provides indications regarding e.g. network or slave problems. Additionally, the protocol supports redundancy such that if the network is constructed like a ring, a broken or disconnected cable or node somewhere in the ring does not prevent the operation of other nodes. Furthermore, EtherCAT features time synchronization between master and slaves.

It is possible that an intruding device which gains access to the network transmits messages into the network, which easily disrupts the EtherCAT communication. This will interfere with both control and monitoring of the process and equipment, as data exchange is hindered and furthermore time synchronization-dependent functionality is disrupted.

### 3.4.7.4 EtherNet/IP

EtherNet/IP is an Ethernet-based protocol which implements the Common Industrial Protocol (CIP) which is also used in DeviceNet. EtherNet/IP uses the UDP and TCP protocols for communication. EtherNet/IP follows the master-slave and peer-to-peer communication models as common in other protocols.

The EtherNet/IP protocol presents the same “Identity” object as DeviceNet, thus this information can be used for detection and study of EtherNet/IP devices. An EtherNet/IP master may check the information of a device, e.g. vendor ID and product ID, in order to authenticate the device. Another similarity to

DeviceNet is the heartbeat mechanism by which the network device states are monitored.

EtherNet/IP devices are subject to the generic threats which face essentially all Ethernet-based devices, e.g. espionage by logging of the network communication, denial-of-service attacks, potential man-in-the-middle attacks, or attacks on a specific layer in the OSI model.

### **3.5 Recommendations for enhancing security**

A large number of recommendations for enhancing the security in communications between devices can be identified, but it is not practical nor is it the purpose of this document to exhaustively list these. It must be remembered that no system is 100 % secure so that it could not be broken, given enough resources, time and motivation; this also holds true for communication networks in industrial control systems. The set of feasible improvements to this kind of security depends largely on the individual characteristics of a system, and the modeled threats, tolerated risk and derived priority for addressing vulnerabilities shall guide the implementation of security improvements.

Physical security can be considered of key importance concerning access to the communication networks, especially in the case of traditional fieldbuses, in which physical access to the bus is necessary for malicious behavior. This is an important aspect also for Ethernet-based networks, although attention must be paid in these networks also to remote access. In terms of the security at the plant, one item to consider includes the cabling of the bus; the possibilities for an intruder to trace the cabling and document the system could be reduced. Also, locking devices inside cabinets or electrical rooms discloses fewer clues about their use, connections and configurations, compared to devices which are not enclosed. The access

rights to physical locations in which critical systems are operating should be assessed and, if deemed necessary based on vulnerability and tolerated risk, restricted appropriately. It is also possible to implement critical communication links using optical fiber, which are not easily intercepted.

Unused Ethernet ports in e.g. network equipment (switches, hubs) or in field devices (with integrated switch functionality) can be protected from unauthorized use e.g. with plug guards, which are physically plugged into the port. The plug guard requires a key to be removed, thus preventing anyone not possessing the key from connecting to the network.

Functionality which is password-protected could attempt to ensure that the factory-default password is changed to some other password. More importantly, such password-protection should require that the selected password fulfills certain requirements regarding strength, such as requiring a mix of letters, numbers and special characters, and having a minimum length of e.g. six or eight characters.

Care should be taken to improve security also at the lowest layers in the OSI model. As an example, if the data link layer is compromised for instance by a falsified source address or tampered data, the layers above in the OSI model are also affected although they might not be aware of the situation.

When selecting communication protocols for the process control network, but also e.g. for remote connections, attention should be paid to the security features which are provided by the protocols. As examples can be mentioned the strength of encryption or the protection against replay attacks using sequence numbering and the ease of circumventing these mechanisms. Also, when procuring equipment for networks, the security functionality offered by different

devices should be assessed, preferring devices with advanced security options. A look at the user's manual, which is commonly available for download from the Internet without cost, can provide an impression of how mature the security of a device is.

The threats against a communication network in a plant shall be assessed and modeled, either by an internal or external assessment team. In this analysis, it is crucial that the participating personnel think as an attacker. A list of vulnerabilities and possible exploits shall be established, against which a list of security measures shall be constructed. Highest priority should be assigned to implementing the measures which remedy the vulnerabilities with highest risk, followed by the remedies which are easiest and quickest to implement. This is not to say, that low-risk vulnerabilities and hard-to-implement security measures can be ignored or avoided, but they may be assigned to a later phase in order to quickly raise security.

## **4 Security in field devices**

This section of the white paper discusses security aspects of field devices. By this is meant the sensors and actuators at the lowest layer of the automation pyramid, which interact with, control and monitor the actual process and equipment in a system. These devices are commonly based on embedded firmware with functionality to allow customization to suit the application needs.

### **4.1 Security threats and issues**

There are mainly two threats against field devices which should be taken care of: Information leakage and tampering of the devices.

#### **4.1.1 Information leakage**

Depending on where the field device is employed, information leakage can occur through mainly three channels: Through an IT leak, human leak, or physical leak. At the manufacturer side, an IT leak could happen where the field devices are programmed and configured. If responsible computers are infected with a Trojan horse, an attacker could easily obtain configuration files and access keys. Social engineering attacks should not be underestimated and can give an attacker the possibility to access confidential information about field devices through a human leak.

The physical leak could happen on the manufacturer side, but is more probable when the device is employed in the field. If a malicious attacker is able to obtain a field device he is able to analyze it extensively. Reverse engineering can be conducted on the analyzed hard- and software.

#### **4.1.2 Tampering risks**

Field devices can be manipulated in general in two different ways: The attacker can insert spoofed firmware updates or change device parameters through the PC software controlling the device, or using the device's integrated user interface such as a keypad and display.

If an attacker is able to control the PC, configuring the devices at the manufacturer, he can alter configuration parameters and the installed firmware. An attacker will alter device firmware with either one of the following two motivations: First in order to provoke dysfunction of the device, or second to improve the performance of the device.

A malfunctioning device can create damage wherever the field device is employed. Furthermore, not only the field device will be affected, the entire production chain in which the field device works will be influenced.

Tampering with a field device in order to achieve better performance can be driven by financial benefits. If an attacker can successfully and significantly improve the performance of a field device, he can buy cheaper devices, tune them, and sell them consequently as more expensive devices. Usually firmware updates are encrypted with a symmetric key which is stored on the field device. Furthermore, the encrypted firmware updates can usually be retrieved from the Internet. Therefore, in order to modify a firmware update, the attacker would first need to obtain the encryption key, decrypt the firmware, modify it, and finally encrypt it again. Since the encryption key is stored on the field devices, a malicious attacker needs to search in memory for the key.

## 4.2 Simple field devices

By “simple” field device is herein referred to as a device which does not employ a full-scale operating system, but possibly some kind of scheduler mechanism. These scheduler mechanisms may be developed completely vendor-specifically, thus there may be no information publically available on their workings. Commonly, the design decision to use a scheduler instead of an operating system may be related to the low complexity of the device, limited resources such as memory, or other constraints.

As scheduler-based devices consist of embedded firmware, the reverse engineering and the modification or tampering is non-trivial. However, if an attacker is able to obtain the binary code from the device, it may be possible to analyze its behavior and, with sufficient effort, identify vulnerabilities. Nevertheless, depending on the device, if its functionality and role within different systems is not significant enough, an attacker’s motivation may not be sufficient to go through this reverse-engineering effort.

If the device provides a user interface, it is possible that an attacker targets this interface and tries to alter the behavior of the device. E.g. buttons or switches can easily be used by a person who has physical access to the device. The wiring and connections of a device may also be the target of tampering, if they are accessible.

## 4.3 Embedded devices with real-time operating systems

Most embedded field devices utilize some real-time operating system (RTOS), of which there exist a large number of different implementations. The operating systems enable multi-tasking and necessary support functions to the device firmware. Documentation on the different systems may be available in varying amounts; however the generic operating system theory applies to virtually all systems.

Because these operating systems are part of the embedded firmware, the direct modification of the device binary is not an easy task. However, if one succeeds in e.g. reading the binary code of the device, it may be possible to detect text strings stored in the code which contain clues as to which operating system is used. Other text strings may also disclose software components which are used or version numbers.

Similar threats relate to the RTOS-based devices as simple field devices, however these slightly more advanced systems may be of higher importance in a control system and thereby the motivation to attack these devices with RTOSs may be higher.

It is common that some methods of diagnostics or debugging are left enabled in the devices, for the purpose of e.g. field troubleshooting. These interfaces present a possible way for studying the device and its functionality.

## **4.4 Embedded devices with general-purpose operating systems**

### **4.4.1 Operating system vulnerabilities**

Due to the complex nature of operating systems and the software environment executed on top, vulnerabilities do and will continue to exist. Sometimes the vulnerabilities are found in the operating system kernel itself, however in many cases the vulnerability is caused by the different services and applications which are executed using the services of the operating system.

By the word “vulnerability” is meant an error (e.g. due to misspecification, incorrect implementation or simply a human mistake) which can be used by an attacker to perform malicious activity on a system or network. Different kinds of activity may be possible, for example executing code or commands, accessing restricted data without proper authorization, or conducting denial-of-service attacks.

It is common knowledge that general-purpose operating systems (such as Microsoft Windows or Linux-distributions) are targeted by malware. One reason for targeting these systems is their popularity; if a malware can be targeted at a widely used system, the potential impact and associated achievable notoriety for the creator is higher. Also operating systems intended e.g. for mobile phones and smartphones, such as Symbian or Android, have been targeted by malware.

### **4.4.2 Open-source systems**

Open-source systems may, from one perspective, be considered at risk as the source code is readily available to anyone, including malware creators. Thus, with proper analysis it is possible to detect vulnerabilities based on the source code

analysis and modeling of different scenarios. Furthermore, online issue and vulnerability tracking systems may highlight the detected issues, sometimes also presenting which version of the system includes the problem and when it has been corrected.

This information can be helpful in tracking the history of security issues in a system; however this kind of log can also be used by malware creators to make the malicious software adapt itself based on the installed version of an operating system. Nevertheless, nowadays it is very common for exploit researchers to make binary differences e.g. between patched and unpatched Windows drivers. Therefore, it is equally possible to reconstruct vulnerability under Windows as under Linux.

On the other hand, the open-source community provides a wealth of competence and resources which review documentation and source-code, while also actively contributing to the development including correction of identified issues. The open-source nature of a system should thus not be viewed as a pure disadvantage, as long as the backing community of developers and stakeholders is competent and capable of maintaining it.

### **4.4.3 General-purpose operating systems**

The experience from desktop operating systems has shown that all kinds of operating systems have vulnerabilities regardless of open source or closed source. Microsoft Windows for instance has been improved by Microsoft through constant and serious effort, because it has been shown to not be secure enough. Current Microsoft Windows versions have a significantly superior level of security compared to older versions. Linux implemented early important security features like division of privileges, whereas under Windows until

recently the default user administrator privileges had. Because of the threats, things have been improved on both sides, and it is difficult to state that Linux is more secure than Windows or vice versa.

Both, Windows and Linux are implemented in embedded devices. Knowing that is relatively easy to update desktop operating systems, this might not be true for embedded operating systems. An update would be very difficult if the concerned device is running in a production environment and cannot be rebooted without interrupting an entire factory. Moreover, not every field device has a permanent Internet or network connection. A manual update procedure would be necessary and depending on the number of affected devices this requires a considerable amount of time.

Vulnerabilities are provided to the operating system vendors from a lot of different sources on a weekly basis. General-purpose operating systems do provide a lot of advantages, but if an attacker is able to find a single vulnerability in it, he may exploit every device running the specific system. Therefore, somebody needs to take care, that field devices employing a general purpose operating system are constantly kept up-to-date. To not use a general purpose operating system and write from scratch a custom operating system might, depending on the complexity, result in fact in a far less secure system (also called security by obscurity).

When the operating system of a device is updated, the device usually needs to be rebooted. But in some cases updates can create conflicts with former software employed in the device. Therefore, extensive testing needs to be applied before updating productive systems in order to keep the negative effect of the downtime as small as possible.

## **4.5 Recommendations for enhancing security in devices**

In general it is very difficult to protect a device, once the attacker is able to gain physical access to it. If possible the disassembly of the device should be made more difficult, nevertheless if an attacker has sufficient time and equipment this does not hinder him from analyzing an obfuscated device.

In the design of devices, security needs to be addressed at all levels and in different scenarios. Using an attacker's perspective, enhancing security in the device should be a continuous process about identifying weakest links and strengthening them. Threats to a product or product family should be modeled and appropriate measures to correct these determined and implemented.

### **4.5.1 Debugging interfaces**

Interfaces for debugging and manufacturing (e.g. JTAG or similar, serial ports for text output) shall not be utilizable in the production version of a device. The interfaces may be rendered disabled either by not assembling electrical components (e.g. zero-ohm jumpers) or by security functionality in the ICs in the product. A debugging interface which needs to be enabled in a field-installed device could e.g. be protected using cryptography, so that authentication using a key is needed before the interface opens. Authentication keys need to be stored securely, protected from the attacker.

### **4.5.2 Communication interfaces**

Many devices feature one or more communication interfaces. All such interfaces need to be appropriately protected against malformed or bad messages, preventing issues such as buffer overflows or crashes from occurring. Communication protocols which are

not intended for the end-user shall not be enabled in a device.

Detailed information about the device, or information revealing the technical implementation of the device, shall not be communicated outside the device unless absolutely necessary. Secret or critical information, which needs to be communicated outside the device, must be encrypted with a strong algorithm and a good key. Such keys need to be stored securely so that they cannot be accessed by the attacker.

Communication interfaces to a PC program may be targeted by an attacker, either by monitoring the traffic on a serial, USB or Ethernet link, or by reverse engineering the PC program itself. By studying the program or the communication, the attacker may detect commands which are hidden from the user interface of the PC program, or may apply existing commands in new ways. Monitoring e.g. the serial port communication is easy using available PC programs. It is also possible that the user records the communication and replays select portions in order to decipher the functionality of the device.

If the communication interfaces of a device integrate some security features (e.g. reminiscent of a firewall in Ethernet applications), then the default configuration should be that all access to the device is prevented. The purpose is that only the smallest required number of changes to the security configuration is made in order to allow the system to work as expected.

### **4.5.3 Firmware protection**

The binary firmware of a product can be protected from being extracted e.g. by encrypting the non-volatile memory in which the firmware is stored. This approach presents some challenges in terms of decrypting the firmware securely at start-up, and storing the keys so that they cannot be extracted from the device. By

protecting the contents of e.g. a flash memory from being extracted, attempts to reverse-engineer a device are no longer trivial.

Sometimes there may be a need to prevent the copying of binary firmware from one product to another. Since it is not uncommon for different products belonging to the same product family to use a single piece of hardware, with features being disabled in software, the copying of binary from a feature-rich product to a low-cost product can be motivated by financial gain, or enabling features which have been disabled from the entire product family.

Another approach to preventing firmware from executing in an unauthorized device is to implement a kind of license management. This could e.g. be design so that a device needs an unforgeable serial number which corresponds to authorizing a certain firmware to execute in the device. This mechanism needs protection against tampering using cryptography and secure management of the cryptography keys, serial numbers and firmware packages.

There are different ways to prevent this copying, such as the above mentioned encryption of firmware and license management methods. Creating different hardware is another option; perhaps the most effective way is to implement the feature-restricted product using a binary-incompatible processor or microcontroller. Thus, the feature-rich binary cannot be copied to the feature-restricted product; as the binary is incompatible it will not execute.

If a product does not have encrypted firmware storage, but it is introduced to the next generation of the product, this likely also constitutes a kind of hardware differentiation, as the security keys needed for encryption may need a different model of processor, microcontroller or memory to be used.

#### 4.5.4 Device parameters and configuration

The parameters and configuration of a device can be protected against modification e.g. by defining user access levels with password-protection. Such mechanisms shall require that the password configured for the different user levels have sufficient strength. As the device most likely contains a default password, the device shall encourage (possibly even require) the user to change the default password. It is especially important is that passwords are not hardcoded into the device.

Only proper user and software authentication can really create a level of security for automation field devices. The Stuxnet case has shown that default passwords undermine many security measures, therefore the field devices should not be possible to be configured with default passwords.

The storage of the passwords should be protected, potentially using encryption so that plaintext is not present. If the passwords need to be communicated between ICs, the passwords should be communicated in ciphered mode.

#### 4.5.5 Firmware updating

The updating of firmware in the device needs to be secure. A major security issue arises if an attacker is able to insert a modified firmware into a field device; it would be possible to control how the device is functioning. Several techniques to mitigate this risk can be considered.

The device should detect if firmware has been modified, e.g. using error checking algorithms in a bootloader; in this case the system should go to a safe state and not execute the incorrect firmware.

The firmware needs to be encrypted. An attacker would not be able to read an encrypted firmware

and therefore it is nearly unfeasible to alter it. In this case, the secret key of the field device should be stored securely and to the possible extent changed regularly. Therefore, encryption solely cannot be seen as a sufficient security measure. If possible, proper key management is needed. Key management provides functionality to exchange existing keys and also revoke keys which gets compromised. Moreover, when encryption is used no Master key should be employed. Basically, this means that every device should have its own key, thus limiting the damage from disclosure of a single key.

If possible, firmware updates should be digitally signed with a private key. The corresponding public key can be stored in the field device (for instance in a read-only memory, ROM) and used in order to verify legitimate firmware updates.

No “Master Tool” should be available, which would allow modification and tuning of the field device, without proper authentication. Furthermore, if a field device is modified by the responsible controlling software, this software should be able to log this activity. Additionally the logging should not be done exclusively on the local machine, but should be sent to a central server which keeps track of all modifications.

#### 4.5.6 Superfluous information

There is often a lot of extra data related to a product or device which is needed e.g. only during the development stage. There is a risk that this kind of information is left in the production version of a product, even though it is not needed for the principal functionality of the device. This presents a risk in terms of information leakage, which may further lead to risk of tampering. It is therefore essential that the amount of information, which is not related to functionality, is minimized in the device.

## 5 Security in wireless communications

In computer networking, wireless communication such as WLAN has been used for many years. For close-range wireless communication, technologies such as Bluetooth or Zigbee are popular choices. Wireless technology is also entering the industrial automation market, with technologies for Wireless HART, Ethernet or PROFIBUS.

The International Telecommunication Union (ITU) has reserved special frequency bands for industrial, scientific and medical usage, commonly known as ISM bands. Many current wireless approaches to industrial automation rely on IEEE 802.15.4 “Low rate WPAN” technology, or on Bluetooth technology, which are capable of operating at one or more ISM bands. Bluetooth and 802.15.4 technologies are both capable of operating in the 2.4 GHz ISM band, however 802.15.4 can also operate in the 902-928 MHz ISM band. Wireless LAN, based on IEEE 802.11, can also operate in the 2.4 GHz ISM band, although higher frequencies are also available for this communication.

IEEE 802.15.4 and Bluetooth are designed for short range communications with ranges around 10-20 meters. IEEE 802.11, depending on the protocol used, offers ranges in excess of 50 meters indoor and over 100 meters outdoors. The data rates of the technologies differ, as does the transmission range, due to design objectives such as low power consumption and low data rate. 802.15.4-based devices have very low power consumption, reducing the achievable data rate and transmission range, while the high data rate and long range of 802.11-based devices translates into high power consumption.

### 5.1 Security of wireless technology

Because it is not necessary to have physical access to a wire or network, wireless systems are vulnerable targets to eavesdropping and possibly also tampering. Additionally, some networks are so-called ad hoc networks in which nodes participate in routing and data forwarding, and thus cannot rely on fixed devices acting as routers. Thus, connections may need to be established to unfamiliar devices. Because in some cases, it is desirable to reduce computational capacity and thus power consumption, the feasibility of cryptographic algorithms and protocols may in some applications be limited.

Wireless links can be targeted by jamming attacks in the form of a device transmitting at a very high power, with the target of disrupting authorized communication in a system. By jamming is commonly meant that a device deliberately transmits different kinds of signals into a busy medium, decreasing the signal-to-noise ratio of the on-going communication. This corresponds to a denial-of-service attack. The possibility to succeed with this kind of attack depends on the network, the used technology and devices, the jamming device and the overall environment. Some wireless technologies implement features to improve the network performance in case the medium contains a lot of interference; as an example, IEEE 802.15.4 provides dynamic channel selection which allows the device to select the best available channel for operation. A device attempting to overcome this flexibility of the network needs to be able to jam over a wide spectrum, which contributes to cost and complexity in the jammer. However, given enough resources and motivation, this can also be achieved.

### 5.1.1 IEEE 802.15.4

The IEEE 802.15.4 standard is the basis of some commonly used communication protocols such as WirelessHART, Zigbee and 6LoWPAN. The MAC layer of the IEEE 802.15.4 specifies a security mechanism which allows encryption and decryption of the messages which are communicated between devices; however this security feature is optional. The security mechanism is designed to be flexible, allowing the application of the cryptography only on required frames, or with certain nodes in the network. This allows the extra overhead of security to be avoided in such cases where it is not deemed necessary. The firmware in the application layer of a device specifies the level of security required by setting control parameters in the stack; per default no security is enabled.

The WirelessHART protocol uses the 128-bit AES encryption which is provided by the 802.15.4 compliant devices and does not allow it to be disabled. Apart from protecting the data contents of the message, it also protects the network/transport layer information of packets so that the routing of packets is correct.

The Zigbee protocol specifies additional security services at the network and application layers, using symmetric cryptography to protect the frames and authorize devices. Keys are managed by a central device and are either pre-installed into devices, transported using secure communication, or established without being directly transported between devices.

### 5.1.2 Wireless LAN

The security of Wireless LAN (WLAN) networks is a key consideration in IT systems management, however as it is becoming increasingly common to have Industrial Ethernet networks connected to WLAN stations. Unless security is properly addressed also in these

configurations, the industrial communication network and control system is left highly vulnerable.

Sometimes, WLANs are configured with no or poor security measures. Additionally, the physical security around the WLAN, and the possibility to access the WLAN from outside e.g. the building in which it is setup, increases the risk of intrusion. Furthermore, employees changing the network configuration or setting up weakly protected access points poses further risk. Insufficient security measures, such as weak encryption standards or reliance on poor filtering, leave holes in the security which may provide a false sense of security; while only deter hackers with basic knowledge, it enables advanced attackers to access the system.

### 5.1.3 Bluetooth

Bluetooth devices must be paired before connections can be made; this is a basic requirement for protecting private data. Once a pair of devices has been authorized to connect, they can be configured to do so without future user intervention. Basic protection which can be configured by the user includes making the device non-discoverable, preventing the device from showing up on a Bluetooth device scan. A non-discoverable device can theoretically be detected using brute-force methods; however this will generally require too much time to be practical. Some devices can also be configured as “non-connectable”, meaning that not even paired devices can establish a connection to this device.

The Bluetooth specification addresses the problem with passive eavesdropping and man-in-the-middle attacks by specifying “Secure Simple Pairing” of devices, which is mandatory in devices with Bluetooth version 2.1 or higher. The cryptography of the pairing procedure is enhanced, without affecting the user intervention which is needed. This improvement reduces the

possibilities for a device sniffing the Bluetooth communication to be able to determine the credentials used by a device to establish a pairing.

User ignorance, typically in the case of mobile phones or PDAs, presents a significant security risk in that devices may be left in discoverable mode. Users may also be the subject of a “social engineering” attack, such as proposed in [8], in which they unsuspectingly accept an incoming pairing request. This could expose their sensitive data or put them at risk for sabotage.

Additionally, numerous security issues in the form of Bluetooth viruses or worms, and operating system or application vulnerabilities have been identified. The exploitability of these vulnerabilities varies, however the exploits may cause slow operation or application or device crashes. The effect of such an error in an industrial control system can be considerable.

## **5.2 Recommendations for improving wireless network security**

Due to the broadcasting nature of wireless communications, physical access to the communication network is no longer a necessity. If an attacker is within range of the wireless link, it is possible to start an attack from outside the boundaries of physical access. Therefore, security measures shall focus on configuration of the network and devices so that an attacker cannot exploit vulnerabilities, and to ensure that there are multiple security barriers protecting the system. Thus, even if the attacker breaks some of the barriers, he or she does not easily gain access to the network.

This is not to say that physical security and access can be ignored. Both wired and wireless networking equipment must be protected, so that an intruder cannot easily access the network

from inside the building or premises. Unused ports in networking equipment shall be locked to prevent unauthorized connection to these.

It should be a priority to ensure that employees or other persons with access to the organization do not affect the network configuration, either by modifying existing devices or by connecting and setting up new networking equipment. There exists a risk that new networking equipment introduced to the network has less stringent security configuration, thus exposing a weak link to potential attackers.

From a management perspective, it is imperative that the organization defines security guidelines, policies, arranges training for staff and third parties, and establishes a security “culture”. This culture shall include continuous assessment, feedback and actions to maintain security in the organization. It is essential to recognize the people in this process, and to ensure that they can voice their concerns and that their understanding of security is aligned to the security policies and guidelines as necessary.

It is important to configure the wireless network correctly. As has been outlined above, the first step is to ensure that security modes are enabled, if they are not by default. Where used, passwords must be selected so that they have good strength. Where the encryption modes are selectable, and there is a reason not to use the best available encryption (e.g. reducing power consumption in wireless nodes) the ease of cracking the encryptions must be weighed against the tolerated risk. Then, the appropriate encryption is taken into use.

In applications where Bluetooth is used as the wireless protocol, the devices should be configured with the lowest possible amount of visibility to non-familiar devices. Preferably, devices should be non-discoverable if they need not be paired with new devices.

## 6 Summary

This paper has provided an overall introduction into the information security of industrial automation. The current state of standardization has been reviewed and compared to the level of information security standardization in the ICT field. Additionally, the security issues in inter-device communication and in field devices alone have been discussed, followed by a brief overview of the security issues in wireless interfaces.

Security in industrial automation has got much more attention after the Stuxnet case. Measures to improve and implement security have been started before Stuxnet, but thoughts and attitudes have now changed. It has been shown in this paper that traditional fieldbuses are missing security aspects in many ways. There are no standardized rules and requirements for field devices against security threats like tampering, hacking and similar activities. This means that it will be difficult to implement security measures in lower automation levels by retrofitting or re-engineering current communication standards in the short term.

Therefore, it is essential that end-users and owners/operators apply feasible cyber security programs in their processes and plants to get protection against attacks via upper automation levels and to restrict physical access in affected zones. Existing practices of ICT business and already published industrial standards and guidelines can be used as a foundation for this work.

A security program shall be a de facto – requirement for new installations and plants. Risks shall be identified and preventive measures shall be applied through the delivery chain from customer requirements to system integrators and device vendors. Also installation, commissioning, maintenance, service and

operator staff needs to follow proper procedures in order to ensure that the security of customer operations is not at risk.

Field level communication developers and field devices manufacturers need to place more focus on security in future development. There are easier and more difficult areas to handle, but as this Pandora's box is once opened there is no turning back. Transparent and quantitative methods are required and on-going standardization work needs strong support from all sides. The automation business has credible history solving challenges e.g. in safety, EMC and functional safety – it's time for security now.

## 7 References

- [1] F-Secure Corporation Terminology.  
[http://www.f-secure.com/en/web/labs\\_global/terminology](http://www.f-secure.com/en/web/labs_global/terminology)
- [2] ITU, ITU-T Study Group 17 – Security.  
<http://www.itu.int/ITU-T/studygroups/com17/index.asp>
- [3] ISO, Telecommunication standardization committee.  
<http://isotc.iso.org/livelink/livelink/open/jtc1sc6>
- [4] ISO, Technical committee: Security.  
[http://isotc.iso.org/livelink/livelink/open/jtc1\\_bp](http://isotc.iso.org/livelink/livelink/open/jtc1_bp)
- [5] IEC Technical Committee 65.  
[http://www.iec.ch/dyn/www/f?p=103:7:0::::FSP\\_ORG\\_ID:1250](http://www.iec.ch/dyn/www/f?p=103:7:0::::FSP_ORG_ID:1250)
- [6] ISA, ISA99, Industrial Automation and Control Systems Security.  
<http://www.isa.org/MSTemplate.cfm?MicrositeID=988&CommitteeID=6821>
- [7] Åkerberg, J. & Björkman, M. (2009). *Exploring Security in PROFINET IO*.  
33<sup>rd</sup> Annual IEEE International Computer Software and Applications Conference.
- [8] Symantec Bluetooth Security Review.  
<http://www.symantec.com/connect/articles/bluetooth-security-review-part-1>

## Biographies

**Arthur Gervais** is a Junior Security Consultant at Nixu. Currently finishing his double degree as Master of Science in IT-Security, he has experience especially in network security. In 2011 he was awarded the Best Student Award from the German Federal Office for Information Security (BSI).

**Mikko Hyppönen** is Chief Research Officer at F-Secure Corporation. He has studied thousands of virus cases during the last 20 years. He has published texts on his research findings in several international publications, including Scientific American, The New York Times and CNN.com.

**Janne Kuivalainen** is Director, Control Platform and Products at Vacon Plc. He has been working over 10 years with embedded device related R&D and in power plant engineering in the 90's. He is member of SESKO/SK65 - national electrotechnical standardization committee for industrial-process measurement, control and automation in Finland.

**Juhani Mäkelä** is a lead developer in the Development and Nixu Open divisions of Nixu. He has a long experience of working with embedded systems in the telecommunication and commercial areas. The last four years he has been developing a mobile security platform for Linux-based smartphones.

**Jouko Orava** is Manager, Control Platform and Architecture at Vacon Plc. He has almost 20 years experience of frequency converter development in areas of embedded software and fieldbuses. The last years he has been developing embedded systems and architectures for frequency converters.

**Magnus Sundell** is a Development Engineer at Vacon Plc. He holds a M.Sc. degree in Computer Science and has long experience developing embedded systems for industrial communication. His main tasks are software architecture and requirements management in industrial connectivity.