Safety starts with development

Frank Eberle, Product Development, Pilz GmbH & Co. KG,
Ostfildern, Germany

As the level of networking in industry rises, so too does system vulnerability. So manufacturers of automation components must take appropriate measures.

IEC 62443-4-1 "Security for industrial automation and control systems – Part 4-1: Secure product development lifecycle requirements" takes the "Security By Design" approach. However, during development if a manufacturer complies with the requirements of the standard IEC 61508 "Functional safety of electrical/electronic/programmable electronic systems" from the outset, the requirements of IEC 62443-4-1 can be met more easily, if they are not already met in full.

"Security" is understood to be the guarantee of the protection objectives confidentiality, integrity and availability. The security requirements of the worlds of IT and automation differ significantly. Confidentiality of information has the highest priority in an office environment, whereas data availability and integrity is most important in the production area. On the one hand it is a major prerequisite for smooth manufacturing processes; on the other hand, an attack on the integrity of a safety system can result in major accidents. That's why Edition 2.0 of the standard IEC 61508-1 includes an addendum in clause 7.4 "Hazard and risk analysis". This says that a threat analysis should be carried out if a security threat is regarded as "reasonably" foreseeable. So manufacturers of safety systems in particular must address the issue of security. However, even manufacturers of systems that do not implement safety-related functions should deal with security in order to prevent attacks against production processes.

Demand for streamline security

At the moment, security on production plants is often implemented through security components such as firewalls and VPN gateways. Communication relationships between automation components and with external systems will increase due to I4.0 and IoT. This will result in higher costs for managing external security solutions. When wireless technologies are used, firewalls only offer limited protection because an attack could occur directly via the wireless interface and the lower protocol layers To counter these problems, security measures must be implemented directly within the systems.

What does “Security By Design” mean?

The term “Security By Design” or “Secure By Design” describes a development approach in which a system’s security features are considered systematically as early as the design phase. So the extent to which security functions are implemented in a system is not left to chance or to the assessments of individual developers. Instead, threat modelling is used to determine the threats to which a system is exposed. From here it is possible to work out targeted measures in order to minimise the security risk.

Under a broader interpretation, “Security By Design” can also be seen as an approach in which the security of a product is considered holistically over the complete product lifecycle. A much quoted and well documented example for this approach is the “Security Development Lifecycle (SDL)” process developed by Microsoft. At the beginning of the 2000s, the negative headlines referring to security problems in Microsoft products were starting to accumulate. This prompted the company to address the issue of security systematically, leading to the development of SDL. Many other software and device manufacturers now follow a similar approach.

Standards as the starting point

Security has long played a central role in classic IT. However, due to the different priorities with regard to confidentiality and availability, it is not just a simple process of transferring the requirements to automation.

In contrast, IEC 62443 “Industrial communication networks – Network and system security” is an international standard series that deals comprehensively with IT security in automation; some parts have already been adopted. The issues range from risk analysis to best practices and security by design. As such, IEC 62443 currently offers the best orientation guide for plant operators and device manufacturers when it comes to implementing security effectively.

IEC 62443 was defined originally by the ISA99 committee “Industrial Automation and Control Systems Security” and was adopted by the IEC standards committee. The graphic below shows the individual parts of the standard, which are divided into four areas:

IEC 62443: Industrial communication networks – Network and system security

General

- 1-1 Terminology, concepts and models
- 1-2 Master glossary of terms and abbreviations
- 1-3 System security compliance metrics
- 1-4 IACS security lifecycle and use-case

Policies & Procedures

- 2-1 Requirements for an IACS security management system
- 2-2 Implementation guidance for an IACS security management system
- 2-3 Patch management in the IACS environment
- 2-4 Security program requirements for IACS service providers

System

- 3-1 Security technologies for IACS
- 3-2 Security risk assessment and system design
- 3-2 System security requirements and security levels

Component / Product

- 4-1 Secure product development lifecycle requirements
- 4-2 Technical security requirements for IACS components

The specific requirements are subdivided into groups, called “Practices”. These are described briefly below.

Practice 1 – Security management

This practice lays down various requirements of the management of the development process. These include the existence of a general product life cycle process, the need to identify responsibilities and the need to train staff in terms of their role in the process and their security expertise. There are also requirements regarding the security of the development environment and handling subcomponents supplied by third-party suppliers.

Practice 2 – Specification of security requirements

As the name suggests, this practice describes the requirements for specifying security requirements. For example, “Product security context” stipulates that the system context from a security perspective must be defined for the system being developed. The context describes, for example, the physical security features of the environment (e.g. the system must be operated in a locked cabinet) and the features of the network environment (e.g. the system must be protected by a firewall). The security context is an important input variable for threat modelling, which is another requirement that is demanded.

Practice 3 – Secure by design

This deals with the requirements of the system design. For example, recognised security techniques and design models such as “Security by Design” are to be used. So the initial interpretation of the term “Security by Design” is implemented through these requirements.

Practice 4 – Secure implementation

The requirements of the practice “Secure implementation” are intended to ensure that no security vulnerabilities arise due to implementation errors. They include compliance with recognised coding principles, plus the implementation of a static code analysis and code review.

Practice 5 – Security verification and validation testing

This practice establishes the type of security tests to be carried out. It also states the requirements for the independence of testers.

Practice 6 – Management of security-related issues

The practice management of security-related issues describes the requirements for managing security issues within the product. In addition to requirements for problem analysis, this also includes the receipt of messages (e.g. from customers or security researchers) and the notification of users when security issues are discovered.

Practice 7 – Security update management

This deals with the requirements of managing security updates. This includes, for example, ensuring that an update will actually rectify a vulnerability as intended and will not cause any new issues. There is also a requirement for manufacturers to inform users as to whether security updates can be installed on dependent components (e.g. the operating system on which the product is used) without repercussions.

Practice 8 – Security guidelines

This practice defines the requirements for the content of the user documentation. For example there is a requirement to outline the measures required to harden system security and the considerations to be made when decommissioning the device.

Using synergies and adapting process

When you consider IEC 62443-4-1 with its 47 individual requirements, implementation appears extremely complex. This is particularly the case when product development has previously followed a “head to keyboard” approach, in other words, it has not been based on appropriate processes. In this case it is important first of all to implement the requirement SM-1: “Development process”. This states that a general lifecycle process must be documented and applied. However, if such a process already exists and if this considers functional safety requirements in accordance with IEC 61508 or one of the industry-specific standards derived from it, then it is possible to identify similar or identical requirements that are already being implemented. The requirement SM-5 “Process scoping” also states that the process is to be adapted to the respective development project, based on a security analysis. So individual requirements or part requirements do not need to be considered if a system does not have any external interfaces, for example, or if a change project only involves updating language catalogues or replacing discontinued components.

Examples of commonalities and synergies

Where specifically are synergies to be found when implementing IEC 61508 and IEC 62443-4-1?

Example 1: Staff training

The requirement SM-4: “Security Expertise” states that there must be a process for identifying training requirements and for training staff, so that they can correctly fulfil their roles and responsibilities within the security process. Clause 6 of IEC 61508-1, “Management of functional safety” formulates comparable (if more detailed) requirements in sections 6.2.12 to 6.2.15. The specific knowledge that must be communicated for safety and security will differ, but issues such as “defensive programming” or application of the familiar MISRA programming standard (Motor Industry Software Reliability Association) from the automotive sector are equally important to both areas, so content synergies will result.

Example 2: Coding

Practice 4 of IEC-62443-4-1 – “Secure Implementation” defines two requirements:

1) Requirement SI-1: “Security implementation review“ requires that a process be employed to perform code reviews in order to check various aspects at coding level. Typically this is understood to include a review of the source code by one or more persons. There is also a requirement to use appropriate software tools to perform a static code analysis. Comparable requirements can be found in IEC 61508-3. Section 7.4.6 “Requirements for code implementation” stipulates that each software module must be tested. Reference is made to clause C.5.14 “Formal inspections” and C.5.15 “Walk-through (software)” from IEC 61508-7. Anyone who has already implemented these requirements to achieve functional safety in their development process need no longer worry about how to organise and document such reviews or how to manage any discrepancies that are identified. Naturally it will still be necessary for people with security expertise to take part in the review. Table A.9 of IEC 61508-3, “Software verification”, also recommends statistical analysis as a measure. The computer-aided approach is stated as an option in clause B.6.4 of IEC 61508-7.

2) Requirement SI-2: “Secure coding standards” stipulates that programming rules must be defined to avoid security errors and that these rules should be checked and updated periodically. Here too, there is a comparable requirement in IEC 61508-3. Clause 7.4.4.12 stipulates that the programming languages should be used in accordance with suitable standards. Reference is made to IEC 61508-7 with regard to the content of such rules. Appropriate instructions can be found there in clause C.2.6 “Design and coding standards”. One aspect of such rules is to avoid programming errors. They represent a general quality issue and, in an extreme case, can impact both safety and security. Typical programming errors such as memory overflow and a missing input data check represent a risk to safety. Equally, however, they are also the

classic errors seen in security programming. That’s why an existing set of rules that has been created in order to achieve functional safety is also a good start when it comes to security.

Example 3: Managing issues

In Practice 6 – “Management of security-related issues”, various requirements are laid down for how to deal with security issues that are identified. There is a requirement to have in place processes for receiving security messages (e.g. from customers or security researchers), for notifying affected users, analysing the reported issues and avoiding similar problems in future. Here too there are comparable requirements in Clause 6 of IEC 61508-1, “Management of functional safety”, so any company that implements these requirements will already have suitable processes in place, which it may be possible to apply unchanged to the management of security issues.

Using these examples it becomes clear that processes used to meet the requirements from IEC 61508 are also suitable for implementing the requirements from IEC 62443-4-1, or require only a few minor enhancements. So the challenge for secure product development lies less in the definition of suitable processes and more in the technical area. Any manufacturer wishing to address the issue of “Security” must ensure that everyone involved in the development has sufficient technical knowledge.

Security needs addressing over the whole product lifecycle

Another huge challenge is that security is a “moving target”: an automation component such as a PLC can be classed as “secure” one day, but the next day the threat level may change, impacting on the device’s security from attack. This means that measures against cyber threats must be constantly updated. Responsibility lies primarily with plant operators, for whom data security equals investment protection.

Conversely, machine builders and component manufacturers also have a responsibility to inform operators immediately about any new security issues, and to provide updates for the software on their devices to rectify any vulnerabilities. However, this means that both sides have to work closely together over the whole lifecycle of the products.

Responsible manufacturers should address the issue of security if their products are to be part of a digitalised world. This is particularly valid if these products need to provide protection in the sense of safety. This means that although these two aspects of automation continue to be independent, they must be closely aligned. The good news is this: anyone who is already well versed in safety will have an easier time with security, because the procedures are similar.

Author: Frank Eberle, Product Development Pilz GmbH & Co. KG