# Sino-German White Paper on IT Security Tests for *Industrie 4.0* and Intelligent Manufacturing

Sino-German Intelligent Manufacturing /
*Industrie 4.0* Standardisation Sub-Working Group

IMS 国家智能制造标准化总体组 National Intelligent Manufacturing Standardisation Administration Group
STANDARDIZATION COUNCIL INDUSTRIE 4.0
PLATTFORM INDUSTRIE 4.0
GPQI Dialogues for Innovation and Trade

**Published by**
Deutsche Gesellschaft für
Internationale Zusammenarbeit (GIZ) GmbH

**Registered offices**
Bonn and Eschborn, Germany

Global Project Quality Infrastructure

Tayuan Diplomatic Office Building
No.14, Liangmahe Nanlu, chaoyang District
100600 Beijing, PR China
E info@gpqi.org I www.gpqi.org

**Design**
Oliver Hick-Schulz / Elena Reiniger

**Photo credits**
Adobe Stock / Vladimir

**On behalf of**
German Federal Ministry for Economic Affairs and Climate Action (BMWK)
Berlin, Germany 2022
Beijing, China 2022

**Text**
Standardization Council Industrie 4.0
German Commission for Electrical, Electronic & Information
Technologies of DIN and VDE
60596 Frankfurt am Main

National Intelligent Manufacturing
Standardisation Administration Group
China Electronics Standardization Institute,
No.1 Andingmen East Street,
Dongcheng District, Beijing, 100007, China

**Authors/Experts**
DR. ROBERT ALTSCHAFFEL, Otto-von-Guericke-University Magdeburg • YILMAZ CANKAYA, Siemens • PROF. JAN DE MEER, HTW c/o smartspacelab.eu • PROF. DR. YONGJIAN DING, Magdeburg-Stendal University of Applied Sciences • YUAN GAO, IBM • YUN GUO, Huaneng Shandong Shidao Bay NPC • ERKIN KIRDAN, TUM • DR. LIN LI, CESI • JOCHEN LINK, ING-LINK Engineering Office • DR. XINXIN LOU, University of Bielefeld • PROF. DR. CHRISTOPH RULAND, University of Siegen • DR. JOCHEN SASSMANNSHAUSEN, University of Siegen • PETER SIEBER, HIMA • MARTIN SZEMKUS, ICSS • DR. KARL WAEDT, Framatome GmbH (DEU lead expert) • DR. VENESA WATSON, General Electric • XIN XIE, accenture (formerly Siemens) • TIANZHE YU, ifak • LIANG ZHANG, Framatome GmbH • DR. ZITONG ZHAO, CESI (CHN lead expert)

The German Federal Ministry for Economic Affairs and Climate Action (BMWK) has commissioned the Deutsche Gesellschaft für Internationale Zusammenarbeit (GIZ) GmbH to implement the Global Project Quality Infrastructure (GPQI).

Implemented by

Federal Ministry
for Economic Affairs
and Climate Action

giz Deutsche Gesellschaft
für Internationale
Zusammenarbeit (GIZ) GmbH

This publication is a result of the Sub-Working Group Industrie 4.0/Smart Manufacturing of the Sino-German Standardisation Cooperation Commission in cooperation with

中华人民共和国工业和信息化部

SAC 中国国家标准化管理委员会
Standardization Administration of the P.R.C.

# Contributors

**NATIONAL INTELLIGENT MANUFACTURING STANDARDISATION ADMINISTRATION GROUP**
The National Intelligent Manufacturing Standardisation Administration Group (IMSG) was established to promote and accelerate the progress of intelligent manufacturing in China under the leadership of the Standardisation Administration of China (SAC) and Ministry of Industry and Information Technology (MIIT). It is responsible for carrying out practical work on intelligent manufacturing standardisation, including participation in international standard-setting on intelligent manufacturing as well as organising exchange and cooperation on international standards.

**STANDARDIZATION COUNCIL INDUSTRIE 4.0**
The Standardization Council Industrie 4.0 (SCI 4.0) was founded at the Hannover Messe 2016 as a German standardisation hub by Bitkom, DIN, DKE, VDMA and ZVEI. The initiative aims to initiate standards for digital production and to coordinate these standards nationally and internationally. SCI 4.0 orchestrates implementation of the standardisation strategy of the German Platform Industrie 4.0, which includes coordination with standardisation organisations (SDOs) and international partners as well as interlocking with pilot projects. The aim of this coordinated approach is to ensure that standards exploiting the potential of Industrie 4.0 are developed in a coordinated manner. SCI 4.0 is supported by DKE and the German Federal Ministry for Economic Affairs and Climate Action (BMWK).

**PLATTFORM INDUSTRIE 4.0**
The Plattform Industrie 4.0 is the central network in Germany for advancing the digital transformation in production. More than 350 stakeholders from more than 150 organisations are actively involved in the Plattform, in close cooperation between politics, business, science, trade unions and associations. As one of the largest international and national networks, the Plattform supports German companies in implementing Industrie 4.0, especially by making existing Industrie 4.0 practical examples known to companies and bringing them into the mainstream. In addition, it provides important impetus, with concrete recommendations for action in over 200 specialist publications and refers to support services and test environments. The Plattform's numerous international collaborations underline its strong role in international discussions on the topic of Industrie 4.0.
You can find more information at **www.plattform-i40.de**.

**GLOBAL PROJECT QUALITY INFRASTRUCTURE**
The German Federal Ministry for Economic Affairs and Climate Action (BMWK) established the Global Project Quality Infrastructure (GPQI) to promote the development of well-functioning and internationally coherent quality infrastructures. GPQI supports political and technical dialogue and implements bilaterally agreed activities in collaboration with all relevant stakeholders. The project aims to reduce technical barriers to trade and enhance product safety through bilateral political and technical dialogue on quality infrastructure (QI) with some of Germany's key trading partners.

# Contents

# 1. Introduction

The 6th meeting of the Sino-German Intelligent Manufacturing 2025/Industrie 4.0 Standardisation Working Group (hereinafter referred to as the Working Group) was held in Heidelberg, Germany, from 27 - 29 June 2018. The conference was organised by the German Federal Ministry for Economic Affairs and Climate Action (BMWK, formerly the German Federal Ministry for Economic Affairs and Energy). At the meeting, Sino-German partners agreed to propose the white paper on current concepts of security testing for Industrie 4.0/Intelligent Manufacturing.

The white paper will consider international/domestic (including Chinese and German) security test guidance and standards on Industrie 4.0/Intelligent Manufacturing, especially in the context of the hierarchical structure of security standards ISO/IEC 27000 series, IEC 62443 series and IECEE, IEC 15408 (Common Criteria). It will also describe the requirements and challenges of security testing.

Digitalisation of the industry has changed considerably the objectives of a security test, from testing mostly closed-network or isolated hardware and software components to products and services which in part run in cloud and are accessible by an increasing number of end users, even via mobile phones. Since the complexity of the system under test also increases the attack surface and variety of attack vectors, security testing practices need to be adapted accordingly.

In recent years, several emerging technologies have also substantially affected both the scope and implementation of security testing. Hence, this white paper will address the security testing of solutions from specific technical domains, such as the testing of cryptographic algorithm implementation, the security of Machine Learning (ML) and Artificial Intelligence (AI) algorithms, joint functional safety and security-related (IEC TR 63069) testing, security and privacy-related testing of big data and cloud computing, e.g. with regard to de-identification.Security testing, a practice which ensures the security of a product, system or service, addresses verification and validation activities at all lifecycle phases, such as threat modelling to identify attack vectors and verification of security design at the design phase; at the implementation level, security testing will address the secure software design and software source code and Hardware Description Language (HDL) level for specific programming languages, including tests on compliance with guidance on secure programming and tool-based fuzz testing and the source code level.

Although some tasks in security testing can be automated, the human factor is still decisive in finding vulnerabilities or weaknesses. Security tests such as penetration and fuzz testing, in particular, fall into this category and are addressed in this white paper in order to promote ever increasing advanced approaches to ad hoc security testing. Thus, the skills required of specialised test staff are addressed at a general level (ISO/IEC 27021) and with regard to the respective technical domains, e.g. for the testing of cryptographic algorithms.

Beyond the requirements to be met by security test staff, the white paper also addresses the requirements on laboratories performing security tests, evaluations and certifications. In particular, a methodological approach to implementing security tests is crucial for products and services subject to evaluations and certifications. Hence, this document also mentions applicable standards and methodologies (for example, OSSTMM).

As shown in Figure 1, this document is structured in line with the principal lifecycle phases of I4.0/IM products, platforms and systems. Accordingly, it first considers security testing during the development phase of IIoT components, products and IACS platforms (Section 5). Then it looks at security testing during systems engineering and

integration (Section 7) and for operators and service providers (Section 8). Security testing considerations along the supply chain, including requirements for test input data, are addressed in Section 9.

The subsequent sections relate to requirements on security test equipment and tools, including ML/AI-based approaches (Section 10), qualifications for security test staff (Section 11) and security test labs (Section 12).

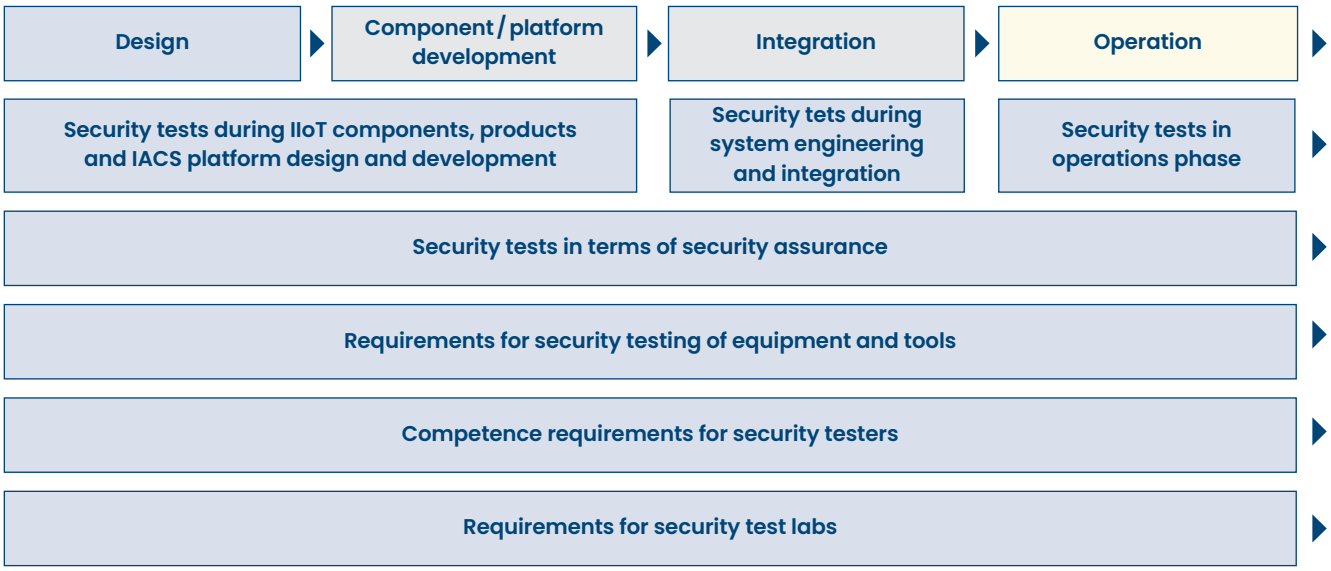


**Figure 1** —Coverage of this white paper along system life cycle

# 2. Scope

Security testing refers to the informal and formal methods used to verify the security posture or resilience of a system (hardware, software or a service) and organisation (people and policies) against security threats and more specifically against attack vectors. Security testing is normally applied at each lifecycle phase of a component, product, platform or system (in terms of technology, process and people), including the design and development phase as a part of the security-by-design principle. This is a critical approach, as relating security testing only to the production-ready or in-operation end systems can be an expensive and complex undertaking. As security testing relates to people and policies, it is done on a periodic basis to confirm the effectiveness of security training, procedures, etc. on identifying potential for continuous improvement. The methods applied to systems and organisations typically vary depending on the requirements defined internally or externally (national and international standards and guidelines) and the applicability of the method. For instance, social engineering and fuzzing are both security testing techniques; however, the former is designed for testing people in particular, while the latter targets technology such as the software elements (code, file extension, communication protocol, etc.). Additionally, there exist formally defined security testing standards specific to either OT or IT environments.

As stated above, this document locates security testing in the system life cycle, including the operations environment, to facilitate a securely running system. Therefore, the scope of this work also looks at aspects such as security testing in design, development, engineering and integration phases of systems composed of components, products and services, as well as security tests in the operations phase, including system operators and service providers, and in the supply chain.

As stated in the introduction, this white paper will introduce a selection of formal security testing methods, which include the IECEE approach for IEC 62443 (including the new TeleTrusT evaluation method for IEC 62443-4-2:2019), the Common Criteria (ISO/IEC 15408), the Open Source Security Testing Methodology Manual (OSSTMM) and several others.

Similarly, there is no limitation on sources for guidance, and these will include international and national standards as listed in the Annex (§15).

Additionally, the white paper will address the need for security testing, as well as specific guidance on certain technical topics, including Trusted Platform Modules, Virtual Trust Modules, Hardware Description Languages, Automatic Code Generation, Machine Learning and Artificial Intelligence-based security testing, with reference to the respective technology-related standards, in the event that no dedicated security testing standards are available.

Due to the broad scope of security controls and topics relevant to security testing, this white paper cannot comprise all topics relevant to security testing and can only go down to a limited level of detail for some selections. For example, testing for forensic readiness is addressed, but will not cover in detail the security testing requirements for all related standards, including ISO/IEC 27037, ISO/IEC 27041, ISO/IEC 27042, ISO/IEC 27043 and multipart ISO/IEC 27050.

# 3. How to use this document

Although it is recommended that this document is read in its entirety to get maximum benefit, some hints are provided below for most targeted readings and for different types of audience.

Security testers may find it useful to read sections entitled 'Security testing during design and development of IIoT components, products and IACS platforms', 'Security Tests in Operations Phase ' and 'Security Testing in terms of Security Assurance'. Such readers may also find the section entitled 'Competence Requirements for Security Testers' interesting for an overview of competencies required.

For system (hardware and software) engineers or developers, the sections entitled 'Security testing during design and development of IIoT components, products and IACS platforms' and Security Tests during Systems Engineering and Integration' may make for useful reading.

For audiences dealing mostly with the governance and management of security tests, the sections entitled 'Security Testing in terms of Security Assurance', 'Requirements for Security Testing of Equipment and Tools', 'Competence Requirements for Security Testers' and 'Requirements for Security Test Labs' may be beneficial.

# 4. Terms, definitions and abbreviations

**Note** (on security testing terminology):
**In some standards and guidance documents, alternative terms are used similar to 'security testing' or as overarching or related terms. For example, 'performing IT product security evaluations' is largely similar to 'performing security tests with the purpose of compliance evaluation'. The evaluations are typically performed by the means of security tests that demonstrate compliance with given claims, requirements or targets of evaluation. Accordingly, the term 'security evaluation' will be kept where the corresponding source document makes use of it in the title or content.**

## 4.1 Terms and definitions

**The definitions and descriptions of terms used in the different addressed standards are set out below. For further definitions, please refer to the first part of the respective standards series.**

| Term | Definition/description |
|---|---|
| **Black box testing** | Testing method that examines the functionality of a software application without delving into its internal structure. Applicable to security testing of software and hardware. |
| **Black hat hacker** | Person attempting to find computer security vulnerabilities and exploiting them for personal financial gain or other malicious reasons [1] |
| **Black hat hacker** | A team of security and domain specialists that protect and defend a digital system |
| **Common Vulnerability Scoring System** | Open industry standard for assessing the severity of computer system security vulnerabilities on a scale from 0 to 10 (most severe). Based on metrics that approximate the ease and impact of exploit. |
| **Ethical hacker** | Synonym for white hat hacker |
| **Fuzzing** | Fuzzing or fuzz testing is a black box software testing technique, which basically consists in finding implementation bugs using malformed/semi-malformed data injection in an automated fashion [2] |
| **Grey box testing** | A combination of white box and black box testing. The aim is to search for defects due to improper structure or usage of software applications. Parts of the internal structure of the system are known. |
| **Hyperjacking** | A cybersecurity attack in which a threat agent takes malicious control of the hypervisor that creates the virtual environment within a virtual machine host [3]. |
| **Paravirtualisation** | A technique that presents a software interface to virtual machines which is similar – but not identical to – the underlying hardware-software interface (supported e.g. by the Linux Kernel for increased efficiency and interface API consistency). |
| **Penetration testing** | An authorised, simulated cyberattack on a digital system, performed to evaluate its security. The purpose is to identify vulnerabilities and strengths as a basis for security risk assessment.<br>Also known as: 'pen test', 'pentest' and 'ethical hacking' |
| **Red team** | A team of white hat hackers and domain specialists who imitate real-world attacks on digital systems |
| **Red team – blue team exercise** | A planned simultaneous blue team and red team exercise for a representative facility of digital systems, typically with a one-week duration |
| **Security posture** | Overall status of cybersecurity readiness |
| **Security testing** | A process intended to reveal the security vulnerabilities of a digital system. Due to the limitations of security testing, passing the tests is not a 100% confirmation that no security vulnerabilities exist or that the digital system adequately satisfies all security requirements. |

| Term | Definition/description |
|---|---|
| **Testability** | The degree to which a model has sufficient information to support automatic test case generation [4] |
| **Test traceability matrix** | Documentation of the relation between items (in documentation or software) and tests.<br>[Based on ISO/IEC/EEEE 29119-1:2013 §4.90] |
| **White box testing** | Testing of the internal structures and workings of an application, as opposed to its functionality (i.e. black box testing). Internal knowledge of the system and programming skills are used to design test cases. Input data is chosen to exercise paths through the code and determine the expected output data.<br>Also known as 'transparent box testing' and 'structural testing'. |
| **White hat hacker** | A computer security expert who specialises in security testing methodologies, including pen tests. The purpose is to assess the security posture of a digital system or organisation.<br>A white hat hacker tries to improve security, e.g. by informing manufacturers about any vulnerabilities discovered. |
| **White box testing** | Testing of the internal structures and workings of an application, as opposed to its functionality (i.e. black box testing). Internal knowledge of the system and programming skills are used to design test cases. Input data is chosen to exercise paths through the code and determine the expected output data.<br>Also known as 'transparent box testing' and 'structural testing'. |

## 4.2 Abbreviations

| Abbreviations | Acronyms and meaning |
|---|---|
| **ABAC** | Attribute Based Access Control |
| **ACL** | Access Control List |
| **AI** | Artificial Intelligence |
| **AML** | Automation ML |
| **ANF** | Application Normative Framework |
| **API** | Application Programming Interface |
| **ASC** | Application Security Control |
| **ASIC** | Application-Specific Integrated Circuit |
| **BIOS** | Basic Input/Output System |
| **BDRA** | Big Data Reference Architecture |
| **CA** | Certificate Authority |
| **CC** | Common Criteria |
| **CFC** | Continuous Function Chart |
| **COTS** | Commercial-off-the-Shelf |
| **CPS** | Cyber Physical System |
| **CPLD** | Complex Programmable Logic Device |
| **CPPS** | Cyber Physical Production System |
| **CR** | Component Requirements |
| **CVSS** | Common Vulnerability Scoring System |
| **DAA** | Direct Anonymous Attestation |
| **DCS** | Distributed Control System |
| **DoS** | Denial of Service |
| **DDoS** | Distributed Denial of Service |
| **EAL** | Evaluation Assurance Level |
| **EK** | Endorsement Key |
| **EPID** | Enhanced Privacy ID |
| **FB** | Function Block |
| **FPGA** | Field Programmable Gate Array |
| **FR** | Foundational Requirements |
| **GDPR** | General Data Protection Regulation |
| **HDL** | Hardware Description Language |
| **I4.0** | Industrie 4.0 / Industry 4.0 |
| **IACS** | Industrial Automation and Control System |
| **IECEE** | IEC System of Conformity Assessment Schemes for Electrotechnical Equipment and Components |
| **IIoT** | Industrial Internet of Things |
| **IIP** | Industrial Internet Platform |

| Abbreviations | Acronyms and meaning |
|---|---|
| **IM** | Intelligent Manufacturing |
| **IMSA** | Intelligent Manufacturing System Architecture |
| **IT** | Information Technology |
| **JTAG** | Joint Test Action Group |
| **LNI 4.0** | Labs Network Industrie 4.0 |
| **MISRA** | Motor Industry Software Reliability Association |
| **ML** | Machine Learning |
| **ONF** | Organisation Normative Framework |
| **OPC UA** | OPC Unified Architecture |
| **OPST** | OSSTMM Professional Security Tester |
| **OSSTMM** | Open Source Security Testing Methodology Manual |
| **OT** | Operations Technology |
| **PDP** | Policy Decision Point |
| **PEP** | Policy Enforcement Point |
| **PICS** | Protocol Implementation Conformity Statements |
| **PIP** | Policy Information Point |
| **PKI** | Public Key Infrastructure |
| **PP** | Protection Profile |
| **QA** | Quality Assurance |
| **QoS** | Quality of Service |
| **RA** | Remote Attestation |
| **RACI** | Responsible Accountable Consulted Informed |
| **RAMI 4.0** | Reference Architectural Model Industrie 4.0 |
| **RBAC** | Role Based Access Control |
| **RBG** | Random Bit Generator |
| **ROT** | Root of Trust |
| **RTOS** | Real-Time Operating System |
| **SAR** | Security Assurance Requirements |
| **SCADA** | Supervisory Control and Data Acquisition |
| **SDL** | Security Development Lifecycle |
| **SFR** | Security Functional Requirements |
| **SIS** | Safety Instrumented System |
| **SL** | Security Level |
| **SoD** | Separation of Duties |
| **SVV** | Security Verification & Validation |
| **ST** | Security Target |
| **TCG** | Trusted Computing Group |
| **TMT** | Threat Modelling Tool |

| Abbreviations | Acronyms and meaning |
|---|---|
| **TOE** | Target of Evaluation |
| **TPM** | Trusted Platform Module |
| **VHDL** | Very High Speed Integrated Circuit Hardware Description |
| **VMM** | Virtual Machine Monitor |
| **V&V** | Verification & Validation |
| **vTM** | Virtual Trust Module |
| **XACML** | eXtensible Access Control Markup Language |

# 5. Types of security tests

As this white paper assumes that security is an integral part of a product or system development life cycle, such products and systems should necessarily have security requirements besides functional requirements. The security requirements will cover secure design, secure coding, secure configuration, as well as secure operations in terms of security policies, training courses, user guides and regular verification and validation of security measures.

## 5.1 Conformance testing

Conformance tests are structured security verification and validation tests, in which security measures are checked based on a list of controls and the result of any control can be selected from a set of predetermined options. Such tests may be performed either in an automated or manual manner. Below is a sample list of such conformance tests:

- security audit to verify security requirements of the product or system,
- security configuration reviews to verify hardening security measures,
- compliance with security standards such as IEC 62443 at assurance levels Security Level 1 and Level 2,
- compliance with ISO 27001 to verify organisational security measures,
- secure code review to verify secure code related requirements.

## 5.2 Penetration testing

A penetration test, also known as pen test, pentest or ethical hacking, is an authorised simulated cyberattack on a digital system, performed to evaluate its security. The purpose is to identify vulnerabilities and strengths, as a basis for a security risk assessment.

A penetration tester

- potentially finds security vulnerabilities,
- potentially gains read-access or write-access to data and/or
- potentially manipulates the system behaviour.

A penetration tester does not take credit from and does not rely upon any lifecycle tests, product evaluation tests or external audits or certifications. Accordingly, even if development documentation is available, no information from tests performed during development should be relied upon.

### 5.2.1 Historical evolution of pen testing

How did pen testing initially evolve? During the mid 1960s, time-sharing computers made resources accessible via communication lines. This raised security concerns. In late 1967, the US Department of Defense organised teams of 'penetrators'. Originally called 'tiger teams', they are now known as **Red Teams**. Sponsored by government or industry, the teams attempted to

- break the defences of computer systems
- uncover and eventually patch security weaknesses.

### 5.2.2 Key steps of a pen test

Currently the key penetration testing steps include (example):

0. **Reconnaissance (or just 'recon') phase**
   Gathering information about the target. This can later be used for more focused attacks, including subsequent activities to acquire further information, e.g. via targeted phishing emails.

1. **Identifying security vulnerabilities**
   (e.g. by brute force attacks, open ports, unpatched operating systems or applications,

man-in-the middle attacks, replay attacks, zero-day exploits).

2. **Identifying exploits for vulnerabilities**
   (e.g. maliciously writing to a memory region that has no impact on system behaviour is not yet an exploit). Often it takes longer and requires additional knowledge to develop an exploit for a new vulnerability.

3. **Designing a penetration proof of concept around a set of vulnerabilities**
   (e.g. get access to client computer, do privilege escalation, go to next server). This should be done after an iteration of 1 and 2.

4. **Testing the penetration attack**

5. **Establishing a connection or installing the attack payload**
   (e.g. despite an air gap) on a real target system.

6. **Initiating or triggering the attack**

7. **Exploiting the attack**
   for retrieval of classified information, system control takeover or attack extension – just for demonstration without incurring any damage, (e.g. via privilege escalation, password cracking).

Penetration testing may rely on security assessment tools. Often these tools run on a specialised operating system distribution, which:

- comes with penetration testing tools installed;
- has configurations ready for pen testing;
- e.g. setting of network interface cards into promiscuous mode
- is based on a special Linux distributions, e.g.
  - Kali Linux (successor of BackTrack) based on Debian Linux
  - BackBox based on Ubuntu Linux
  - Pentoo based on Gentoo Linux
- deploys specialised frameworks such as Metasploit.

Maintaining such specialised operating system distributions helps with the effective performance of penetration tests, also as part of product validations.

**Note:**
**Care must be taken with frameworks (such as Metasploit) that include some built-in anti-forensic and evasion tools. These are intentionally implemented and configured in such a way that the effect of the penetration tests (which may include changes in the target) cannot be tracked. Accordingly, after completion of these tests, the initial software must be restored.**

### 5.2.3 Planning pen tests against industrial systems

A pen test activity is considered as a project, since scoping, test environment preparation, scheduling of resources and test steps, including their execution, and reporting work must be managed. Although execution time for the pen test as a project is very short compared to conventional implementation projects, a pen test project brings the attacker's view into the security verification and validation activities; thus, expected outcome is crucial in terms of security quality assurance.

When planning a pen test, the following must be taken into consideration:

- the scope of the I4.0/IM systems to be considered (limited scope);
- a decision on the time slot for execution, e.g. during plant maintenance or outage;
- legal clearance with regard to potential consequences on damage or leaked data;
- preparation to reinstantiate the pen-tested system to its initial state after completion of all tests.

Although not part of the penetration test itself, it is essential to first ensure that the I4.0/IM systems are protected, e.g. by security hardening measures and secure configurations.

Pen tests are typically scheduled on a regular basis in order to assess the security posture of a factory or plant. Often the tests are performed on a yearly basis at different manufacturing sites or different facilities or systems at the same site. In some situations, dedicated pen tests are performed, e.g. before a company acquires a plant or factory from another company or before two companies

merge, in order to assess the security posture before the industrial networks are interconnected.

### 5.2.4 Pen testing skills

Penetration testers are required to have sufficient skills in the use of tools specific to the scope of the target under test. On the other hand, not every security tester capable of using tools can find vulnerabilities or weaknesses in the system under test. Penetration testers with the mindset of sensing and finding anomalies and trying to break the system using those anomalies can uncover security findings that would highly degrade the security posture of the system. In particular, uncovering security weaknesses that might result from application logic flaws, incorrectly designed application workflows or using weaknesses in different components to reach a high-severity weakness are examples of findings which are very difficult to identify by other security testing practices.

As stated in Section 11.8, a penetration tester needs additional knowledge and expertise in specific technology areas (i.e. TLS client authentication on end clients, PKI key generation and deployment using protocols such as CMP, specific custom implementation of protocols such as customised OPC UA etc.) in the event that the system is developed based on higher security assurance levels against security threats and attacks. In such cases, a penetration tester may require some preparation time before the pen test commences, to gain more expertise with the technology used.

# 6. Security testing during design and development of IIoT components, ucts and IACS platforms

The whole development life cycle of digital products must address principles of secure design, implementation, verification and validation as key elements in the overall consideration of cybersecurity along the supply chain. Accordingly, security testing must be addressed already during the generic (non project-specific) validation of products and platforms. Section 6.1 addresses security aspects during software development. Section 6.3 considers the testing of programming language-specific vulnerabilities. Section 6.4 addresses security tests during hardware development, with a focus on Hardware Description Languages (HDL), hardware configuration and Trusted Platform Modules (TPM).

The processes specified by this practice are used to ensure that product features are implemented securely.

## 6.1 Security tests in design phase

Security tests performed in the design phase of components, products and platforms are mostly aimed at determining security threats that may impact the target under consideration. The security threats may be selected from a lengthy list of available security threats documented in guidelines and standards; however, for the sake of completeness, it would be better to perform a threat modelling based on methodologies, as set out in guidelines or standards. For example, there are specific security threat modelling methodologies for software (see section 11.3, [5] or ISO/IEC 27034); however, such methodologies may be utilised with some customisation to systems composed of hardware and software.

In order to ensure the security of components, products and platforms, it is recommended to integrate security threat modelling (and possibly also risk assessment, if the value of the assets in operations can be estimated) with the component, product and platform life cycle, so that security will be a part of the process.
In the security threat modelling activity, in addition to the security testers, it would be beneficial to involve the product owner, architect, security architect and experts capable of handling the methodology and tools, in order to bring the attacker's point of view when identifying security threats.

Detailed component diagrams, data flow diagrams and interface diagrams can be used as an input for the threat modelling activity. These will help attendees to understand respectively the system components and subcomponents needed to consider supply chain-related threats, the way data may be manipulated to impair the security of the target, and the attack surfaces that exist.

## 6.2 Security tests during software development

The following sections will address integrated software security testing (6.2.1), security V&V testing (6.2.2), traceability of security tests to security requirements (6.2.3) and independence of security tests (6.2.4).

### 6.2.1 Integrated software security testing

ISO/IEC/IEEE 29119-2 supports dynamic testing, functional and non-functional testing, manual and automated testing, and scripted and unscripted testing. As indicated by the 'Security Testing' box in Figure 2, the selected security tests are considered as a part of the software tests.

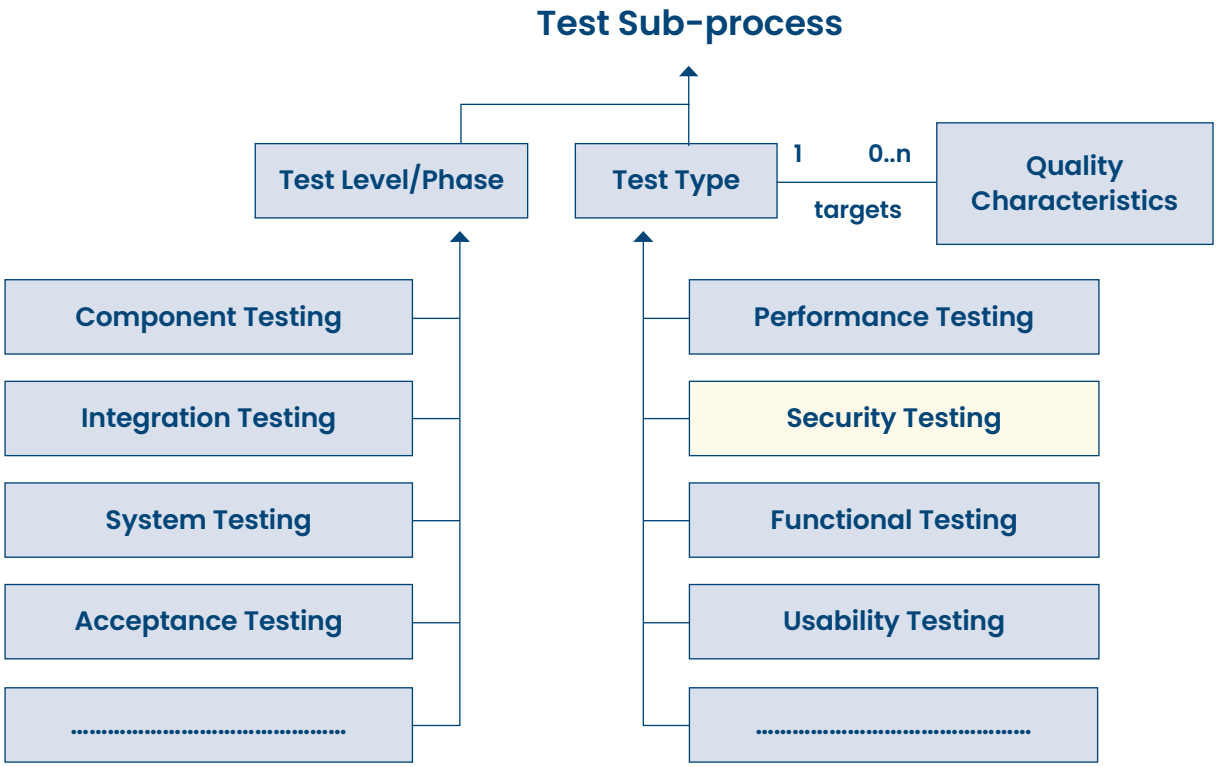


**Figure 2** — Security testing as part of software testing ISO/IEC 29119-1:2013 §5.2

In accordance with ISO/IEC 29119-1:2013 §4.22, testing is conducted to evaluate the degree to which a test item, including associated data and information, are protected to ensure that unauthorised persons or systems cannot use, read or modify them, and authorised persons or systems are not denied access to them.

**In the software development context, security testing shall demonstrate resilience against:**

- unauthorised access (read or modify);
- disclosure of confidential data;
- Denial of Service (DoS) or Distributed Denial of Service (DDoS) attacks.

**Security testing approaches in accordance with ISO/IEC 29119-3:2013 Annex M.2 include:**

- identification of security controls;
- agreement of security controls test coverage, e.g. between security tester and head of testing;
- use of tools;
- use of scripting.

At this stage, the security and software tester must assume that the right set of security controls has been selected.

### 6.2.2 Security verification and validation testing

The processes specified by this practice are used to document the security testing required to ensure that all security requirements have been met for the product and that the security of the product is maintained when used in its product security context and configured to employ its defence-in-depth strategy.

Security testing can be performed at various times and by various personnel during the SDL, based on the type of testing and development model used by the vendor. For example, fuzz testing could be performed during software development by the software development team and later in the cycle by a test team. As part of validation testing, 'normal pen tests' can be deployed. The word 'normal' simply indicates that this is a limited-effort security test in a representative

environment and not a targeted pen test for a specific I4.0/IM target in its final and often unique deployment environment.

### 6.2.3 Traceability of security tests

While tracing the links from security test cases to security requirements is part of verification (addressed in 6.2.3), execution of the security test cases can be seen as part of validation. In cases where validation activities are described as part of the security controls (e.g. in line with the concepts of ISO/IEC 27034-5 and ISO/IEC 27034-5-1), this is essentially implementation of the designed security validation steps.

As part of threat mitigation testing, the product- and platform-specific security defence-in-depth (DiD) concepts must be addressed by the security tests coverage. In particular, if two security controls are deployed for layered DiD, both must be individually tested, as well as with regard to common causes that may result in the failure of both security controls.

While software performance tests and functional tests are needed to validate the extent to which performance requirements and functionality requirements are met, security tests are required to demonstrate that graded security requirements are met. In the context of software development this is usually documented by a 'test traceability matrix' (ISO/IEC 29119-1:2013 4.90) also called 'verification cross reference matrix', 'requirements test matrix' or 'requirements verification table'.

Often the primary assets for security (ISO/IEC 27005:2018 B.1) relate to processes such as business continuity. Similarly, top-level requirements relate to business processes. However, security tests must cover the supporting assets (ISO/IEC 27005:2018 B.1), such as hardware, network resources and software. Accordingly, it is necessary to demonstrate that the security risk assessment and the traceability of security tests ultimately corroborate compliance with the precise and well-structured requirements (which were derived from the initial requirements and project needs).

In the context of I4.0/IM, the requirements and test cases can be linked to supporting assets, especially when these are semi-formally defined, e.g. by the use of Automation ML.

In cases where security controls are semi-formally described, e.g. according to the new ISO/IEC 27034 series, security testing must include the verification and validation criteria specified with each security control, according to the 'level of trust' (security level) of the security control.

### 6.2.4 Independence of software security testing according to ISO/IEC/IEE 29119-1

'Independence in Testing' (ISO/IEC 29119-1:2013 §E.3) ensures that in addition to the author(s) testing their product and platform, the level of independence between author and tester increases gradually:

- security tests are designed and executed **by a person other than the author of the software,** typically another author, reporting to the same manager;
- security tests are designed and executed **by a tester, reporting to the same manager;**
- security tests are designed and executed **by testers independent of the producing organisational** unit (still in-house);
- security tests are designed and executed **by testers employed by an external organisation** (consultants), but working in the same organisation as the author;
- security tests are designed and executed **by testers in an external organisation** (third party testing).

The appropriate level of independence must be in line with the potential impact of remaining security vulnerabilities.

**Note:**
**The developers of software and hardware products should not know what test will be performed. This will ensure that developers are not limiting their tests to already predefined test cases.**

| Abbreviations | Reference | Level of independence |
|---|---|---|
| **Security requirements testing** | SVV-1 – Security requirements testing | Independent department |
| **Threat mitigation testing** | SVV-2 – Threat mitigation testing | Independent department |
| **Abuse case testing** | SVV-3 – Vulnerability testing | Independent person |
| **Static code analysis** | SI-1 – Security implementation review | None |
| **Attack surface analysis** | SVV-3 – Vulnerability testing | Independent department |
| **Known vulnerability scanning** | SVV-3 – Vulnerability testing | Independent department |
| **Software composition analysis** | SVV-3 – Vulnerability testing | None |
| **Penetration testing** | SVV-4 – Penetration testing | Independent department or organisation |

**Table 1** — Required level of independence of testers from developers according to IEC 62443-4-1

### 6.2.5 Required level of independence of security testers according to IEC 62443-4-1

In accordance with IEC 62443-4-1, security-related processes should be assigned to personnel capable of the tasks. IEC 62443-4-1 requires security testers to be independent of the developers. For example, it specifies the required level of independence of testers from developers with regard to Security Verification & Validation (SVV), as shown in table 1.

**In the table, explanations of the independence levels are listed as follows:**

- **None** – no independence required. Developer can perform the testing.
- **Independent person** – the person who performs the testing cannot be one of the developers of the product.
- **Independent department** – the person who performs the testing cannot report to the same first line manager as any developers of the product. Alternatively, they could be a member of a quality assurance (QA) department.
- **Independent organisation** – the person who performs the testing cannot be part of the same organisation as any developers of the product. An organisation can be a separate legal entity, a division of a company or a department of a company that reports to a different executive, such as a vice president or similar level.

## 6.3 Testing programming laguage-specific security vulnerabilities

Program specification languages are primarily for software development. However, Hardware Definition Languages (HDL), as addressed in Section 6.4.1, share some commonalities.

Testability is a key concept for software and hardware specifications. Typically, informal specifications at a higher abstraction level provide excellent end-user readability at the cost of reduced precision. Incomplete semi-formal specifications, using e.g. UML notations, attempt to find a compromise between readability and accuracy, while making no claim to be complete. Functional Diagram specifications based on Function Blocks and interconnections, e.g. in compliance with PLCopen, provide domain-specific Function Blocks and interconnection types that allow high accuracy, but limited to specific application domains. In these specific situations – which are important for I4.0/IM, since they are frequently encountered in the industrial automation domain – automatic code generation from graphical specifications is feasible (also when generating into an HDL). This avoids many types of software errors and excludes some types of security vulnerabilities (e.g. when generated code does not allocate memory on the heap).

However, for general I4.0/IM applications, specific software programming languages or hardware definition languages (HDL) are also needed. In recent years, we have seen both the trend from less precise programming constructs to more restrictive programming languages and the transition towards less formal languages. A tendency towards more restrictive programming languages can be found in projects that are gradually shifting from the use of Javascript to TypeScript. In this case, TypeScript allows software programmers to better structure their code (with Interfaces, Classes, ...) and to optionally indicate the permitted types (not just by inference from initially assigned values). On the other hand, new programming languages, such as Kotlin, allow software programmers to optionally provide type specifications, thus allowing users of Scala or Java to develop their applications more efficiently (instead of being restricted to static typing). Similarly, in the industry domain, including for machine learning applications, Python is enjoying a continuously increasing user base while being essentially loosely typed as compared to C++, C# or Java.

Accordingly, security testing during development must take into consideration the capabilities and limitations of programming languages and respective artefacts, including development environments, tool-based optimisations and testability.

### 6.3.1 General description

ISO/IEC TR 24772 (including parts 1 to 3 from 2019/2020, see Annex 15.1) specifies software programming language vulnerabilities that should be avoided in the development of systems where assured behaviour is required for security, safety, mission-critical and business-critical software.

In general, this guidance is applicable to the software developed, reviewed or maintained for any application. ISO/IEC TR 24772 does not address software engineering and management issues.

ISO/IEC TR 24772 seeks to avoid the debate about where low-level design ends and implementation begins, by treating selected issues that some might consider design issues rather than coding issues.

The body of ISO/IEC TR 24772 provides users of programming languages with a language-independent overview of potential vulnerabilities in their usage.

### 6.3.2 Intended audience

The intended audience for ISO/IEC TR 24772 is those who are concerned with ensuring the predictable execution of their system's software; that is, those who are developing, qualifying or maintaining a software system and need to avoid language constructs that could cause the software to execute in a manner other than intended.

Developers of applications that have clear safety, security or mission-criticality are expected to be aware of the risks associated with their code and could use ISO/IEC TR 24772 to ensure that their development practices address the issues presented by the chosen programming languages.

A weakness in a non-critical application may provide the route by which an attacker gains control of a system or otherwise disrupts co-hosted applications that are critical. It is hoped that all developers would use ISO/IEC TR 24772 to ensure that common vulnerabilities are removed or at least minimised for all software applications.

**Specific audiences for this International Technical Report include developers, maintainers and regulators of:**

- **safety-critical** applications that might cause loss of life, human injury or damage to the environment;
- **security-critical** applications that must ensure properties of confidentiality, integrity and availability;
- **mission-critical** applications that must avoid loss or damage to property or finance;
- **business-critical** applications where correct operation is essential to the successful operation of the business;
- **scientific, modelling and simulation** applications which require a high degree of confidence in the results of possibly complex, expensive and extended calculation.

### 6.3.3 Main contents

ISO/IEC TR 24772 (including parts 1 to 3 from 2019/2020) gathers descriptions of programming language vulnerabilities, as well as selected application vulnerabilities which have occurred in the past and are likely to occur again. Each vulnerability and its possible mitigations are described in the body of the report in a language-independent manner, though illustrative examples may be language specific. In addition, it describes the vulnerabilities and their mitigations in a manner specific to the language.

ISO/IEC TR 24772-1 contains descriptions that are intended to be language-independent to the greatest possible extent and the generic guidance applied to particular programming languages.
The descriptions include suggestions for ways of avoiding the vulnerabilities. Some are simply the avoidance of particular coding constructs, but others may involve increased review or other verification and validation methods. Source code checking tools can be used to automatically enforce some coding rules and standards.

1. **Vulnerability Issues** provides rationale for ISO/IEC TR 24772 and explains how many of the vulnerabilities occur.

2. **Programming Language Vulnerabilities** provides language-independent descriptions of vulnerabilities in programming languages that can lead to application vulnerabilities. Each description provides:

   - a summary of the vulnerability,
   - characteristics of languages where the vulnerability may be found,
   - typical mechanisms of failure,
   - techniques that programmers can use to avoid the vulnerability,
   - ways that language designers can modify language specifications in future to help programmers mitigate the vulnerability.

3. **Application Vulnerabilities** provides descriptions of selected application vulnerabilities which have been found and exploited in many applications, have well-known mitigation techniques, and result from design decisions made by coders in the absence of suitable language library routines or other mechanisms. For these vulnerabilities, each description provides:

   - a summary of the vulnerability,
   - typical mechanisms of failure,
   - techniques that programmers can use to avoid the vulnerability.

4. **New Vulnerabilities** provides new vulnerabilities that have not yet had corresponding programming language text developed.

5. **Vulnerability Taxonomy** and List is a categorisation of the vulnerabilities of this report in the form of a hierarchical outline and a list of the vulnerabilities arranged in alphabetic order by their three-letter code.

6. **Language Specific Vulnerability** Template is a template for writing programming language-specific annexes that explain how the vulnerabilities of clause 6 are realised in that programming language (or show how they are absent), and how they might be mitigated in language-specific terms.

7. **The annexes,** each named for a particular programming language, list the vulnerabilities of Programming Language and Application Vulnerabilities. They describe how each vulnerability appears in the specific language and how it may be mitigated in that language, whenever possible. All language-dependent descriptions assume that the user adheres to the standard for the language as listed in the sub-clause of each annex.

ISO/IEC TR 24772-3:2020 (see Annex 15.1) provides specific guidance for ANSI C, which remains one of the key programming languages used for embedded systems, due to its suitability for firmware programming and for efficient use of hardware resources.

The MISRA C (Motor Industry Software Reliability Association) guidance specifically aims to restrict language constructs for safety-related

applications and is continuously updated, see 'MISRA Compliance:2020 Achieving Compliance with MISRA Coding Guidelines' [6]. Testing that the MISRA C guidance is followed ensures that related potential vulnerabilities are already systematically excluded at the software source code level.

ANSI C is still the main language for implementation of the newest Linux kernel versions, with more than 27.8 million lines of code in January 2020 (26.1 million LOC in January 2019) [7]. Accordingly, testing with regard to security-related vulnerabilities in the kernel source code is expected to be performed by the open source development community.

## 6.4 Security tests during hardware development

When it comes to hardware testing, 'normal pen tests' for hardware must first be performed. This includes checks for debug interfaces, fault injection attacks and side channel attacks. In-depth hardware security tests will depend on the hardware technology used.

The following sections address the 'software defined hardware', e.g. hardware defined by Hardware Definition Languages (HDL).

IIoT devices include custom hardware which is often implemented via FPGAs, CPLDs, ASICs or a mix, e.g. of microprocessors and FPGAs. Field Programmable Gate Arrays (FPGAs) allow for hardware to be defined via software, e.g. VHDL or Verilog. FPGAs, as used in satellites, for example, can be very complex. One key technical advantage consists in the efficient custom-built implementation of very fast algorithms, since input/output processing is directly implemented at hardware level. From a security perspective, FPGA-based implementations are less vulnerable to security attacks compared with classical software implementations. For this reason, it is necessary to adjust the corresponding security tests. However, the tools used to define and simulate FPGA-based hardware modules are very complex, typically in the range of 0.5 Gbytes executable sizes. Accordingly, most of the software and security testing is shifted from testing the Target of Evaluation (TOE), the hardware module itself, to the tools that are used to define the hardware. Therefore, when it comes to security testing of the tools, the approaches for software development and integration apply.

At the level of hardware modules, there remain some specific issues, e.g. on re-loading images (bitstreams) of FPGAs (except for FPGA types used in aerospace, which do not support re-initialisation).

Mixed hardware components that comprise both microprocessors and FPGAs are used with increasing frequency for IIoT products. For example, in late 2015, Intel acquired the FPGA and CPLD manufacturer Altera (for $16.7 billion). These mixed components (e.g. Intel Agilex SoC which integrates quad-core ARM and FPGA) allow for fast I/O and efficient implementation of time-critical algorithms, in combination with multicore processors that ensure execution of the automation software. Security testing for such mixed components must consider both the FPGA and software aspects.

### 6.4.1 Hardware Description Language (HDL)-specific security tests

This section addresses the security testing of generic HDL, implementation-specific HDL (VHDL, Verilog) and generated HDL.

#### General HDL-level security tests

In general, Hardware Description Languages (HDL) [8] allow for the definition of complex data structures (as structures of other data structures, similarly to procedural programming languages), with some limitations, e.g. regarding recursive definitions and pointers. The elementary types can be bits, bytes, fixed-point floating numbers of varying precision etc. HDL procedures can be defined and composed to process HDL structures.

#### VHDL and Veriolog-specific security testing

VHDL (Very High Speed Integrated Circuit Hardware Description) and Verilog allow for electronic design automation by supporting the definition and simulation of integrated circuits (ICs) as defined by input ports, output ports and internal memories.
The 2008 standard version of VHDL supports very powerful hardware descriptions by introducing the concept of software templates, similar to the

template concept of the C++ programming language. This allows for even more configurable hardware definitions.

Security tests at VHDL source code level must consider the complexity of the language and the potential vulnerabilities associated. In addition, it should be borne in mind that many vendors of HDL-related tools implement only a subset of the VHDL or Verilog standard (e.g. 25% [9]). Accordingly, security testing should also take into account whether less common constructs were used. These would be more prone to errors during implementation and in the respective simulation tools. In the worst case, this would result in a hidden vulnerability that is not – or not easily – detected using available VHDL simulation tools and hardware test tools. With regard to testability, therefore, special care is needed when deploying language features that are not commonly used and not well supported. In such cases, it may even be worth updating the Coding Style Guide used by developers, in order to avoid corner cases.

#### Security testing of HDL generated from procedural languages

While hardware description languages provide benefits in efficiency and the potential of miniaturisation for many applications, especially in IoT, electronic design automation skills are often not part of the formal training of software developers. As a compromise, sophisticated tools are used to convert programming language data structures and functions into HDL descriptions. While this goes along with the limitations in the freedom of declaring data structures and function prototypes (e.g. only a subset of the current ANSI C programming language 'C18'), it still introduces an additional level of complexity. This must be specifically addressed during security testing.

### 6.4.2 Hardware configuration specific security tests

In many cases, hardware components are defined via 'pure' HDL. However, in IIoT systems and in new embedded systems, a common approach is to use one or more predefined processor cores. These can be ARM cores or other HDL vendor specific cores (e.g. by Xilinx, Altera/Intel, Actel/Microsemi). Some of these cores can be HDL-level descriptions of legacy microprocessors, e.g. Motorola 6800, which are even available at no cost (e.g. at opencores.org). However, comparison tests show that processor instructions are not implemented (and tested) to the same degree as in the initial legacy products. This must be taken into consideration for security tests.

### 6.4.3 Hardware level hypervisor-specific security tests

A hypervisor or Virtual Machine Monitor (VMM) runs virtual machines directly on top of the hardware. A hypervisor controls the hardware and manages all guest operating systems (virtual machines). Hypervisors are also called 'type 1', 'bare metal' or 'native' hypervisors.Around 2005, several factorsled to a resurgence of virtualisation technologies. These included available hardware capabilities allowing virtual machines to run simultaneously, the need to run large multiprocessor installations, and improved security, reliability, and device independence.

In contrast to IT enterprise systems, specific robustness, security and real-time capabilities are required for embedded hypervisors that are targeting embedded systems and real-time operating system (RTOS) environments. As manufacturers of embedded systems usually have the source code to their operating systems, they can make use of the performance advantages of paravirtualisation. Paravirtualisation requires the embedded guest operating system to be explicitly ported for the para-API. Applications that are accessible through the paravirtual machine interface environment ensure run-mode compatibility e,g. across multiple encryption algorithm models. The 'paravirt-ops' (pv-ops) source code is part of the Linux kernel and provides a hypervisor-agnostic interface between the hypervisor and guest kernels.

The use of hypervisor technology by malware and rootkits installing themselves as a hypervisor can make the malware more difficult to detect. This type of cybersecurity attack, known as hyperjacking, could intercept any operations of the

operating system (e.g. while a password is typed in) without any anti-malware software detecting it, as the malware runs below the operating system.

The hypervisor represents a single point of failure with regard to the security and protection of sensitive information. As part of the hyperjacking, a threat agent must inject/replace the original hypervisor with his rogue hypervisor, run the rogue hypervisor on top of the original hypervisor or get control of the original hypervisor.

The following should be checked as part of security testing against hyperjacking [3]:

- security management of the hypervisor should be kept separate from regular network traffic;
- guest operating systems should not have access to the hypervisor;
- management tools should not be installed or used from any guest operating system;
- the hypervisor should be regularly patched.

### 6.4.4 Testing of TPM and vTM

#### Testing of dedicated single-chip and host processor-supported TPM solutions

A Trusted Platform Module (TPM) is a system component that has state which is separate from the host system to which it reports. The interaction between a TPM and the host system is through interfaces defined in ISO/IEC 11889.

A TPM may be constructed using physical resources that are permanently and exclusively dedicated to the TPM. Alternatively, a TPM may use physical resources that are temporarily assigned to the TPM. A TPM's physical resources may be located within the same physical boundaries (single-chip component) or within different physical boundaries.

The TPM component has a processor, RAM, ROM, and Flash memory. The host system cannot directly read from or write to the TPM memory. The only interaction with the TPM is through its I/O buffers.

While the behaviour of a TPM is defined in detail by C language reference implementation, including detailed comments in the ISO/IEC 11889 parts (more than 1,000 pages), actual implementation can be in another programming language. However, the external observable behaviour must be exactly the same as for the reference implementation. The TPM design makes use of different Root of Trust (ROT) concepts (Root of Trust for Measurement, Root of Trust for Reporting and Root of Trust for Storage). Manufacturers are encouraged to provide assurances that the root has been implemented in a way that renders it trustworthy. A certificate may identify the manufacturer and the Evaluated Assurance Level (EAL) achieved.

Another implementation of a TPM supported by ISO/IEC 11889 is to have the code run on the host processor while the processor is in a special execution mode. For these TPMs, parts of system memory are partitioned by hardware so that the memory used by the TPM is not accessible by the host processor unless it is in this special mode.
The correct handling of primary seed authorisations, field upgrade mode, logout control etc. is typically tested and certified according to Common Criteria. The TPM concept itself makes use of infrastructure that is also tested and certified according to Evaluation Assurance Levels, e.g. for chains of trust.

There are several different schemes for achieving mode switching, including System Management Mode, Trust Zone™ and processor virtualisation.

The special importance of security tests of TPMs during development is due to their inherent mode of operation. By definition, TPMs have to maintain internal data (private keys) without providing functions that can access the memory regions. Accordingly, the typical test coverage criteria cannot be applied once a TPM is shipped. The TPM receives data, performs operations on it (e.g. by using internal keys) but does not provide further testability support.

For TPM 2.0 vendors, [10] recommends implementing a Software TPM as a software emulator. While a software TPM is open to many vulnerabilities, including tampering and bugs in the operating

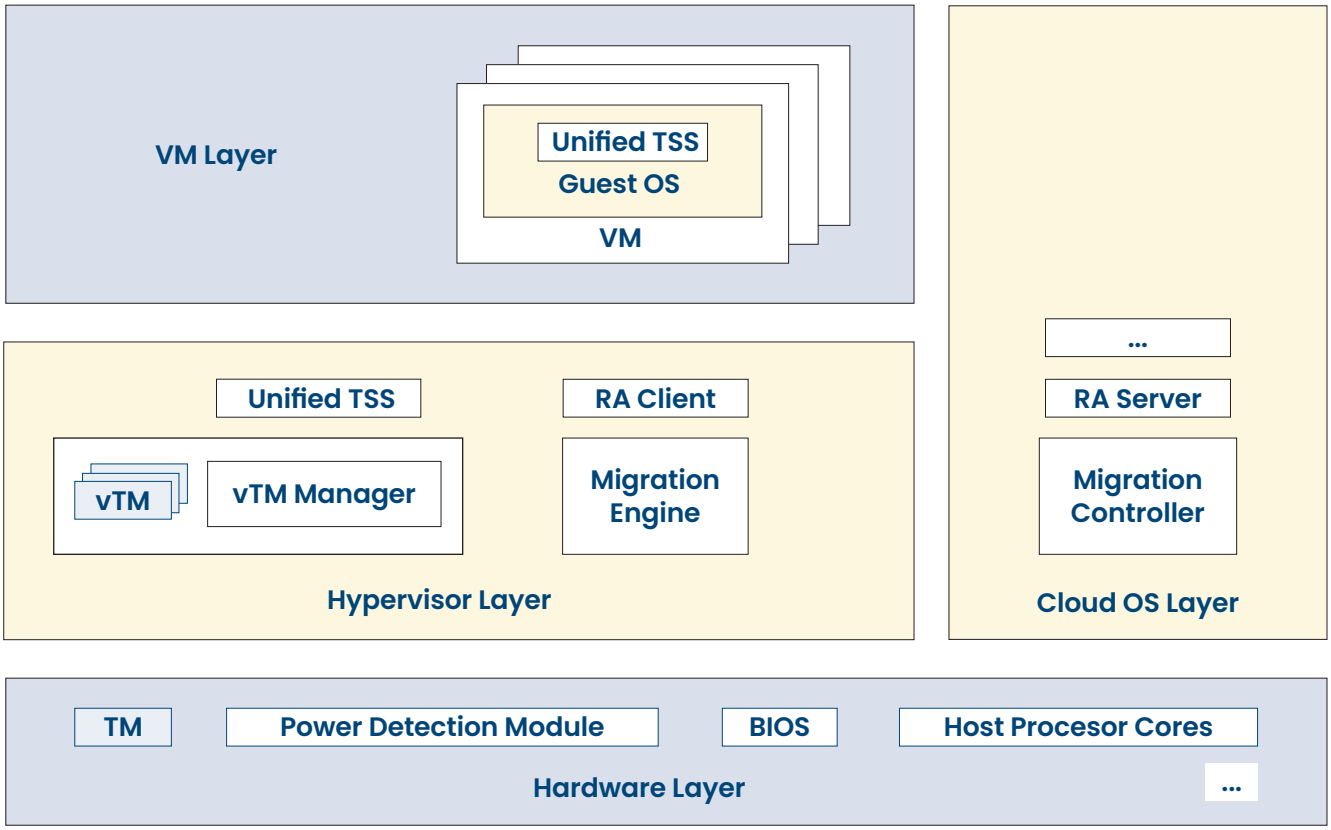


**Figure 3** — Virtualised Roots of Trust Concept

system running it, it is very good for testing and building a system prototype with a TPM in it. According to [10], a software TPM could provide the right solution/approach for testing purposes.

An example of tools for testing virtualisation solutions is the VASTO toolkit [11]. VASTO (Virtualisation ASsessment Toolkit) is a collection of Metasploit modules for use as a testing tool to perform penetration tests or security audits of virtualisation solutions, e.g. including VMware. For example, in a past version, VMware clients with (Port 80 and 443) access to the hosts were able to directly access the hypervisor through a browser and even use the host to download the vSphere management client (for convenience). However, an attacker with the vSphere client is able to directly manage the hypervisor once a username and password have been provided. Therefore, the above tool would support testing of the hypervisor in such scenarios.

### TPM key attestation

TPM key attestation prevents the export of a private key to an unauthorised device and can bind a user's private key identity to a device. TPM key attestation provides a hardware-rooted user identity, instead of the default software-rooted user identity. TPM key attestation leverages a TPM's Endorsement Key (EK) that is injected into the TPM when it is manufactured and that is unique to each TPM.

Trust in the Endorsement Key (EK) is based on the secure and tamper-proof storage of the EK in the TPM. The EK's certificate chains to the TPM manufacturer's issuing Certification Authority (CA).

Testing the chain means that the Endorsement Key's certificate can be cryptographically verified using the certificate of the TPM manufacturer's issuing Certificate Authority.

### Testing of virtual trust module solutions

ISO/IEC 27070 (Draft 2021) provides a virtualisation concept of the TPM architecture defined by ISO/IEC 11889-1. As indicated in Figure 3, the hardware layer, which contains the Power Detection Module,

Trust Module and BIOS, provides the basis for a Hypervisor, also called Virtual Machine Monitor (VMM). The Hypervisor Layer provides virtual trust module (vTM) instances. The association of vTM instances to hardware level TMs is managed by virtual trust module manager (vTM Manager). This concept involves two parts of a unified Trusted Software Stack, one for the hypervisor layer and one for the virtual machines layer.

Additionally, a migration engine (at the hypervisor layer) and a migration controller (at the cloud operating system layer) are needed for the migration of VMs.

Testing of the virtualised roots of trust approach requires the consideration of a broader context, as compared to a TPM, including:

- the virtualised trusted modules (vTM),
- the vTM Manager which links the vTM modules to VMs,
- the Unified Trusted Software Stack (TSS) at the hypervisor and VM layer,
- the Unified TSS used by the guest operating system of a VM,
- the Remote Attestation Client (RA Client) at the hypervisor layer,
- the Remote Attestation Server (RA Server) at the cloud operating system layer,
- the Migration Engine at the hypervisor layer and the Migration Controller at the cloud operating system layer.

These security tests can make use of results from Common Criteria-based evaluations, but they have to go further with regard to the hypervisor, host operating system (OS), guest OS, cloud OS and functionalities relating to moving virtual machines between hardware entities in the cloud. A general overview of topics to be considered with regard to implementation can be taken from the 'Security guidelines for design and implementation of virtualised servers' standard [ISO/IEC 21878]. These topics must be considered with regard to security testing, while applying the Guest OS, Hypervisor, Host OS and VM concepts together with the vTM, vTM Manager, Unified TSS etc. extensions.

These security tests may build on top of other approaches for testing virtualisation security. As an example, in [12], testing includes analysis of known vulnerabilities and fuzzing, in order to test the virtual device drivers on the virtualisation platforms VirtualBox, Hyper-V and VMware ESXI.

**Note:**
**Only 'bare metal' hypervisors, which are operating directly on top of the hardware (and not based on an operating system) are considered here.**

### 6.4.5 Testing of hardware related privacy aspects

#### Direct Anonymous Attestation (DAA)

In some cases, especially when relating to a large number of IoT devices, it is preferable to maintain the privacy of a Trusted Platform Module (TPM). In order to test whether this is correctly implemented, consideration must be given to Direct Anonymous Attestation (DAA) in accordance with ISO/IEC 20008-2:2013. ISO/IEC 20008-2 addresses mechanisms using a group public key for the implementation of anonymous digital signatures. Parts that should be tested include the process for generating group member signature keys and group public keys, the processes for producing and verifying signatures and the process for revoking group members.

The Direct Anonymous Attestation (DAA) cryptographic primitive allows authentication of a trusted computer, while preserving the privacy of the platform's user. Unlike traditional digital signature algorithms, in which each entity has a unique public verification key and a unique private signature key, DAA provides a common group public verification key associated with many unique private signature keys.

#### Enhanced Privacy ID (EPID)

Similar to the DAA implemented at the Trusted Platform Modules (TPM) level, the Trusted Computing Group (TCG) supports Enhanced Privacy ID (EPID), which is implemented at the microprocessor or chipset level.

DAA and EPID are created so that a device can prove to an external party what kind of device it is (and optionally what software it is running) without disclosing the device identity (proof of group membership without revealing the member). Enhanced Privacy ID (EPID) implements ISO/IEC 80009-4:2017.

When testing these privacy aspects, it should be considered that EPID keys are placed in devices during manufacturing and subsequently used for provisioning other keys for other services in a device.

### 6.4.6 Security testing of random bit generators

Numerous cryptographic applications require the use of random bits. These cryptographic applications include the following:

- random keys and initialisation values for encryption;
- random private keys for digital signature algorithms;
- random values to be used in entity authentication mechanisms;
- random values to be used in key establishment protocols;
- random PIN and password generation.

ISO/IEC 18031 distinguishes non-deterministic and deterministic random bit generators (also called pseudo random bit generators), while recognising the use of hybrid random bit generators. Typically, non-deterministic and hybrid random bit generators are based on a conceptual model with an entropy source (or multiple entropy sources integrated by a suitable signal noise generating design), a stochastic model and a cryptographic post-processing.

A Random Bit Generator (RBG) may be implemented in different hardware technologies, including in FPGA technology. The security test should first verify the conceptual model with regard to potential systematic faults, e.g. non-biased entropy source, design of the noise source, derived from a well-documented publicly available design, suitable pseudo random algorithm etc.

Then the security test should address implementation by using adequate test instrumentation and a validation of the stochastic model. Finally, the cryptographic post-processing should be validated.

**Note:**
**The terms Random Number Generator (RNG) and random binary generator are used in a way similar to the term Random Bit Generator (RBG), while an implementation potentially must meet further boundary conditions and may use a different cryptographic post-processing function.**

# 7. Security tests during systems engineering and integration

Section 7.1 considers security testing aspects when applying automatic code generation, as promoted by PLCopen, for example. Section 7.2 addresses security testing of applied security controls, e.g. security controls applied as part of security hardening. Section 7.3 addresses testing of the security impact on functional safety, e.g. to ensure that a security control will not adversely impact a safety function.

## 7.1 Automatic code generation-related security testing

### 7.1.1 Graphical specifications and code generation

In order to reduce the amount of manual coding and to increase the ease of reuse and configuration of existing source code functionality that is provided by well-tested functions, the concept of Function Blocks (FBs) and Functional Diagrams (or Function Block Diagrams) is used in the automation domain, e.g. based on IEC 61131-3, IEC 60880 and IEC 62138. A Continuous Function Chart (CFC) implementation language is a graphical programming language that makes use of FBs to graphically program large, complex function block diagrams. The resulting function block diagrams can be read like circuit diagrams or block diagrams from the Electrical Power Systems (EPS) domain. Accordingly, software development is more an engineering activity done graphically with a CFC editor, instead of coding in a programming language for embedded systems.

The CFC specification is typically stored in a project database that contains the function blocks and engineered functional diagrams. The source code is generated with a code generator out of the project database. The generated source code is then compiled and linked with the function block libraries that correspond to the Function Blocks used in the graphical specification.

In early versions of some standards, e.g. in IEC 60880:1986 (which already contained two sections on IT security) the Continuous Function Chart (CFC) approach and the accompanying code generation were not addressed explicitly. Accordingly, customers or regulators could require code reviews and security tests of the complete code, including the automatically generated source code. Later on, e.g. with 60880:2006, this was explicitly considered. Accordingly, security testing shifts from the testing of generated source code to evaluation of code generators and the process of code generation and linking of the final embedded software.

### 7.1.2 Test of secure code generation processes and environment

Automatic code generation comes with a reduced need for security testing. This is due to the fact that key functionality is included in the Function Block Modules (the manually coded functions corresponding to graphical Function Blocks and provided by well-tested Function Block Libraries). Furthermore, the graphically defined data flow is limited according to the interrelations (signals) between the graphical Function Blocks. Especially when used for applications relevant to functional safety, the data associated with each Function Block instance and each data interconnection (signal) can be allocated statically. This 'security by design' step excludes potential security vulnerabilities that are related to dynamic memory management, such as memory leaks and certain types of buffer overflows (when statically pre-allocated).

Nevertheless, with regard to security testing, the integrity of the code generation process must be ensured. This includes verification of the process as a whole, from the use of a CFC editor and database handling to linking of the final embedded loadable software image.

### 7.1.3 Test of output of code generators

Testing of the code generators ensures that generated source code will be correct (exactly implement the specification). However, with regard to complexity or potential manipulations, a 'health check' of the generated source may still be considered a resilience measure. This check could be done by alternative means, e.g. by selective source code inspection, by simulation runs of the compiled source code or by its use in a simulation environment (which may require a different compiler and linker as compared with the embedded system) or by the use of source code quality inspection tools, such as a linter (a static code analysis tool), which flags programming errors, coding style breaches or vulnerable software constructs.

**Note:**
**IEC 61131-3 'Programmable controllers – Part 3: Programming languages' is a key part of the PLCopen approach that is also followed by Industrie 4.0. As an independent organisation, PLCopen provides efficiency in industrial automation by supporting the graphical engineering of specifications and partially automatic implementations in order to reduce cost in industrial engineering.**

### 7.1.4 Reverse engineering of generated code

The ability to parse generated source code, knowing its structure (syntax and semantics), gives rise to a specific security test opportunity with automatic code generation. Accordingly, part of the initial graphical specification from the project database can be reverse engineered, thus allowing identification of potential manipulations (or unintentional errors) during the code generation process.

**Note:**
**Completely generating the initial graphical CFC specification from the source code is not feasible, as the graphical specification contains further details (e.g. the exact position of a Function Block on a Functional Diagram page) that are not needed and not reflected in the source code. Accordingly, in order to allow complete verification, further annotations (e.g. comments at specific locations in the source code) are needed.**

## 7.2 Testing of applied security controls

Application security controls are typically structured according to a domain-specific standard. The selection and configuration of security controls must consider the security grading. Accordingly, testing of the security controls must demonstrate that the stringency of the security level (security degree) specific controls is enforced. In several industry domains there are 4 security levels or security degrees, e.g. for ISO/SAE 21434:2021 (cybersecurity engineering for road vehicles), IEC 63096 (nuclear domain), IEC 62443 (horizontal automation). Once the correct security level has been identified, tests must be in line with the stipulations, including reliance on tests that are claimed to have been met by product or component suppliers.

**In principle, the tests will contain at least the following approaches:**

- completeness coverage,
- effectiveness of individual measures,
- effectiveness of combined measures.

Completeness coverage is similar to a quality assurance step (verification part of V&V). For example, if some settings are expected to be enabled in a configuration file, the test will check (e.g. manually or via a script) whether the settings (in a text file, registry or database) are indeed as requested by the security control.

The effectiveness check of individual measures does not credit the completeness check. For example, if the security control is supposed to disable some network ports and operating system services (e.g. provided via TCP and/or UDP), then a test program must check whether these ports and operating system services are indeed no long reachable.

Some security controls may consist of several elementary security controls, as can be represented e.g. with ISO/IEC 27034-5-1. In these cases, the

overall correct administration and configuration must be tested in order to demonstrate the effectiveness of combined measures.

The representation of security control specific tests can be semi-formally specified in line with ISO/IEC 27034-5. Each Application Security Control (ASC) is represented by the security level, the protective measure and additionally at least one security test measure (validation). As ASCs can be structured, this applies also to elementary and tree-like composed ASCs. The benefit of this approach becomes evident in a scenario where ASCs are developed by an independent supplier company or are delivered as part of a product. In these cases, the original designers of the ASC will also indicate the security validation steps.

The ASC approach is best leveraged by using an Organisation Normative Framework (ONF) and multiple Application Normative Frameworks (ANFs). The ONF comprises the security controls, including indication of the security level and corresponding security validation recommendations for Application Security Controls (ASCs) that can be shared by multiple application supported by an organisation. The Application Normative Framework (ANF) comprises security controls that are specific for a given application (e.g. software application or automation equipment). Versioning of ASCs allows the gradual improvement of security controls and corresponding tests, including linking to potential vulnerabilities (which may give rise to the need for a new version of a security control).

The Application Security Control (ASC) approach of ISO/IEC 27034 also assumes that an RACI (Responsible, Accountable, Consulted, Informed) scheme is in place. Among other things, the RACI indication of roles allows for streamlined communication of security test results, as deficiencies may point to security vulnerabilities.

Security controls relating to the interoperability of systems require special attention, as the involved systems may originate from different suppliers or may comprise systems from different product or platform generations. Security tests for error handling relating to interoperability should be addressed with the same emphasis as for performance tests.
Tests using security controls may implicitly or (preferably) explicitly assume representative threat models. A threat model may depend on the specific use of a device. For example, if some vulnerable protocol is excluded, the risks (and test cases) change.

As part of a 'brown field' approach, components and products should be in place so that the effectiveness of security controls for the different security requirements can be tested in an integrated environment.

## 7.3 Testing of security impact on functional safety

IEC TR 63069, IEC 62859 and the Sino-German White Paper on Functional Safety for Industrie 4.0 and Intelligent Manufacturing (2020) provide requirements and recommendations on how to jointly consider functional safety and cybersecurity. This typically relates to the most stringent security levels. Such requirements include e.g. that the failure of a security control should not adversely impact the safety function which the security control is supposed to protect. A related recommendation is to implement the security control in an isolated environment. Thus, for example, the security function will not use the same real-time resources as the safety function. Exhausting the main memory, disk space or processing responsivity are thus avoided by design. Nevertheless, security tests should demonstrate that the corresponding claims are indeed met.

For tests which look at functional safety and cybersecurity together, special consideration should be given to the validation of security controls that are in place for conduits between different security zones. Often, equipment that processes higher safety functions is assigned to rather isolated security zones. Accordingly, the test scenarios should assume attackers trying to penetrate from a lower security zone to a security zone that contains equipment which executes functional safety-related processes.

These tests must cover the firewalls and, in some cases, physically unidirectional security gateways (data diodes) between zones.

In addition to testing protective security controls, it is also necessary to test detective security controls. In an industrial automation environment, typically a given set of industrial communication protocols are deployed. Detective security controls will monitor that no other types of communication are going on and potentially perform deep packet inspection on the content of the datagrams. The corresponding security tests must demonstrate the effectiveness of detective controls and that there is no retroaction (with a possible impact on functional safety) by the implemented controls.

# 8. Security tests in operations phase

## 8.1 Security testing in operational environments

Since the industrial systems host divergent sets of products (hardware, software) and services in layered and segmented networks, and since such systems partly provide critical services, availability of such systems is more important than their confidentiality and integrity security dimensions. Therefore, such systems are also designed for less downtime, which has a direct effect on security testing in operations.

Both planning (see Section 5.2.3 for details) and execution of security tests in operational environments cost more effort and require more caution than security tests in the design and implementation phase. As an implicit consequence of this, security testing engagements require more coordination with the customer and product team operating the system. Due to availability constraints, the duration and time of security testing activity is restricted by maintenance windows, which puts extra pressure on security testers and the security testing project.

A security testing activity for larger systems, complete manufacturing or a critical infrastructure environment in operations mode may be planned as one large project with many subprojects. In such a case, the large project may be a coordination project that includes planning, scoping and scheduling of subprojects and relates the outputs of the subprojects.

One advantage of a security testing activity in an operational environment is the availability of data and data flows in order to better understand system behaviour, generate realistic attack vectors and determine payloads to verify the exploitability of vulnerabilities.

### 8.1.1 Security testing on digital twins

One recent technological development that helps to perform security testing activities which are not constrained by time and additional caution is the digital twin. Digital twins in operational mode are coupled either tightly or loosely with the twinned component or system under consideration. Tight coupling means the response of the digital twin obeys time constraints, whereas a loosely coupled twin is less time constrained or not time constrained at all. Both couplings are stateful and must be considered during test performances.

There are already products on the market which provide digital twins for industrial components of a system, but also digital twins for the system as a whole. On the other hand, some of these digital twins, especially those which are twinning the whole system, do not refer completely to the system's entire data, but only to some abstractions based on abstract features of critical component functions.

At the component level, where the digital twin of a component is loosely coupled and refers to state data of the respective component behaving in an operational environment, security testing can be performed on the digital twin in a time-unconstrained manner. This specific situation would allow security testing to be carried out in a project without additional time pressure, which offers a chance to achieve completeness conditions of security and safety constraints.

However, where a digital twin is stateful and tightly coupled with a system component or with the whole system, the behaviour of these latter is to be analysed or predicted by carrying out a simulation by means of the tightly coupled digital twin. The coupling is a control loop, which means that there are sensors which generate data representing the current state of the twinned component or system. Since the digital twin comprises a model

of the behaviour of the component or system, the digital twin can evaluate the measured state conditions against the model state with respect to correctness of safety and security constraints or with respect to precision or accuracy. The result of the digital twin's evaluation is fed back to the input of the component or system considered and may lead to adjustment of the controls of the input parameters (performed by an actuator).

Consequently, a tightly coupled digital twin to be tested must consider time as an independent variable that constrains the state and output of the system observed with respect to the controls of the digital twin. For example, testing the precision or performance accuracy of the digital twin is performed by validating its capability to react on possible critical system state changes or its capability to identify errors or deviations. A digital twin must provide these capabilities so that real-world system imperfections can be evaluated. The digital twin's performance characteristics to be evaluated are based on a well-known reference signal inputted to the component or system that is called steady-state error testing.

Testing of a digital twin must take notice of the inherent capabilities of a digital twin coupled with a real-world system or components of it. This comprises:

1. making decisions by means of an accurate and updatable model of system behaviour, including the dynamics of the system's objects or subjects;

2. evaluating the response time and the digital twin's performance of data generated by sensors and received by actuators of the system;

3. measuring the performance of the digital twin, taking into account signals for the system's environment;

4. evaluating the efficiency and accuracy of the automation and autonomy algorithms.

Furthermore, security testing is based on certain types of meta-data, such as the value of the system from the point of view of an attacker, as well as the behaviour of internal users of the system or results of phishing attacks etc. capturing any information that helps to identify vulnerabilities of the system. On the one hand, the digital twin may execute so-called breach and attack simulations on the coupled system to check the risks of possible cyberattacks on components or the whole system. On the other hand, the coupled system must be tested by a 'red team' against cyberattacks under real-time operating conditions of the digital twin and the interacting system under consideration.

## 8.2 Testing for operators

The following sections address several access control methods, including role-based access control, list-based access control, attribute-based and multi-factor authentication-based access control. For I4.0/IM in particular, Attribute-Based Access Control (ABAC) will support a fine granularity of access to individual objects.

RBAC is an approach to restricting system access to authorised users. The RBAC access-control mechanism is defined around roles and privileges. Permission to perform certain operations is assigned to a specific role. Staff acquire the permission to perform a particular system function by being assigned the respective role. As users are not assigned permissions directly, but acquire them through their roles, management of individual staff rights becomes a matter of assigning roles to staff accounts.

Security tests of RBAC implementations must consider the following:

- a subject (staff or automated agent) can have multiple roles,
- a role can have multiple subjects,
- a role can have many permissions,
- a permission can be assigned to many roles.

RBAC is particularly well suited to enforcing Separation of Duties (SoD) requirements. SoD ensures that two or more staff members must be involved in authorising critical operations. A key principle of SoD is that no individual staff member should be

able to commit a breach of security through dual privilege. This implies that no staff member may hold a role that exercises audit, control or review authority over another concurrently held role by the same staff member. This must be considered during security testing in cases where SoD is relied upon.

In addition to the testing of secure RBAC settings, a policy on granting and revoking the assignments must be in place.

During the lifecycle phase, special care must be taken with setting up RBAC roles, subject and permissions for commercial-off-the-shelf (COTS) products at the beginning of the integration and validation lifecycle phase. This is due to devices coming with default user groups and user accounts, including default passwords. These must be identified and appropriately assigned, even if the respective functionality is not used.

As an example, we will briefly consider a board-level management engine. Access to the functionality of a board-level management engine is role based and must suit the needs of large data processing service centres with hundreds of hardware boards that have to be configurable at a level below the operating system. Correct handling of this role-based access, which allows complete hardware reconfigurations (e.g. replacing firmware without the involvement of the operating system) must be tested.

Accordingly, a very first test will reveal e.g. whether an Intel Management Engine (ME), which is typically part of all motherboards containing the Intel Active Management Technology (or similarly the AMD Secure Technology) has a password set. Even if the ME functionality is not used in a system, it can be used by an attacker to turn the computer on and off, and to login remotely into the computer regardless of whether an operating system is installed or not.

IEC 62351-8:2020 on 'Power systems management and associated information exchange – Data and communications security – Part 8: Role-based access control for power system management' addresses the ABAC topic in the context of the Smart Grid multi-part standards IEC 61850 and IEC 62351.

### 8.2.2 Testing of Access Control List (ACL) implementations

An Access-Control List (ACL) supports the implementation of a discretionary access-control system via a list of permissions attached to an object. The ACL can be with respect to a computer file system, networking or SQL implementations.

Each accessible object contains an identifier to its ACL. The privileges or permissions determine specific access rights, such as whether a user can read from, write to or execute an object.
For networking devices that support ACLs, the list can be configured to control both the inbound and outbound traffic. Accordingly, they must be tested similarly to firewall rules.

Assuming that a review of the ACL concepts confirms a robust design with regard to access restrictions, the security test validates that the restrictions are indeed effective, e.g. no privilege escalation is possible based on currently known vulnerabilities.

### 8.2.3 Testing of Attribute-Based Access Control (ABAC)

A drawback of Role-Based Access Control (RBAC) is that it can lead to role explosion. Attribute-Based Access Control (ABAC), also known as Policy-Based Access Control or Claims-Based Access Control (Microsoft-specific term) defines an access control scheme, whereby access rights are granted to staff members through the use of policies which combine attributes. Different types of attributes can be used, including:

- user attributes
- resource attributes
- object attributes
- environment attributes.

The ABAC scheme supports Boolean logic. This allows for more sophisticated access requests like: IF the requestor is maintenance staff, THEN allow update of set-points. This can be refined further,

e.g. to allow the update of set-points only for a specific object or group of objects (e.g. a 'View' in OPC Unified Architecture terms).

Accordingly, ABAC is considered an advanced, next-generation access control scheme, because it provides dynamic, fine-granular and context-aware access control to resources. ABAC supports access control policies that include specific attributes from different digital systems. The different fine-granular policies are needed in order to securely resolve access control requests and also to achieve regulatory compliance.
Security testing for the correct implementation of ABAC schemes includes Policy Enforcement Points (PEPs, which generate access requests), Policy Decision Points (PDPs, which evaluate incoming access requests) and Policy Information Points (PIPs, which provide information from external resources), e.g. databases.

As ABAC can be used to apply attribute-based, fine-grained authorisation to the Application Programming Interface (API) functions or object methods, the respective APIs must be considered during security testing together with testing of the PEPs, PDPs and PIPs.

Due to its generic nature, the ABAC scheme can be applied to ensure application security of proprietary safety critical applications, Content Management Systems (CMS), Enterprise Resource Planning (ERP), web applications, big data applications (e.g. with Hadoop) and others. This requires security tests in the respective proprietary software context, web development context, etc. In I4.0/IM manufacturing, one specific challenge is the development and testing of reusable ABAC access control schemes that integrate well with the interoperability solutions of OPC UA already agreed.

For interoperability in the I4.0/IM context, the XACML (eXtensible Access Control Markup Language) can be deployed. This allows for part of the access control-related security tests to be performed at a generic level and with tool-based support, as the XACML is completely based on XML.

### 8.2.4 Testing of Multi-Factor Authentication-based (MFA-based) Access Control

With Multi-Factor Authentication (MFA), a user of digital equipment is granted access only after successfully providing two or more pieces of evidence (factors) to an authentication mechanism. A factor can be:

- **Possession Factor** – a physical object in the possession of the staff member, such as a Near-Field Communication (NFC) card, a USB stick with a secret token or a card with a cryptographic chip;
- **Knowledge Factor** – a secret known to the staff member, such as a password, a PIN;
- **Inherent Factor** – biometric characteristics (fingerprint, face, voice, iris) or behavioural biometrics such as keystroke dynamics;
- **Location Factor** – based on the physical location of staff member, e.g. hard-wired to a test-bay network or in the proximity of a given GPS location.

Security testing of these factors comprises the verification of policies applied for managing MFA-based access control schemes (e.g. issuing of Possession Factors), secure implementation (e.g. of card-based solutions), the strength (length and complexity) of the Knowledge Factor and the secure handling of biometric characteristics.

## 8.3 Security testing for service providers

Digital services providers include many flavours of '... as a Service' business offers, e.g. 'Platform as a Service', 'Software as a Service' (SaaS as a software licensing and delivery model) or hosted virtualised environments. Security testing of these 'as a Service' offers can be based on combinations of testing approaches addressed in previous sections, e.g. for the operating system software, standard software, application software and networking software.

The involvement of multiple business partner companies, as addressed by the draft ISO/IEC 24392 on Industrial Internet Platforms (IIP) additionally requires testing of security controls along the supply-chains involved in the IIP. As an example, one business partner may place a tender for the implementation of a specific Printed Circuit Board (PCB) and the IIP will assist with communication of the offer to suitable suppliers and selection of a best quality of service offer. Integration and assembly of the PCB, e.g. for a consumer IoT device, may require real-time access to smart sensors and IIoT actuators. Accordingly, the IoT-specific security testing must be considered together with the secure exchange of both signalling data and order/purchasing transactions. The next two sections will address security testing of cloud service providers (§8.3.1) and big data service providers (§8.3.2).

### 8.3.1 Security testing for cloud service providers

During the testing and application of cloud services, cloud service providers and customers must comply with the following requirements:

- **Responsibility for security management remains unchanged.** The responsibility of information security management should not be transferred with service outsourcing. No matter whether customer data and business data are located in the internal information system or the cloud platform of the cloud service provider, the customer is the person ultimately responsible for information security.

- **Ownership of the resource does not change.** The data, equipment and other resources provided by the customer to the cloud service provider, as well as the data and documents collected, generated and stored during operation of the customer's business system on the cloud platform, shall belong to the customer, and the customer's access, utilisation and control of these resources and other rights are not restricted.

- **The level of security management remains unchanged.** Cloud computing platforms that carry customer data and services should be managed in accordance with information system security management requirements, and cloud service providers that provide cloud computing services to customers should comply with information system security management standards.

- **Adhere to the principle of test first and then apply.** Cloud service providers should have the ability to ensure the security of customer data and business systems and pass relevant security tests.

'Cloud service provider'-related security testing standards include but are not limited to:

- ISO/IEC 23168:2018 IT — Cloud computing — Framework of trust for the processing of multi-sourced data;
- ISO/IEC 27018:2019 IT — Security techniques — Code of practice for the protection of personally identifiable information (PII) in public clouds acting as PII processors;
- ISO/IEC 17788:2014 IT — Cloud computing — Overview and vocabulary;
- ISO/IEC TR 23188:2020 IT — Cloud computing — Edge computing landscape.

### 8.3.2 Security testing for big data-related service providers

In addition to the cloud security testing-specific considerations, testing of big data-related services places challenges on demonstrating compliance with privacy requirements. ISO/IEC 29190 provides a Privacy Capability Assessment Model, but without a special focus on big data.

ISO/IEC JTC1/SC42 defined a Big Data Reference Architecture (BDRA) as ISO/IEC 20547-3:2020. However, the corresponding security processes and capabilities standards that are expected to be jointly developed by JTC1/SC27 WG4/WG4 and ISO/IEC JTC1/SC42 are not yet in place.

Special care must be taken when testing the de-identification approach that is claimed to be supported by big data services. De-identification may be completely irreversible or may be

implemented in such a way that a trusted group of users may still be able to undo the de-identification up to a predefined level, e.g. in order to be able to associate different data sets to the health status of a person or characteristics of a cyber-physical system, without the ability to identify the original object.

A further focus on big data cybersecurity testing should be scalability. This requires efficient approaches that can cope with an exponential growth of data sets.

ISO/IEC 27018:2019 on the code of practice for protection of personally identifiable information (PII) should be considered for big data security tests that demonstrate compliance with General Data Protection Regulation (GDPR) in the EU or provisions similar to GDPR in other geographical regions.

# 9. Security testing in terms of security assurance

With the updated definitions of critical infrastructure, supplier companies also become a focus of the corresponding stringent regulation. The subsequent sections address some of the related concerns.

## 9.1 Test input information requirements

Before starting a security test or requesting an independent security evaluation, it is necessary to determine the test input information. This will depend on the intended approach, e.g. black-box, grey-box based on OSSTMM with a focus on behavioural testing or in-depth evaluation e.g. for a security target (one device) or protection profile (family of devices) according to the ISO/IEC 15408 (Common Criteria), see section 11.6.

While a comprehensive user manual may be sufficient for black-box and grey-box evaluations, more in-depth evaluations will need the availability of the source code of the software and firmware involved. This must be considered along the supply chain, especially where IIoT equipment or edge devices with embedded software are purchased before security tests are performed or if the tests will be performed by or on behalf of the purchaser or integrator company. These security test input information requirements should be handed over to the customer, together with further stipulations in line with ISO/IEC 27036-2.

The input information for testers should also comprise documentation on security-related tests already performed, including on security vulnerabilities.

## 9.2 Liability constraints

Vendors must ensure that the delivered software was indeed tested and does not contain an unreasonable number of security vulnerabilities. As a prerequisite, the purchased or custom-developed software needs to have an unambiguous marking of the shipped versions. If it is the case that so-called alpha software versions, beta software versions or release candidate software are deployed for preliminary functional tests or integration tests, these temporary versions must be disposed of before final integration tests. This should be documented as part of the trustworthy handover, including between suppliers and integrators from different countries. Trustworthy handover will implicitly help to avoid non-repudiation-related issues.

Comprehensible and traceable documentation about the software from different suppliers, as well as software that has been internally developed, must be provided and maintained as part of smart manufacturing systems. In the event of a vulnerability being detected during factory or plant operation, this will allow tracking of the root cause to the software component concerned and of the security integration tests and supplier product security tests that failed to detect the vulnerability.

Service Level Agreements (SLAs) must be in place in order to ensure that security vulnerabilities detected during deployment can and will be handled with acceptable time delays. Typically, this requires legal contracts, which ensure that sufficient security expert resources are available at the software and IIoT supplier companies potentially concerned. SLAs should address both patch management and updates management, as timely updates to more mature and well-tested new software versions may help reduce the number of vulnerabilities.

The ability to trace the triggering path for a security vulnerability may require forensic readiness to be in place. This will support efficient error finding and avoid legal disputes on inappropriate use or insecure configurations.

With regard to liability and contractual obligations, the potential impact on personal damage or property should be considered when agreeing on security test-related efforts. The effort should be graded in line with the potential impact and agreed penalties.

## 9.3 Security testing in accordance with IECEE

This section explains the IECEE approach in general and when specifically applied to a selection of IEC 62443 series standards.

### 9.3.1 IEC System of Conformity Assessment Schemes

The IEC System of Conformity Assessment Schemes for Electro-technical Equipment and Components (IECEE System) [13] is an assessment scheme already applied to many processes, products and solutions outside the security domain.

Recognising the need to facilitate international trade in sound electrical devices and components and to further provide simplicity and convenience to manufacturers and other users for standard certification/compliance, the IEC devised the IECEE CB Scheme for Mutual Recognition of Test Certificates for Electro-technical Equipment and Components (CB Scheme). This IECEE CB scheme is a multilateral certification system based on IEC standards and adopts the principle of mutual recognition/reciprocal acceptance of safety test reports and certificates for electrical and electronic equipment, devices and components to obtain certification or approval at national levels around the world [13].

Overall, the IECEE covers 23 categories of electrical and electronic equipment and testing services and addresses the safety, quality, efficiency and overall performance of components, devices and equipment for homes, offices, workshops, health facilities and so forth. As the CB scheme is intended for certification based on IEC standards, where national standards are used, differences are taken into account; however, it is assumed that there is a high degree of harmony between these national standards and the relevant IEC counterparts. Certification relies on a global network of approved CB Testing Laboratories (CBTL) in participating countries. Here, the products are tested in accordance with applicable technical standards and the results are submitted to an 'Issuing and Recognising' National Certification Body (NCB) that can authorise legal access to CBTL-tested products without additional testing [14]. After successful testing, an Issuing and Recognising NCB issues a CB Test Certificate to inform other NCBs that the tested electrical product(s) has/have been tested according to applicable standard(s) and is/are found to be acceptable for use in the IECEE. It must be noted that this CB Test Certificate is only valid with the IECEE-documented CB Test Report included, in accordance with the agreed format.

#### 9.3.2 IEC System of Conformity Assessment for IEC 62443

As an example, [13] presents an overview of how to apply the CB scheme to assessments in accordance with IEC 62443 Security for Industrial Automation and Control Systems series of standards, to result in an IECEE Certificate of Conformity – Industrial Cyber Security Capability. IEC 62443 is derived from the ISO/IEC 27000 ISMS family of standards and adapted to address and improve the safety, availability, integrity and confidentiality specifically of Industrial and Control Systems (IACS) and their sub-components.

The IEC 62443 series of standards can be utilised across industrial control segments and is fast becoming a key standard in the industry. The IEC 62443 series of standards generally specify requirements for security capabilities, which may be technical capabilities (security mechanisms) or process capabilities (human procedures) [13]. The achieved grading allows the user to understand the minimum security capabilities of the device.

IEC 62443 conformance assessment consists of evaluating the security capabilities that a manufacturer (Applicant) uses to develop, integrate and/or maintain specific products or solutions. There are two possible evaluations [13]:

| | IEC 62443-2-4 | IEC 62443-3-3 | IEC 62443-4-1 | IEC 62443-4-2 (Future consideration) |
|---|---|---|---|---|
| **Process** | ✓ Scenario 1 | | ✓ Scenario 1 | |
| **Product** | ✓ Scenario 2 | ✓ Scenario 1* Optionally in conjunction with an IEC 62443-4-1 scenario 2 certificate*** | ✓ Scenario 2 possible in conjunction with an IEC 62443-3-3 or IEC 62443-4-2 scenario 1 certificate** | ✓ Scenario 1* in conjunction with an IEC 62443-4-1 scenario 2 certificate**** |
| **Solution** | ✓ Scenario 3 | | | |

**Figure 4** IECEE scheme for IEC 62443 – Certificate of Conformity Scenario matrix [13]

1. Evaluation of the ability of the Applicant to provide IEC 62443-compliant security capabilities. The focus here is on evidence that supports the submissions (e.g. process/product documentation) made by the Applicant. The submissions typically contain specific requirements and processes used to implement the security capabilities that are being tested.

2. Evaluation to demonstrate that these capabilities have been applied to either:
   a) a specific product, or
   b) a specific solution.

Following the IECEE testing procedure, an IECEE Certificate of Conformity can be issued under two scenarios:

1. Scenario 1 – **Capability Assessment**: an assessment of a set of technical capabilities or process-oriented capabilities.

2. Scenario 2 – **Application of Capabilities Assessment**: use of a Scenario 1 process-oriented capability for a specific product or solution.

Figure 4 is a matrix that depicts how the different parts of IEC 62443 are applied in these two scenarios.

The Applicant is required to: (a) identify the IEC 62443 sub-standards that should feature in their assessment; (b) select the specific security requirements from the identified sub-standards to be evaluated as a part of their assessment; and (c) identify the product or solution against which the assessment should be conducted.

[13] Further lists the following possible Certificates of Conformity according to IEC 62443:

- Product Capability Assessment (IEC 62443-2-4, IEC 62443-3-3, IEC 62443-4-2)
- Process Capability Assessment (IEC 62443-2-4, IEC 62443-4-1)
- Solution Capability Assessment (future consideration)
- Product Application of Capabilities Assessment (IEC 62443-4-1)
- Process Application of Capabilities Assessment (future consideration)
- Solution Application of Capabilities Assessment (IEC 62443-2-4).

As parts of the IEC 62443 standard series are still being developed or updated, updates of the related IECEE schemes will be provided in a future revision of this security tests white paper.

## 9.4 TeleTrusT evaluation method for IEC 62443-4-2

In June 2019, the non-profit IT Security Association Germany (TeleTrusT) issued its evaluation method for IEC 62443-4-2:2019 [15] as part of its Security for IACS initiatives. Part 4-2 of IEC 62443 is on technical security requirements for IACS components. Accordingly, the method proposes an 'evaluation approach to verify the fulfilment of the requirements' of IEC 62443-4-2. It is not a certification scheme, but the basis for a conformity assessment [15].

The evaluation method primarily pertains to the technical qualities of a component, while assuming that component development was based on a development process in accordance with IEC 62443-4-1. This implies that results (deliverables) of the development process will be made available and can be resorted to during component evaluation.

In terms of the IECEE (see Section 9.3), the evaluation method perceives itself as the implementation of a product certification in accordance with IEC 62443-4-2 and scenario 1.

**IEC 62443-4-2 distinguishes four categories of devices:**

- embedded devices (PLC, SIS controllers, DCS controllers, sensors),
- host devices (notebooks, PCs, workstations),
- network devices (industrial routers, switches),
- applications (configuration software, historian).

Often these are COTS components, so risk mitigations are needed. For a component evaluation, the following are determined first:

1. the Security Level Capability (SL-C) to be achieved;

2. the selection of Component Requirements (CR) to be met.

**Note:**
**The current version of the evaluation method takes into account Security Levels SL-1 to SL-3, thus addressing evaluations in the medium/substantial assurance range. The most stringent SL-4 assumes an attacker with high potential, high motivation and high resources.**

The TeleTrusT documentation [15] points out that in the event of vulnerabilities being found, an evaluation in line with these criteria must be performed.

The evaluation criteria for a complete attack shall use at least:

- the time needed (for the design of the attack and for its execution),

- the expertise required,

- knowledge of the component (e.g. publicly accessible or available only to the development team),
- the window of opportunity,

- the attacker's equipment.

Additionally, a vulnerability assessment according to the Common Vulnerability Scoring System (CVSS) can be performed. New findings could also be rated in line with CVSS.

The evaluation facility should orientate its own testing methods by ISO/IEC 17025 (see also Section 10).

The normative appendices A and B describe the component specification and evaluation report requirements. The comprehensive normative appendix C lists the abbreviations and explanations of the Component Requirements (CR) of IEC 62443 grouped by the seven Foundational Requirements (FR). The informative appendices describe the vulnerability assessment and reuse of deliverables from an IEC 62443-4-1 assessment.

The above criteria are also addressed by the ISA Security Compliance Institute with regard to functional security assessment for components [16]. The ISA Security Compliance Institute (ISCI) functions as an operational group within ISA's Automation Standards Compliance Institute (ASCI). ISA provides management services to ASCI to

support programs established by compliance institutes such as the ISA Security Compliance Institute (ISCI). As a not-for-profit automation controls industry consortium, ISCI manages the ISASecure™ conformance certification program [17].

ISCI has published Certification Requirements Specifications for the assessment of component requirements according to ISA IEC 62443-4-2. This assessment scheme is published as 'CSA-311 Component Security Assurance – Functional security assessment for components' [16]. Its component requirements specification is applicable for software applications, embedded devices, host devices and network devices. It is structured into:

- **Overview table** of Foundational Requirements (FR1 – FR7), including Reference Name and applicability according to the Security Level (SL).

- **Detailed description table**, including Requirement Description, Validation Activity, if the Validation is required by an independent test, the reference on the source of IEC 62443, the Capability Security Level, additional Rational and Supplemental Guidance.

As of 2021, the scope of the ISASecure certifications includes assessment of off-the-shelf IAC products and IAC product development security lifecycle practices. ISASecure does not offer assessments for integrator site engineering practices or asset owner operations and maintenance practices. ISASecure certifies off-the-shelf systems; not the on-site engineered/deployed systems.

ISCI offers three certifications with four security assurance levels (SAL) in alignment with ISA/IEC 62443.

## 9.5 ISO/IEC 15408 (Common Criteria)-based security testing

### 9.5.1 Common Criteria-based security testing and evaluation

The exponential growth of technology and diversity of technological concepts, compounded by the increase in cybersecurity breaches, has prompted the need for assurance that products and systems provide adequate security.

**Historical evolution**

In response to increasing IT security threats, various countries began their initiatives to develop evaluation criteria that build upon the concepts of Trusted Computer System Evaluation Criteria (TSEC), for example, Europe – ITSEC (1991); Canada – CTCPEC (1993); US – Federal Criteria (Draft 1993) [18]. In the face of this, the Common Criteria for Information Technology Security Evaluation (abbreviated to Common Criteria or CC) was developed for universal use to facilitate consistent and objective evaluations of hardware, software and firmware. For this purpose, the CC define an IT security evaluation methodology that serves as both a useful guide for developing products and systems with IT security functions and a guide for procuring commercial products and systems with security functions.

The goal of the CC standard is to provide overall assurance that the process of specification, implementation and evaluation of a product with security functionality has been conducted in a rigorous, standard and repeatable manner and at a level that corresponds to its target use environment.

**CC security testing philosophy**

The CC agree that the threats to security and organisational security policy commitments should be described clearly, and corresponding security countermeasures should be demonstrated to be capable of fully meeting their purpose. Measures should be taken that are helpful in identifying vulnerabilities, eliminating and mitigating the consequences and/or notifying the exploited vulnerabilities. Security tests, such as pen tests, which can reduce the probability of vulnerabilities should be adopted.

In CC, assurance is based on the evaluation of IT products (before being trusted). The assurance means that IT products fulfil their security objectives. CC include two types of evaluation: Security Target (ST)/Target of Evaluation (TOE) evaluation, and Protection Profiles (PPs) evaluation.

### Structure of CC documentation

CC documentation is organised into three parts, where key concepts such as the Target of Evaluation (TOE), Protection Profile (PP), Security Target (ST), Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) are used to describe the evaluation methodology.

In ISO/IEC 15408 part 1 **Introduction and General Model**, the general concepts and principles of IT security evaluation are defined and a general model of evaluation is presented.

In ISO/IEC 15408 part 2 **Security Functional Requirements,** a set of security functional components is established as the basis for security functional requirements that describe the desired security behaviour expected of a TOE.

Finally, in ISO/IEC 15408 part 3 **Security Assurance Requirements,** a set of assurance components are defined as the basis for security assurance requirements of the TOE. These assurance packages are referred to as Evaluation Assurance Levels (EALs), which is a scale by which confidence in the security of IT products and systems are rated. In addition to these three parts, the Common Methodology for Information Technology Security Evaluation (CEM) was created to provide the methodology for IT security evaluation, based on recommendations of the CC.

### Graded security testing and assessment requirements

The EALs are considered a key part of the CC, as they provide a numerical rating that indicates the rigour and depth of an evaluation, which is critical to the CC audience (developer, evaluator and consumer). For example, for a developer, a higher EAL means additional documentation, detail and support for the security analysis are needed. At the same time, to the evaluator, this means a more testing and rigorous analysis is needed. For the consumer, this means a high level of confidence in the system's security functions is gained [19]. The CC lists seven EALs, with EAL 1 being the most basic and EAL 7 being the most stringent. The levels denote the different development and testing approaches followed, generally with an increasing number and/or intensity of formal testing for higher levels. This suggests that products compliant with a higher EAL are more secure, though not completely secure. A description of the testing associated with each EAL is given below:

### Evaluation Assurance Level 1 (EAL1) – functionally tested

EAL1 is the lowest assurance level for indicating confidence in the security functions of a system. The associated evaluation is applicable where a low level of confidence in the operation is required for the TOE, as a part of an obligation for demonstrable due care. The rigour of this evaluation is geared to detecting obvious errors, but potential threats to security are not deemed as serious. The analysis is supported by independent testing to assess the security behaviour of the TOE, by using the functional and interface specification of the TOE itself.

### Evaluation Assurance Level 2 (EAL2) – structurally tested

EAL2 requires minimal support from the developer; specifically, the developer should provide the design information and test results. Support beyond this, such as where more effort is demanded of the developer, is not applicable for this EAL and should not require a substantially increased investment in cost or time. Analysis of the security functions of a TOE here uses its functional and interface specification, as well as the high-level design of the TOE subsystems. As in EAL1, independent testing of the security functions is performed and a search for obvious vulnerabilities is conducted. Additionally, the evaluator performs a review of the evidence of black-box testing, as provided by the developer. Some network devices, such as certain firewalls, require and have achieved at least EAL2 certification.

### Evaluation Assurance Level 3 (EAL3) – methodically tested and checked

EAL3 is applicable where developers or users require a moderate level of independently assured security at the design stage, without substantial re-engineering of sound development practices. This entails a thorough investigation of the TOE and its development processes. An EAL3 analysis represents a meaningful increase in assurance from EAL2 – similarly, systems here are examined for obvious vulnerabilities. However, the analysis is

supported by grey-box testing and selective independent confirmation of the developer test results. Development environment controls and TOE configuration management are also referenced in this analysis. Operating systems, such as SUSE Linux Enterprise Server v.8, are known to have at least EAL3 certification.

**Evaluation Assurance Level 4 (EAL4) – methodically designed, tested and reviewed**

EAL4 is applicable where there is a requirement for a moderate to high level of independently assured security in conventional commodity products, and where there is willingness to incur some additional security-specific engineering costs. This is the highest assurance level at which it may be economically feasible to retrofit to an existing product line. Here, a developer can gain maximum assurance from positive security engineering, based on good commercial development practices, which are rigorous but not overly specialised. EAL4 analysis is supported by the low-level design of the TOE modules, and a subset of the implementation. Development controls are supported by a lifecycle model, identification of tools and automated configuration management. Similar to lower EALs, an independent search for obvious vulnerabilities is also conducted. Typically, EAL4 certification is required for smart cards/operating systems (e.g. of Infineon Technologies) and microcontrollers (e.g. from ST Micro, Infineon Technologies, AMTEL smartcards).

**Evaluation Assurance Level 5 (EAL5) – semi-formally designed and tested**

EAL5 represents a significant increase in assurance from EAL4. While both permit a developer to gain maximum assurance from security engineering based on rigorous commercial development practices, EAL5 is supported by moderate application of **specialised** security engineering techniques (potentially incurring unreasonable costs). Here, analysis includes all of the implementation and assurance is supplemented by a formal model and a semi-formal presentation of the functional specification and high-level design, and a semi-formal demonstration of correspondence. The search for vulnerabilities must ensure relative resistance to penetration attack. Modular design is required and covert channel analysis may also be required. Operating systems/smart cards (e.g. from Sun Java Card Virtual Machine) have EAL5 certification.

**Evaluation Assurance Level 6 (EAL6) – semi-formally verified design and tested**

EAL6 permits a developer to gain high assurance from the application of specialised security engineering techniques in a rigorous development environment to produce a premium product for protecting high-value assets against significant risks. Here, it is justifiable for the testing process to incur additional costs. EAL6 analysis is supported by a modular and layered approach to design and a structured presentation of the implementation. The independent search for vulnerabilities must ensure high resistance to penetration attack and the search for covert channels must be systematic. The development environment and configuration management controls are further strengthened. As an example, Green Hills Software's INTEGRITY-178B RTOS has been certified to EAL6.

**Evaluation Assurance Level 7 (EAL7) – formally verified design and tested**

EAL7 is applicable to the development of specialised security products for application in extraordinarily high-risk situations, which justifies the typically higher costs incurred for the evaluation process. Practical application of EAL7 is currently limited to products with tightly focused security functionality that is amenable to formal analysis. For an EAL7 evaluation, the formal model is supplemented by a formal presentation of the functional specification and a high-level design to show correspondence. Requirements include evidence of the developer 'white-box' testing and complete independent confirmation of developer test results. As examples, the Tenix Interactive Link Data Diode Device and the Fox-IT Fox Data Diode (one-way data communications device) have achieved EAL7 certification.

### 9.5.2 Common Criteria-related Chinese standards

GB/T 18336 is a Chinese standard which directly maps to the aforementioned ISO/IEC 15408. Beyond GB/T 18336, there are further Chinese

standards which address security testing and evaluation. The following subsections provide a brief overview of Chinese standards relating to international security testing and evaluation standards.

### IT – security technology – methodology for IT security evaluation, GB/T 30270-2013

The Chinese standard GB/T 20370-2013 adopts the international standard ISO/IEC 18045:2005. The methodology for IT security evaluation described in this standard is limited to EAL1~EAL4. It does not provide EAL5~EAL7 or other evaluation guidelines.

The evaluation methodology provided by this standard should be applied when using ISO/IEC 15408, as it is the supporting standard of ISO/IEC 15408.

### Information security technology – common methodology for information systems security assurance evaluation, GB/T 30273-2013

This standard describes the evaluation activities that evaluators need to complete when using the criteria defined by the Evaluation Framework for Information Systems Security Assurance of the four-part GB/T 20274. GB/T 30273 provides guidance for evaluators' evaluation behaviours and activities in specific evaluation scenarios. The standard applies to evaluation of the security of information systems and evaluation of ISPP/ISST.

### Information security technology – assessment criteria for information security service capability, GB/T 30271-2013

This standard specifies the service process model and evaluation criteria for the service capabilities of information security service providers.
This standard applies to evaluation of the capabilities of information security service providers, as well as to service providers, to provide guidance on their own capabilities.

### Information security technology – implementation guide for information security risk assessment, GB/T 31509-2015

This standard specifies the process and method of implementation for information security risk evaluation.

The standard applies to the management of information security risk evaluation projects of non-confidential information systems by various security evaluation agencies or evaluated organisations. It guides the organisation, implementation and acceptance of risk evaluation projects.

### Information security technology – technical requirements and testing and evaluation approaches for network-based intrusion detection system, GB/T 20275-2013

This standard specifies the technical requirements and test evaluation methods of the network intrusion detection system. The requirements include security function requirements, self-security function requirements, security assurance requirements and test evaluation methods. Classification requirements of network intrusion detection systems are proposed.

This standard applies to the design, development, testing and evaluation of network intrusion detection systems.

### Information security technology – testing and evaluation approaches of network and terminal isolation products, GB/T 20277-2015

This standard specifies the test and evaluation methods for network and terminal isolation products, based on the technical requirements of GB/T 20279-2015.

This standard applies to the testing and evaluation of network and terminal isolation products developed in accordance with GB/T 20279-2015 security level requirements.

### Information security technology – testing and evaluation approaches for network vulnerability scanners, GB/T 20280-2006

This standard specifies the evaluation methods for network vulnerability scanning products, including the content of network vulnerability scanning product evaluation, evaluation function objectives and testing environment, and gives specific objectives that must be met for basic product functions, enhancement functions and security assurance requirements.

The purpose of this standard is to provide technical support and guidance for the development, production and certification of network vulnerability scanning products.

**Information security technology – security technical requirements and testing and evaluation approaches for firewall, GB/T 20281-2015**
This standard specifies the security technical requirements, test evaluation methods and security level division of firewalls.

This standard applies to the design, development and testing of firewalls.

**Information security technology – technical requirements, testing and evaluation approaches for information system security audit product, GB/T 20945-2013**
This standard specifies the technical requirements and test evaluation methods for information system security audit products. The technical requirements include security function requirements, their own security function requirements and security assurance requirements. The classification requirements for information system security audit products are proposed.

This standard applies to the design, development, testing and evaluation of information system security audit products.

## 9.6 Open Source Security Testing Methodology Manual (OSSTMM)

OSSTMM is also considered as a practical international security testing approach. The OSSTMM manual is open source (peer-reviewed). Users are encouraged to submit information relating to vulnerabilities uncovered/researched for future inclusion. However, the latest version (currently v4) is only available to members of ISECOM. OSSTMM was initially developed by Pete Herzog and is intended for Internet Security and Testing. The methodology is referred to as the OSSTMM audit and is described as 'an accurate measurement of security at an operational level that is void of assumptions and anecdotal evidence' [20]. In particular, it provides rules and regulations for penetration testing, ethical hacking, security assessments and vulnerability assessments. The domains covered include internet technology security, communications security, human security, process security, wireless security, processes and physical security. Within these rules and regulations, recommendations are made for tools to be used and how to use said tools, as well as how to format the resulting report.

**The OSSTMM defines a 7-step set of activities for security testing that must be completed before the process begins:**

1. **Project scope** – denotes the total possible operating security environment for any interaction with any asset. Essentially, this is the environment within which the test is undertaken. This includes the targets, tools, testing methodologies, aim of the test, etc. to be considered. Also referred to as the rules of engagement, this assists the Analyst in determining the testing strategy.

2. **Confidentiality and non-disclosure assurance** – legal agreement between the Analyst and the client, to ensure the confidentiality of the entire engagement (scope, discovery, etc.).

3. **Emergency contact information** – the contact details of persons who must be contacted in case of emergency and/or discovery of critical vulnerabilities or suspected criminal findings.

4. **Statement of work change process** – the test and target results agreed on between the Analyst and the client.

5. **Test plan** – should not contain plans, processes, techniques or procedures which are outside the area of expertise or competence level of the Analyst.

6. **Test process** – addresses the legal and ethical boundaries of the security testing procedure. For example, safety must be prioritised. Testing procedures must be secure in themselves and must be within the agreed scope. The Analyst must have the requisite knowledge to use the tools and conduct the tests.

7. **Reporting standards** – concerns a description of the agreed format for reporting findings, as well as what not to include in the report/what should be separately reported.

Once these activities have been completed, the OSSTMM audit can be used to further guide the Analyst to achieve a consistent, repeatable and reliable process.

As indicated in Figure 5, the OSSTMM discerns 6 types of security test audits based on the amount of information the tester knows about the targets, what the target knows about the tester or expects from the test, and the legitimacy of the test.

**Blind Audit**: the Analyst engages the target with no prior knowledge of its defence, assets and channels. But the target is prepared for the audit, knowing in advance all the details of the audit. It challenges the scanning, network sniffing and discovery abilities of the Analyst.

**Double Blind Audit:** the Analyst engages the target with no prior knowledge of its defence, assets and channels. The target is not notified in advance of the scope of the audit or the test vectors. A Double Blind Test challenges both the skills of the Analyst and the preparedness of the target.

**Grey Box Audit:** the Analyst engages the target with limited knowledge of its defences and assets and full knowledge of channels. The target is prepared for the audit, knowing in advance all the details of the audit. This can include a Vulnerability Test (e.g. with standard tools such as Nessus or Qualys) and is most often initiated by the target as a self-assessment. The breadth and depth depend upon the quality of the information provided to the Analyst before the test, as well as the Analyst's applicable knowledge.

**Double Grey Box Audit:** the Analyst engages the target with limited knowledge of its defences and assets and full knowledge of channels. The target is notified in advance of the scope and time frame of the audit but not the channels tested or the test vectors.

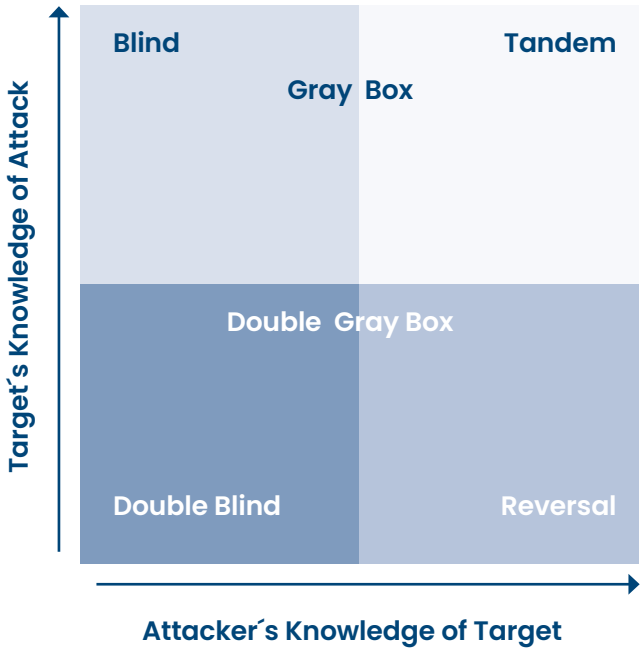


**Figure 5** — Common OSSTMM test types [20]

**Tandem Audit:** the Analyst and target are prepared for the audit, both knowing in advance all the details of the audit. A tandem audit tests the protection and controls of the target. However, it cannot test the preparedness of the target to unknown variables of agitation. This is often known as an In-House Audit or Crystal Box Test.

**Reversal Audit:** the Analyst engages the target with full knowledge of its processes and operational security, but the target knows nothing of what, how, or when the Analyst will be testing. The true nature of this test is to audit the preparedness of the target to unknown variables and vectors of agitation. The breadth and depth depend upon the quality of information provided to the Analyst and the Analyst's applicable knowledge and creativity (see also 5.2.4 in the context of skills for pen testing).

## 9.7 Red team-blue team exercises

Defence against security attacks is often assessed and trained as part of so-called 'red team-blue team exercises'. As was described in Section 9.6 for the OSSTMM methodology, both the attacker's knowledge of the target and the

target's knowledge of the (ongoing) attack are relevant for the purpose of a security test.

The red team is planning a simulated real-world attack with the purpose of impacting physical aggregates or processes or a Cyber Physical Production System (CPPS). Thus, knowledge of both the security domain and the target domain (e.g. a manufacturing facility or power plant) are needed.

The blue team consists of security staff and domain specialists, who are supposed to ensure a continuously adequate security posture for manufacturing sites, plants or process. Often, for facilitating an adequate training of the blue team, it is irrelevant whether an impact materialises due to a security attack or due to a direct initiation of the impact. For example, in the event of some equipment set-points being manipulated or some process limits maliciously altered, this must be detected (by detective security controls) and corrected (by corrective security controls). For example, the blue team could detect manipulations based on alternative real-time monitoring capabilities and provide corrections by restoring previous values from backups. However, for subsequent forensic investigation into how the attack was successfully performed by the red team, the artefacts that led to the impact will be relevant (preparedness by appropriated preventive security controls for digital forensic readiness).

I4.0/IM technologies will facilitate the preparation and execution of red team-blue team tests, as the exercises can make use of digital twins, the Automation ML (AML) models of the CPPS and the RAMI 4.0 / IMSA (IEC/TC 65/JWG 21 Smart Manufacturing Reference Models) grade hierarchical descriptions of the assets. An adequate context for performing such red team-blue team tests is provided by the different innovative test facilities located at multiple cities in Germany and China.

**Note:**
**The focus of LNI 4.0 labs is more on interoperability and proof of concept, e.g. with regard to digital twins.**

In the context of I4.0/IM, red team-blue team testing must be targeted towards Cyber Physical Production Systems (CPPS) and towards Cyber Physical Systems (CPS) [21].

## 9.8 Fuzz testing

Fuzz testing is an automated software testing technique. The core of the fuzz testing technique involves the injection of unexpected, partially invalid and random data (sent to interfaces of a system). The data is injected either into network data flows, function arguments, memory or other locations where the injected data may reveal vulnerabilities due to implementation flaws.

As part of fuzz testing, the impact of injected data must be monitored. Impacts may include

- **a program crash,**
- **triggering of an exception handler,**
- **causing internal assertions to fire,**
- **causing memory leaks.**

There are different types of fuzzers, e.g. smart/dumb, genetical/mutation and white/black box. The fuzzers are applied for a variety of targets, e.g. for injecting data packets into a network or for performing fuzzing at the source code level of software programs (with or without virtual execution). A fuzzer is often provided as a framework that can be extended according to knowledge about the target, e.g. by including some heuristic knowledge in order to progress from a dumb fuzzer (with no or minimum knowledge about the target and environment) towards a smart fuzzer.

There exists a wide variety of tools and frameworks to help implement fuzzing tests that employ genetic algorithms to efficiently increase code coverage of the test cases. Examples include commercial tools such as the DEFENSICS suite from Synopsys, as well as open source solutions such as libFuzzer, FLUFFI and AFL (American Fuzzy Lop). libFuzzer, for example, is a library for coverage-guided fuzz testing, whereas FLUFFI (Fully Localised Utility For Fuzzing Instantaneously) is a distributed evolutionary binary fuzzer for pen-testers.

Many of the well-known technology companies and big open source projects rely on fuzz testing

to ensure the robustness of their implementations. This has helped to find many critical vulnerabilities in all major browsers, as well as in many big software packages, ranging from LibreOffice to the Linux kernel.

An important feature for industrial projects is the ability of fuzzing tools to provide a Software Development Kit (SDK) with Application Programming Interfaces (APIs) for different programming languages. This can be used to tailor the injection of fuzzing data according to industry protocols or even the proprietary network protocols used.

Fuzzing is expected to remain a key security testing technique for I4.0/IM due to increased interoperability and related potential vulnerabilities.
More detailed recent guidance on fuzz testing is provided by [22] on protocol fuzz testing as a part of the Secure Software Development Life Cycle and by [23] on effective fuzz testing for vulnerability research into Programmable Logic Controllers (PLCs).

[24] mentions fuzzying in the context of I4.0 for the security of Cyber Physical Systems (CPS), Industrial Control Systems (ICS) and IoT.

[25], [26] and [27] address security testing methods and techniques for industrial control devices, obstacles and solutions for practical fuzz testing and evaluating fuzz testing.

# 10. Requirements for security testing of equipment and tools

The following sections address source code level test tools (section 10.1) and AI-based test tools (section 10.2), which help in detecting software failures that may potentially present security vulnerabilities.

## 10.1 Source code-level security test tools

An effective way to avoid security vulnerabilities at the source code level is strict adherence to coding style guidelines. These coding guidelines, e.g. as provided at a general level by ISO/IEC TR 24772-1 and for individual programming languages by further standard parts of ISO/IEC TR 24772, are often elaborated and maintained by software and firmware vendors. Similar guidance applies for Hardware Definition Language (HDL)-based development.

Nevertheless, even with these coding style guides in place, the software developers involved may be at a different maturity level with regard to experience in applying the guidance. Peer reviews of software source code will help identify non-compliance with the coding style guides. These peer reviews may consider additional programming language-specific guidance, e.g. of ISO/IEC TR 24772-3 on avoiding vulnerabilities in the ANSI C programming language.

However, there may still remain corner cases and complex software API-related scenarios that are not easy to identify with limited-time manual reviews. As the coding style guidelines are well defined, adherence can be verified by dedicated programming language-specific tools. For example, adherence to the so-called MISRA C (already mentioned in 6.3) coding guideline, which was initially deployed for the automotive domain, can be verified with appropriate tools. These tools are either dedicated, e.g. MISRA C checkers, or they can be purchased as add-ons of more comprehensive source code-level test tool suites.

With regard to security testing, advanced source code-level analysis tools are essential in order to reduce the number of potential vulnerabilities and thus the attack surface. For source code-level test coverage there are two major categories of test utilities. Historically, static source code analysis tools were deployed, e.g. the LINT tools used already on Unix systems. More advanced – and often by orders of magnitude more expensive – source code-level test tools include Coverity or Polyspace, which perform a virtual execution of the source code or of selected source code modules and modelling of software module interfaces that are either not yet developed or part of a separate delivery scope. This allows for a considerably higher coverage, due to virtual execution of the source code, occasionally also called n-dimensional execution (as the parameters to functions are considered as dimensions whose values can be individually randomised). These dynamic tests may run for extensive time periods (e.g. several hours) on multiprocessor systems.

While execution of these dynamic (n-dimensional space) tests can be expensive in terms of licence fees for the dynamic source code test tools, e.g. with licence models per 100,000 lines of code (100 kLOC) and a need for adequate test equipment, the benefits are also high. Part of the software failures can be due to security vulnerabilities, which will be detected together with their root cause. Attackers with no access to the source code will not be able to find security vulnerabilities in a similar way, as they have only binary images. Nevertheless, image-level tools may also detect some software interface when handling security vulnerabilities that could have been identified by the dynamic source-level testing tools, if deployed during software or HDL development and if addressed by the available/purchased features of the test suite software.

## 10.2 AI-based security test tools

Artificial Intelligence (AI)-based approaches are gradually applied with different aims in the security domain. While this is still debated in principle [28], there are already several real-world applications for AI-based approaches.

The German publication 'Artificial Intelligence (AI) in Security Aspects of Industrie 4.0' by Plattform Industrie 4.0 [29] addresses some of the AI-based or AI-supported use cases.

### 10.2.1 AI-supported security controls

As AI-supported security concepts, the following are addressed in §2 of [29]:

- identification and authentication with AI support,
- AI-supported anomaly detection in data stream,
- AI-supported malware detection.

The reliable and consistent strength of AI-based security controls must be demonstrated by security assessments. This may be very challenging, e.g. on how to assess whether slightly manipulated 2D or 3D face scans are reliably rejected by the ML-based algorithms that serve as part of a security control.

### 10.2.2 AI-supported security attacks

§3 of [29] addresses the deployment of AI for

- attacks targeted to office IT,
- attacks of Operational Technology (OT), and even
- attacks targeting the AI systems themselves.

With I4.0/IM, IT and OT worlds are getting more and more intertwined. Accordingly, all of the above scenarios are within the scope of security testing.

The book *Artificial Intelligence for Cybersecurity* [30] introduces the use of AI techniques to realise cybersecurity-related goals. This includes using algorithms to automate the preliminary screening of threats and submitting suspicious objects to professionals so they can initiate a timely response. This AI-assisted automation approach can improve the efficiency of defence activities and reduce the workload on professionals.

The book first introduces practical knowledge required, such as setting up the necessary environment (Anaconda for using Jupyter for Python) and installing required libraries, e.g. pandas for data processing, Natural Language Toolkit (NLTK) for natural language processing (NLP).

The book then introduces tools and approaches for detecting various threats (email, malware, network threats) using AI-relevant techniques – SpamAssasin for spam detection, for example. Spam can also be detected using perceptrons (Neural Network, NN), support-vector machines (SVMs, a supervised learning algorithm that can deal with spam represented in an image), Naive Bayes (including an example using NLTK), or using logistic regression models and decision trees for phishing detection.

For malware detection, the book discusses statistic and dynamic malware analysis. Hidden Markov Models (HMMs) can be used to detect metamorphic malware. Convolutional neural networks (CNNs, which have some image recognition advantage) can also be used for malware detection. When it comes to network anomaly detection, Gaussian distribution is suggested as a means to detect data regularity.

The book also discusses the protection of sensitive information, e.g. by using automated learning algorithms to monitor user account activities for securing the user authentication procedure. The IBM Watson cloud solution is presented as an example of preventing fraud detection. The book also introduces Generative Adversarial Networks (GANs) with relevant Python tools and libraries for carrying out attacks (e.g. attacks on biometric authentication procedures) and defence neural networks from GANs-based attacks. Put briefly, the GANs idea functions by putting two neural networks together so each competes with the other until ultimately a balanced situation is reached. It can be used on IDS.

Finally, [30] discusses the topic of evaluating the aforementioned algorithms and assessing the techniques hackers use to evade AI-securing algorithms/approaches/tools. These include evaluating a detector's performance and evading ML detectors.

### 10.2.3 Generative Adversarial Networks (GAN)

Invented in 2014, GAN is an AI technology from the Unsupervised Learning domain. GAN will be addressed here as an example of technology that can be used to test the strength of authentication mechanisms based on 2D or 3D face scans of persons, for example.

As a generative ML algorithm, GAN generates new examples out of a base structure without an explicit approximation of the pertaining probabilistic distributions. Although still a topic of research, GAN has the potential to learn the behaviour of an Intrusion Detection System (IDS) or Intrusion Prevention System (IPS) and improve its behaviour to the point where the IDS/IPS cannot detect embedded invalid input. The basic idea is to let two neural networks (a Generative Network and a Discriminative Network) perform a contest until they reach a stable equilibrium (Nash equilibrium). The Discriminative Network learns by Supervised Learning to discern real elements of a given set from new elements. During the contest, the Generative Network receives information about the actions of the Discriminative Network, based on which the Generative Network maintains and continuously improves its ability to distinguish between real and synthetic elements. The Generative Network makes use of this knowledge to improve the parameters of its own function for element generation.

There are already many startling examples where GAN learns to generate new pictures out of pictures from handwritten numbers, clothes, furniture, artistic drawings, tone sequences and other, which a human observer cannot discern and considers as genuine. This ability can be used to test current (e.g. face recognition-based) authentication mechanisms.

In [31] an IDSGAN is described, which is trained by a Discriminative Network used instead of the IDS. This cannot be used directly, since its behaviour is assumed to be black box (unknown internal implementation of the IDS).

Evidently, beyond IDS, GAN attacks can be targeted at other security controls. The challenge for security testing consists in demonstrating that applied security controls or security defence-in-depth measures are not vulnerable to such attacks.

# 11. Competence requirements for security testers

The following sections provide an overview of competence requirements for security testers according to current international and national standards, along with discussions on the security testing skills needed in the I4.0/IM context.

Before addressing the specific competence requirements for security testing, Section 11.1 will address the baseline security management requirements.

## 11.1 Baseline security competence requirements

This section will briefly address the baseline security competences that must be acquired by security staff before specialising in technical subdomains of security testing.

The competences required for information security governance and planning include an understanding of information security frameworks, regulations and standards, as well as an ability to identify and implement them in support of business guidelines. In addition to this, staff must be able to understand and identify the context, objectives and benefits of the organisation in terms of information security management. As part of information security planning, work on risk assessment and treatment, professionals must have required knowledge of the topic, be able to determine risks and put in place the processes capable of treating such risks.

Other sets of qualifications relating to the technical domain and deployed technologies are necessary for information security operation and support, for which personnel are expected to perform either safety or security-related processes effectively. In addition, and with the goal of disseminating a security culture among staff concerned, information security awareness will be maintained by means of regular training courses. Further aspects of importance for information security skills are monitoring, measurement (metrics), analysis, evaluation and auditing. Professionals must have the competences to perform these tasks. They should also be familiar with the methodologies and frameworks for both internal and external audits.

Finally, since it is crucial to the organisation to maintain a continual improvement process and keep pace with technological improvements, personnel working on the topic must acquire skills such as balancing the benefits of corrective actions against cost, analysing the business impact of emerging technologies, etc.

## 11.2 Security testing skills-related requirements of ISO/IEC 27021

ISO/IEC 27021:2017 provides a high-level list of the knowledge and skills professionals must acquire in relation to the topic of information security according to their specific tasks. These include the following topics relating to security testing:

- security requirements analysis, security specification,
- security measures analysis, vulnerability analysis,
- secure system design evaluation,
- review methods,
- risk communication and consultation, risk mitigation,
- preventive maintenance and patch management,
- penetration testing,
- physical security provisions evaluation,
- security data analysis,
- security evaluation testing,
- secure coding principles, secure programming techniques,
- information security assessment, testing and sampling techniques, as ISMS auditing-related competences,
- writing, leading and implementing

information security testing plans and processes and audit reports,
- system acceptance testing of information system architectures during the System Development Lifecycle (SDL),
- system development project management, system engineering and system security testing as part of the SDLC.

## 11.3 Understanding of security threat models

The following sections provide an overview of the threat model of IEC 62443-4-1 (§11.3.1), Advanced Persistent Threats (§11.3.2) and the Security Development Lifecycle (SDL) threat model (§11.3.3).

### 11.3.1 Threat model of IEC 62443-4-1

In IEC 62443-4-1, threat models should be specified, verified by the development person, reviewed periodically, updated in line with new situations occurring (either in terms of products or outside world changes). Thirteen characteristics are proposed for consideration in building threat models in the development scope. These characteristics are:

- correct flow of categorised information throughout the system,
- trust boundaries,
- processes,
- data stores,
- interacting external entities,
- internal and external communication
- protocols implemented in the product,
- externally accessible physical ports, including debug ports,
- circuit board connections such as Joint Test Action Group (JTAG) connections or debug headers, which might be used to attack hardware,
- potential attack vectors including attacks on hardware, if applicable,
- potential threats and their severity as defined by a vulnerability scoring system (for example, CVSS),
- mitigations and/or dispositions for each threat,
- security-related issues identified, and external dependencies in the form of drivers or third-party applications (code that is not developed by the supplier) that are linked into the application.

### 11.3.2 Advanced Persistent

#### Threats (APT)

An APT is an adversary (threat source) who possesses sophisticated levels of expertise (threat agents) and is backed up by significant financial resources, which allow it to create opportunities to achieve its objectives using multiple attack vectors (e.g. cyber, physical and deception) as part of an Advanced Persistent Threat (APT). The objectives of an APT typically include establishing and extending footholds within the information technology infrastructure of targeted organisations for purposes of exfiltrating information, undermining or impeding critical aspects of a mission, program or organisation, or positioning itself to carry out these objectives in the future.

#### The Advanced Persistent Threat:

**i.** pursues its objectives repeatedly over an extended period of time;

**ii.** adapts to defenders' efforts to resist it;

**iii.** is determined to maintain the level of interaction needed to execute its objectives.

### 11.3.3 Security Development

Microsoft Security Development Lifecycle (SDL) is a security assurance process for developing software [32]. It highlights the security and privacy at each stage of the software development process. Education, continuous process improvement and accountability are the key concepts of SDL. An SDL optimisation model is proposed in order to control issues caused by the introduction of secure development concepts. It includes five aspects during the software development lifecycle [5]:

- training, policy and organisational capabilities
- requirements and design
- implementation
- verification
- release and response.

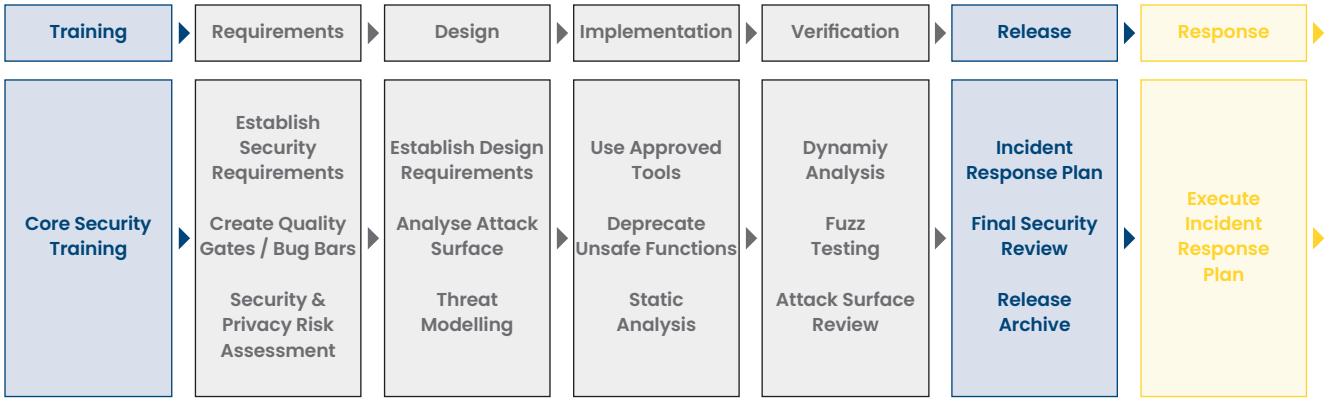


**Figure 6** — SDLC Threat Modelling

Microsoft SDL is a collection of mandatory security activities, presented in the order they should occur and grouped by the phases of the traditional software development lifecycle (SDL) [32]. Figure 6 shows the secure software development process model.

Security threat modelling is an important step during development and is already addressed during the design phase as indicated in Figure 6 above. In threat modelling, the SDLC considers the following 5 key steps [5], which are applied recurrently as indicated in the figure below:

- defining security requirements
- creating an application diagram
- identifying threats
- mitigating threats
- validating that identified threats have been mitigated.
- A Threat Modelling Tool (TMT) can be used to support work on recurrent security threat modelling [32].


**Figure 7** — SDLC Threat Modelling

## 11.4 Requirements for testers of security management and ISMS implementations

The 2013 versions of ISO/IEC 27001 and ISO/IEC 27002 address security testing in several sections. ISO/IEC 27002:2013 §14.2.8 'System security testing' provides guidance on how testing of security functionality should be carried out during development.

ISO/IEC 27002:2013 §14.2.8 'System acceptance testing' addresses acceptance testing programs and related criteria that must be established for new information systems, upgrades and new versions.

ISO/IEC 27002:2013 §12.1.4 'Separation of development, testing and operational environments' provides a list of rules that should be considered in order to minimise risks for the operational environment. This includes the handling of sensitive data (e.g. regarding newly found software vulnerabilities). The handling of test data is explicitly and more comprehensively addressed by ISO/IEC 27002:2013 §14.3.1 'Protection of test data'.

## 11.5 Requirements for ISO/IEC 19790 security testers

ISO/IEC 19790:2012 provides security requirements for cryptographic modules. These modules support cryptographic mechanisms, such as the protection of data against unauthorised disclosure or manipulation, entity authentication and non-repudiation.

The standard provides four increasing, qualitative levels of security requirements. Security Level 1 provides a baseline level of security, which does not include any specific physical security mechanisms. Level 2 adds requirements for tamper evidence. Level 3 requires identity-based authentication mechanisms. Level 4 introduces multi-factor authentication and enforces immediate zeroisation for any detected penetration.

ISO/IEC 19790:2012 §7.11.6 'Vendor testing' specifies security testing and test documentation criteria for Security Levels 1 and 2 and more stringent criteria for Levels 3 and 4. For Levels 1 and 2, functional testing performed on the cryptographic module shall be documented. Automated security diagnostic tools (e.g. detect buffer overflow) must be used. For Levels 3 and 4 the procedures for and results of low-level testing performed on a cryptographic module must be documented.

Key skills for ISO/IEC 19790-related security testers include the ability to demonstrate completeness of implementation of a cryptographic module as compared with the detailed reference design provided by comprehensive multipart standards. The detailed reference design is typically provided in a procedural programming language or a precise pseudo language. In particular, testing must address the handling of error conditions, which are often not explicitly specified in full detail in the reference implementation.

## 11.6 Requirements for ISO/IEC 15408 security evaluators

Workshops for acquiring the skills to perform Common Criteria (CC)-specific testing are provided e.g. by the German Federal Office for Information Security. For example, the workshop described in [33] provides information on the international importance of the Common Criteria, creating Protection Profiles (PP) in line with CC, an introduction to the practical application of CC for the evaluation of targets by accredited laboratories.

The general approach for this training is 'learning by doing'. As a precondition for attending this training, future evaluators must already be familiar with part one of the CC. Upon completion of the training, future evaluators are expected to have a thorough understanding of the meaning of the Evaluation Assurance Levels (EAL) and the qualitative statements of a certification report.

## 11.7 Requirements for OPST security testers

The OSSTMM Professional Security Tester (OPST) [2] certification proves that a candidate has the skills and knowledge to perform accurate and efficient security tests on data networks, in line with OSSTMM v3 o OSSTMM v4. It covers network auditing, ethical hacking, web application testing, intranet application testing and penetration testing. It targets security auditors, network engineers, system and network administrators, developers, network architects and security analysts.

As a prerequisite staff should have a sound knowledge of how networking protocols work, a good understanding of how various security devices and programs work, user-level skills with different operating systems and basic experience with server operations/administration, particularly in setting up and running LINUX daemons and services.

The classes take up to 60 hours over 30 days. The OSSTMM methodology is taught through security testing exercises with an internet-based test network. Courses are designed as an all-practice support for the theory provided, in order to carry

out security testing properly, factually and scientifically through coaching, examples and skill tests.

As part of the 'verification' scope an applicant tester must demonstrate:

- the ability to apply scientific methodology to the process of identifying and verifying vulnerability and weakness, in order to accurately determine security limitations;
- the ability to map known exploits to services;
- the ability to discover exploits of known vulnerabilities for verification;
- the ability to classify new security limitations appropriately.

## 11.8 TeleTrusT Professional

TeleTrusT is a German non-profit organisation (part of Bundesverband IT-Sicherheit e.V.), which considers itself as a 'pioneer in IT security' operating since 1989.

### 11.8.1 TeleTrusT Information Security Professional

The aim of TISP (TeleTrustT Information Security Professional) is to provide evidence of an achieved level of IT security skills, independent of the accredited organisation that issues the respective certificate. Security staff can apply for the respective examination after at least 3 years' practical experience in the IT security domain, followed by a one-week preparation.

The training programme covers hacking methods, application security, security of mobile networks, encryption technologies, Public Key Infrastructures (PKI), authentication, operating system security, security of mobile devices and others.

### 11.8.2 TeleTrusT Professional for Secure Software Engineering

The aim of TPSSE (TeleTrustT Professional for Secure Software Engineering) is to provide evidence of an achieved level of IT security skills for the integration of IT security topics during Software Development Lifecycle Phases. As part of secure software development, typical software development errors are identified and eliminated before they become potential vulnerabilities. This requires a comprehensive knowledge of tools that can be used in the domain of secure software development.

## 11.9 Requirements for testers of communication protocols security

### 11.9.1 Requirements for testers of industrial communication protocols

Only certain industrial network communication protocols include 'security by design' [21]. An outstanding Industrie 4.0 example is ISO/IEC 62541-2, which describe the OPC Unified Architecture (OPC UA) security model. A security check performed by the German Federal Office for Information Security (German BSI) found that design of the protocol is secure. In the respective 'Security evaluation of OPC UA' project, the German BSI also considered a corresponding reference implementation and concluded:

'Extensive analysis of the security functions in the specification of OPC UA confirmed that OPC UA was designed with a focus on security and does not contain latent security vulnerabilities'.

Thus, the focus of OPC UA security tests should be on protocol implementation. Accordingly, the staff that prepare and perform OPC UA tests should have a broad and in-depth understanding of the implementation of network security protocols. In the specific case of OPC UA, security testers may leverage existing comprehensive test suites available to members of the OPC Foundation. Security testers should especially address the Application Programming Interfaces that are provided by implementers. While the OPC Foundation provides reference implementations for a selection of programming languages, further implementations are available, including libraries that provide their own API on top of the OPC UA protocol for both the OPC UA server and OPC UA client side. Testers should also have sufficient background to test the OPC UA implementation without security enabled, with integrity checks enabled and with full cryptographic functionality enabled.

Security testers of Time Sensitive Networking (TSN) protocol implementations should have a sufficiently deep understanding of real-time network communication and Quality of Service (QoS) provisions.

Security testers of MQTT should have an understanding of network communication, IoT and IIoT and Internet security. While MQTT-based solutions are currently deployed in less sophisticated industrial projects, they are connected to platforms via the Internet.

With all network security protocols, security testers require not just an understanding of the network communication but also of the application layer communication, e.g. whether database transactions are executed or signal value streams are continuously transferred.

Where applicable, security testers of communication protocols should have an in-depth understanding of functional safety aspects. As a simplified rule, testers should understand that meeting functional safety requirements should not be hindered by security controls. The tester should be able to evaluate whether the quality of the implementation of the security specific code is at a similarly high level as for the functional safety part.

Security testers of communication protocols should have an understanding of the potential differences between implementations by alternative networking device vendors. In error cases, in particular, two or more communicating devices may react differently in error situations. Each one may perform correctly and meet the required Quality of Service (QoS), but the exchange/interoperability may be incorrect or not synchronised (leaving the overall system potentially in a vulnerable or unstable state).

### 11.9.2 Requirements for testers with a focus on compliance

The following requirements address testers that put their focus on compliance testing and not just performance.

Standards that specify security protocols or security extensions for existing communication protocols should also describe procedures for conformance tests. Conformance test standards usually define lists of so-called Protocol Implementation Conformity Statements (PICS). PICS are statements about features that have been implemented in an implementation. Suitable test suites, test cases and test procedures must be defined in order to check PICS. PICS can be declared optional, mandatory or conditional. Conformity tests are often performed by accredited test laboratories. The test results then give information about which features of an implementation are compliant with the standard. Matching PICS of independent implementations are an essential requirement for interoperability. However, conformance to a standard does not automatically ensure interoperability between independent implementations. Reasons for interoperability of two standard-conformant implementations may be gaps and ambiguities in standards, which allow for different interpretations resulting in different implementations. Another reason could be errors in implementations that are not recognised in tests. For example, a software may show undefined behaviour in scenarios that were not covered by conformance tests.

A way to improve standards are 'tissue' (technical issues) databases, to which implementers, testers, etc. can submit issues that occurred during implementation and interoperability tests with implementations of other vendors. Standards that describe communication security protocols must specify the behaviour of the system as accurately as possible, and must also cover borderline cases in interaction between devices that may lead to undefined (and probably insecure) behaviour.

# 12 Requirements for security test labs

ISO/IEC 24759:2017 provides methods that can be used by testing laboratories to test whether a cryptographic module conforms to the requirements of ISO/IEC 19790:2012. The methods should provide a high degree of objectivity during the testing process and should ensure consistency across testing laboratories.

## 12.1 General requirements for security test labs

ISO/IEC 17020 provides general requirements to be met by test and inspection bodies. It includes only little information on test equipment. As an example, in section 6.2, it requires that measurement devices should be calibrated before putting them into practice. Relevant equipment should be checked regularly. Any computers or automated equipment used for testing must meet protection of data security and integrity requirements. Equipment must be maintained for correct functioning. Installed software must be validated before use. Connections with related hardware should be periodically revalidated, including after any change. Updates must be implemented according to test and inspection requirements.

ISO/IEC 17025 specifies general requirements for the competence of testing and calibration laboratories from various aspects. First of all, impartiality and confidentiality are the most important policies while undertaking activities in the laboratory and accessing data relevant to the tests. Subsequently, ISO/IEC 17025 specifies structural and resource requirements of a laboratory. For example, a laboratory must be an entity with clearly identified management accountability, personnel responsibility and resource authority. Staff qualifications must be considered in terms of a laboratory's resources. Any member of laboratory staff who carries out tests should be trained to acquire sufficient knowledge, including on impartiality and confidentiality policies, documentation activities and recording of required information. In addition, consideration must also be given to equipment, facilities and environmental conditions. For example, facilities and environmental conditions should be compatible with laboratory activities and corresponding requirements must be documented. Procedures for access, control, handling, transportation etc. of laboratory equipment should be specified. Procedures for calibration, measurement, working status indication (defective or not) and equipment maintenance must be in place. Metrological traceability must also be ensured where calibration is required. ISO/IEC 17025 also specifies the process requirements, which include that the laboratory must have a procedure for reviewing requests, tenders and contracts. The laboratory must use appropriate and up-to-date validated methods for activities. Non-standard methods must be validated by the laboratory. A sampling plan is required in some cases. Technical records must be recorded in accordance with various requirements relating to the reports, evaluation of measurement uncertainty and validity results.

Overall, ISO/IEC 17020 provides a general baseline for test and inspection bodies and ISO/IEC 17025 specifies many requirements for the laboratory environment and equipment.

## 12.2 Software security evaluation-specific requirements for test labs

Test labs that perform software security evaluations should consider the test suites and state of the art on security testing in the specific domain. As an example for OPC UA, the OPC Foundation guidance should be considered. Current security evaluation and testing results, including on the security impact on performance [34] [35], certificate management [36], deployment in general [37] and for IIoT [38] should be considered.

With regard to IEC 62443 compliance certifications, some of the IECEE registered labs [39] also perform cybersecurity evaluations.

Common Criteria (IEC 15408) evaluation labs for SW security (considered profiles) have accreditation depending on Evaluation Assurance Levels (EAL). They demonstrate, for example, a product's achievement of a given EAL in accordance with agreed upon protection profiles and optionally with certain added compliance achievement criteria indicated as '+', e.g. to achieve EAL4+.

## 12.3 Hardware security evaluation-specific requirements for test labs

The range of possible equipment for HW attacks is very broad, ranging from non-invasive, to semi-invasive and invasive attacks.

Test labs should make transparent whether they support hardware security evaluations and what equipment and skills are at their disposal, such that the capabilities of different test labs can be compared with regard to decisive criteria.

Common Criteria (IEC 15408) evaluation labs for HW security or both SW and HW security are considerably less common as compared to software-only evaluation labs. This is partially due to the need for specialised hardware equipment for invasive attack simulations, e.g. by analysing security chips and iteratively removing thin layers of physical hardware until potential secret information can be identified or sufficient hardware-level protection is demonstrated.

# 13. Conclusion

This white paper contains an overview of security testing-related guidance and standards that currently exist or are in development. Security testing makes an essential contribution to ensuring an appropriate overall security posture for Industrie 4.0 and Intelligent Manufacturing. Assuming a scalable architecture (in the context of RAMI and IMSA), secure design at all levels of detail and modular state-of-the-art and peer-reviewed implementation, the challenge is to test for security vulnerabilities before they are identified and exploited by threat agents.

Security testing involves the development of appropriate testing schemes at product level, as being prepared currently by IECEE, the appropriate grading of test requirements by Security Levels (SL) and Maturity Levels (ML) in accordance with IEC 62443 or by Evaluation Assurance Levels (EAL) in accordance with ISO/IEC 15408.

Security testing also includes verification of adherence to secure design principles, as assessed e.g. by German BSI for the ISO/IEC 62541-series defined network interoperability protocol (OPC UA). At the level of software source code and Hardware Description Language (HDL)-based implementations, it involves testing in line with programming languages used to enforce adherence to secure programming constructs (source code security testing and fuzzing).

Testing for the secure implementation of cryptographic algorithms and Trusted Platform Modules (TPMs) is guided by dedicated standards or by consideration of reference implementations, e.g. of the ISO/IEC 11889 series for TPM Libraries, with new security testing challenges for virtual trust module (vTM) implementations, as intended by ISO/IEC 27070.

Security testing also mandates appropriate overall technical skills and security testing skills for the specialised staff involved, as outlined at a general level by ISO/IEC 27021 and imposed by the respective standards on cryptographic algorithms, FPGA technology, etc. This comes with the need for in-depth training on security testing-related topics for staff involved. When combined with certification, in addition to the specialised staff, additional security testing-specific requirements must be met by laboratories performing the evaluations, assessments and certifications.
The use of Machine Learning (ML) and Artificial Intelligence (AI) have been addressed as support for the generation and execution of advanced attack scenarios prior to commissioning of the respective systems and before black hat hackers can deploy ML/AI technology against the new systems.

Additional considerations are required with regard to testing whether requirements for functional safety and security (IEC TR 63069) are jointly met, e.g. testing to ensure that security controls have no negative impact on safety.

Security testing for security and privacy aspects, e.g. correct de-identification, involves the consideration of respective big data and cloud computing-related security standards, in particular ISO/IEC JTC1/SC27 WG4/WG5.

With I4.0/IM, security testing solutions must be combined, adapted and streamlined so they provide appropriate graded assurance on the secure use of advanced security defence-in-depth solutions. Since demonstration of 100%-test coverage cannot be achieved even for medium complex software systems, security testing geared to demonstrating that there are no remaining security vulnerabilities will remain a challenge for years to come.

# 14. References


[1] https://www.techopedia.com/definition/26342/black-hat-hacker

[2] www.opst.org, OPST: The OSSTMM Professional Security Tester

[3] Daniel Gray, Hyperjacking – Future Computer Server Threat, 2009-02-09, available at: http://www.syschat.com/hyperjacking-future-computer-server-threat-4917.html. Last accessed 2021-10-02.

[4] https://www.sciencedirect.com/topics/engineering/testability

[5] Microsoft Security Development Lifecycle (SDL) – Threat Modelling, available at: https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling. Accessed on 2021-10-02.

[6] MISRA Compliance:2020, Achieving compliance with MISRA Coding Guidelines https://www.misra.org.uk/LinkClick.aspx?fileticket=vfArSqzP1d0%3d&tabid=57

[7] Linux in 2020: 27.8 million lines of code in the kernel, 1.3 million in system, https://www.theregister.com/2020/01/06/linux_2020_kernel_systemd_code/

[8] F. Kesel, R. Bartholomä, 'Entwurf von digitalen Schaltungen und Systemen mit HDLs und FPGAs', 2. Auflage, Oldenburg Verlag München

[9] P.J. Ashenden, 'The Designeer's Guide to VHDL', Third Edition, ELSEVIER

[10] https://trustedcomputinggroup.org/wp-content/uploads/TPM-2.0-A-Brief-Introduction.pdf, some guidance on testing of TPM 2.0

[11] https://thevirtualizationguy.wordpress.com/2014/08/18/vasto-virtualization-assessment-toolkit/ as an example tool for testing the security of hypervisors

[12] S. Donaldson, N. Coull and D. McLuskie, 'A Methodology for Testing Virtualisation Security', International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA 2017), June 19-20, 2017, London

[13] IECEE OPERATIONAL DOCUMENT (OD), IEC System of Conformity Assessment Schemes for Electro-technical Equipment and Components (IECEE System) – Industrial Cyber Security Program, Edition 2.0, 2020-06 https://www.iecee.org/documents/refdocs/downloads/od-2061_ed.2.0.pdf

[14] TUEV SUED: IECEE CB Scheme, https://www.tuvsud.com/en/services/product-certification/iecee-cb-scheme

[15] IT Security Association Germany (TeleTrusT), TeleTrusT Evaluation Method for IEC 62443-4-2, 2019-05, Security for Industrial Automation and Control Systems IEC 62443 (IEC 62443-4-2:2019), https://www.teletrust.de/fileadmin/docs/fachgruppen/TeleTrusT-Evaluation_Method_IEC62443-4-2_2019-05_ENG.pdf

[16] ISA Security Compliance Institute, 'CSA-311 Component Security Assurance – Functional security assessment for components, Version 1.11,' 2019

[17] ISASecure, 'ISASecure - About us', 2021, available at: https://www.isasecure.org/en-US/About-Us. Accessed on 2021-10-02.

[18] Ariffuddin Aizuddin: The Common Criteria ISO/IEC 15408 – The Insight, Some Thoughts, Questions and Issues, https://www.sans.org/reading-room/whitepapers/standards/common-criteria-iso-iec-15408-insight-thoughts-questions-issues-545

[19] Bundesamt für Sicherheit in der Informationstechnik (BSI): Guidelines for Developer Documentation according to Common Criteria Version 3.1, https://www.commoncriteriaportal.org/files/ccfiles/CommonCriteriaDevelopersGuide_1_0.pdf

[20] The Open Source Security Testing Methodology Manual – OSSTMM 3, https://www.isecom.org/OSSTMM.3.pdf

[21] DIN/DKE German Standardisation Roadmap Industrie 4.0 – Version 4, 2020

[22] S. Sorsa (2018): Protocol Fuzz Testing as a part of Secure Software Development Life Cycle Tampere University of Technology Master of Science thesis, Feb. 2018, Master's Degree Programme in Information Technology Major: Pervasive Systems, https://trepo.tuni.fi/bitstream/handle/123456789/25667/Sorsa.pdf?sequence=4&isAllowed=y

[23] Suchorab, J., Staszkiewicz, K., Walkiewicz, J. and Dudek, M. (2020): Effective Fuzz Testing for Programmable Logic Controllers Vulnerability Research to Ensure Nuclear Safety, International Conference on Nuclear Security 2020 https://conferences.iaea.org/event/181/contributions/15925/attachments/8547/11382/IAEA-CN-278-619_SUCHORAB_et_al.pdf

[24] Kobara, K. (2016): Cyber Physical Security for Industrial Control Systems and IoT, IEICE TRANS. INF. & SYST., (INVITED PAPER Special Section on Information and Communication System Security), VOL.E99–D, NO.4 APRIL 2016, pp 787-795 https://search.ieice.org/bin/pdf_link.php?category=D&lang=E&year=2016&fname=e99-d_4_787&abst

[25] W. Zhao et al., 'Security Testing Methods and Techniques of Industrial Control Devices,' 2013 Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2013, pp. 433-436, doi: 10.1109/IIH-MSP.2013.114

[26] J. Liang, M. Wang, Y. Chen, Y. Jiang and R. Zhang, 'Fuzz testing in practice: Obstacles and solutions,' 2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER), 2018, pp. 562-566, doi: 10.1109/SANER.2018.8330260

[27] George Klees, Andrew Ruef, Benji Cooper, Shiyi Wei, and Michael Hicks. 2018. Evaluating Fuzz Testing. Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18). Association for Computing Machinery, New York, USA, 2123–2138. DOI: https://doi.org/10.1145/3243734.3243804

[28] https://skylightcyber.com/2019/07/18/cylance-i-kill-you/

[29] BMWK, Plattform Industrie 4.0, 'Artificial Intelligence (AI) in Security Aspects of Industrie 4.0', German language publication: Künstliche Intelligenz (KI) in Sicherheitsaspekten der Industrie 4.0, 2019-02, https://www.plattform-i40.de/PI40/Redaktion/DE/Downloads/Publikation/KI-in-sicherheitsaspekten.html
[30] A. Parisi, Hands-On Artificial Intelligence for Cybersecurity: Implement Smart AI Systems for Preventing Cyber Attacks and Detecting Threats and Network Anomalies, Packt Publishing, 342 pages, 2019-08
[31] Zilong Lin, Yong Shi, Zhi Xue (2018), IDSGAN: Generative Adversarial Networks for Attack Generation against Intrusion Detection, arXiv:1809.02077 cs.CR
[32] Microsoft Security Development Lifecycle (SDL) – Process Guidance, available at: https://www.microsoft.com/en-us/securityengineering/sdl. Accessed on 2021-10-02.
[33] https://www.bsi.bund.de/DE/Themen/ZertifizierungundAnerkennung/Personenzertifizierung/CC-Schulung/CC-workshop/ccworkshop.html, 'Anforderungen zur Kompetenzfeststellung als Evaluator im Bereich Common Criteria' (Requirements to be met by CC evaluators)
[34] S. Cavalieri, G. Cutuli and S. Monteleone, 'Evaluating impact of security on OPC UA performance,' 3rd International Conference on Human System Interaction, 2010, pp. 687-694, doi: 10.1109/HSI.2010.5514495
[35] S. Cavalieri, G. Cutuli, 'Performance evaluation of OPC UA,' 2010 IEEE 15th Conference on Emerging Technologies & Factory Automation (ETFA 2010), 2010, pp. 1-8, doi: 10.1109/ETFA.2010.5641184
[36] A. Fernbach and W. Kastner, 'Certificate management in OPC UA applications: An evaluation of different trust models,' Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012), 2012, pp. 1-6, doi: 10.1109/ETFA.2012.6489675
[37] L. Roepert, M. Dahlmanns, I. Berenice Fink, Jan Pennekamp, M. Henze: Assessing the Security of OPC UA Deployments (https://publikationen.uni-tuebingen.de/xmlui/bitstream/handle/10900/100433/assesing_the_security_of_opc_ua.pdf?sequence=1&isAllowed=y)
[38] F. Kohnhäuser, D. Meier, F. Patzer and S. Finster, 'On the Security of IIoT Deployments: An Investigation of Secure Provisioning Solutions for OPC UA,' in IEEE Access, vol. 9, pp. 99299-99311, 2021, doi:10.1109/ACCESS.2021.3096062
[39] List of IECEE Labs, https://www.iecee.org/dyn/www/f?p=106:42:0, Accessed on 2022-03-29
[40] http://www.iso27001security.com/index.html
[41] NIST SP800-82, Rev2, Guide to Industrial Control System (ICS) Security
[42] NIST SP800-161, Supply Chain Risk Management Practices for Federal Information Systems and Organisations
[43] NIST Cyber-security Framework
[44] ISO/IEC 27033 Information technology – Security techniques – Network security Part 1 ~ Part 6
[45] ISO 55000:2014 Asset management – Overview, principles and terminology
[46] ISO/IEC 19770-1:2012 Information technology – IT asset management (ITAM) – Part1 ~ Part8
[47] http://www.opcfoundation-events.com/uploads/media/OPC-UA-Wegbereiter-der-IE40-DE-v2.pdf
[48] https://www.teletrust.de/tisp/, T.I.S.P., TeleTrusT Information Security Professional
[49] https://www.teletrust.de/tpsse/, T.P.S.S.E., TeleTrusT Professional for Secure Software Engineering
[50] https://www.owasp.org/index.php/Fuzzing, OWASP definition of fuzzing
[51] eXtensible Access Control Markup Language (XACML) Version 3.0, OASIS Standard, http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html
[52] NIST SP 800-39: Managing Information Security Risk, available at: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-39.pdf, 2011-03. Accessed on 2021-10-02.
[53] ISO/SAE 21434:2021, Road vehicles – Cybersecurity engineering
[54] Sino-German White Paper on Functional Safety for Industrie 4.0 and Intelligent Manufacturing, 2020-08-03, https://www.bmwi.de/Redaktion/DE/Publikationen/Industrie/industrie-4-0-sino-german-white-paper-on-functional-safety-for-industry-4-0-and-intelligent-manufacturing.html, Accesssed on 2021-12-09
[55] Security Check Performed by German Federal Office for Information Security, https://opcconnect.opcfoundation.org/2016/06/bsi-security-check/, Accessed on 2022-03-29

# 15. Annex

## 15.1 ISO/IEC information security and testing-related standards

The following table lists ISO/IEC 27000-series related information security standards (the 'ISO27k standards') that are either published or being developed together with ISO/IEC standards relating to software testing and security evaluation.

| Standard | Published | Title | Notes |
|---|---|---|---|
| ISO/IEC 11889 | -1 2015 | Information technology — Trusted Platform Module Library — Part 1: Architecture | Defines the architectural elements of a Trusted Platform Module (TPM). |
| | -2 2015 | Information technology — Trusted Platform Module Library — Part 2: Structures | Defines the constants, flags, and structures used to communicate with a TPM. |
| | -3 2015 | Information technology — Trusted Platform Module Library — Part 2: Commands | Detailed description of commands. Code written in C language with extensive comments. Behaviour of the C code is normative. |
| | -4 2015 | Information technology — Trusted Platform Module Library — Part 2: Supporting routines | Supporting framework for the code that performs command actions. |
| ISO/IEC 17788 | 2014 | Information technology — Cloud computing — Overview and vocabulary | Cloud computing-specific vocabulary. |
| ISO/IEC 18031 | 2011 | Information technology — Security techniques — Random bit generation | Specifies the characteristics of the main elements required for a non-deterministic random bit generator. |
| ISO/IEC 18045 | 2011 | Information technology — Security techniques —Methodology for IT security evaluation | Companion document to the evaluation criteria for IT security defined in ISO/IEC 15408. It defines the minimum actions to be performed by an evaluator. |
| ISO/IEC 19790 | 2012 | Information technology — Security techniques — Security requirements for cryptographic modules | Defines four security levels for cryptographic modules to provide for a wide spectrum of data sensitivity. ISO/IEC 19896-2 provides knowledge, skills and effectiveness requirements for security testers of ISO/IEC 19790 compliance. |

| Standard | Published | Title | Notes |
|---|---|---|---|
| ISO/IEC 19896 | -1 2018 | IT security techniques — Competence requirements for information security testers and evaluators — Part 1: Introduction, concepts and general requirements | Defines terms and establishes an organised set of concepts and relationships to understand the competency requirements for information security assurance conformance-testing and evaluation specialists. |
| | -2 2018 | Information technology — Security techniques — Competence requirements for information security testers and evaluators – Part 2: Knowledge, skills and effectiveness requirements for ISO/IEC 19790 testers | Specialised requirements to demonstrate knowledge, skills and effectiveness requirements of individuals in performing security testing projects in accordance with ISO/IEC 19790:2012 and ISO/IEC 24759. |
| | -3 2018 | IT security techniques — Competence requirements for information security testers and evaluators — Part 3: Knowledge, skills and effectiveness requirements for ISO/IEC 15408 evaluators | Provides the specialised requirements to demonstrate competence of individuals in performing IT product security evaluations in accordance with ISO/IEC 15408 (all parts) and ISO/IEC 18045. |
| ISO/IEC 20547 | -3 2020 | Information technology — Big data reference architecture — Part 3: Reference architecture | Specifies the big data reference architecture (BDRA). It provides a user view and a functional view. |
| ISO/IEC 21874 | 2018 | Information technology — Security techniques — Security guidelines for design and implementation of virtualised servers | Guidance on assuring appropriate protection of virtual machines (VMs), application workloads running in VMs and the virtualised infrastructure. |
| ISO/IEC 23168 | 2018 | Information technology — Cloud computing — Framework of trust for processing of multi-sourced data | Cloud computing-specific security guidance. |
| ISO/IEC TR 23188 | 2020 | Information technology — Cloud computing — Edge computing landscape | Consideration of edge devices by cloud computing. |
| ISO/IEC 24392 | 2022 Draft | Information technology — Security techniques —Security reference model for Industrial Internet Platform (IIP) | Industrial Internet Platform (IIP) for secure exchange between platform partners, sub-suppliers and real-time IIoT devices. |
| ISO/IEC 24759 | 2017 | Information technology — Security techniques — Test requirements for cryptographic modules | Specifies methods for testing whether a cryptographic module conforms to the requirements of ISO/IEC 19790 |
| ISO/IEC TR 24772 | -1 2019 | Programming languages — Guidance to avoiding vulnerabilities in programming languages — Part 1: Language-independent guidance | Language-independent guidance on avoiding vulnerabilities in programming languages, extracted as a separate standard part. |
| ISO/IEC TR 24772 | -2 2020 | Programming languages — Guidance to avoiding vulnerabilities in programming languages — Part 2: Ada | Language-specific guidance on avoiding vulnerabilities in the Ada programming language. |
| ISO/IEC TR 24772 | -3 2020 | Programming languages — Guidance to avoiding vulnerabilities in programming languages — Part 3: C | Language-specific guidance on avoiding vulnerabilities in the ANSI C programming language. |

| Standard | Published | Title | Notes |
|---|---|---|---|
| **ISO/IEC 27000** | 2016 | Information security management systems — Overview and vocabulary | Overview/introduction to the ISO27k standards as a whole, plus specialist vocabulary. |
| **ISO/IEC 27001** | 2013 | Information security management systems — Requirements | Formally specifies an ISMS against which thousands of organisations have been certified compliant. |
| **ISO/IEC 27002** | 2013 | Code of practice for information security controls | A reasonably comprehensive suite of information security control objectives and generally accepted good-practice security controls. |
| **ISO/IEC 27003** | 2010 | Information security management system — implementation guidance | Basic advice on implementing ISO27k. |
| **ISO/IEC 27004** | 2016 | Information security management — Measurement | |
| **ISO/IEC 27005** | 2018 | Information security risk management | Discusses risk management principles without specifying particular methods. |
| **ISO/IEC 27006** | 2015 | Requirements for bodies providing audit and certification of information security management systems | Formal guidance for certification bodies. |
| **ISO/IEC 27007** | 2011 | Guidelines for information security management systems auditing | Auditing the management system elements of ISMS. |
| **ISO/IEC TR 27008** | 2011 | Guidelines for auditors on information security management systems controls | Auditing the information security elements of ISMS. |
| **ISO/IEC 27009** | 2016 | Sector-specific application of ISO/IEC 27001 — requirements | Guidance for those developing new ISO27k standards. |
| **ISO/IEC 27010** | 2015 | Information security management for inter-sector and inter-organisational communications | Sharing information on information security between industry sectors and/or nations, particularly those affecting 'critical infrastructure'. |
| **ISO/IEC 27011** | 2016 | Information security management guidelines for telecommunications organisations based on ISO/IEC 27002 | Information security controls for the telecoms industry; also called 'ITU-T Recommendation x.1051'. |
| **ISO/IEC 27013** | 2015 | Guidance on the integrated implementation of ISO/IEC 27001 and ISO/IEC 20000-1 | Combining ISO27k/ISMS with IT Service Management/ITIL. |
| **ISO/IEC 27014** | 2013 | Governance of information security | Governance in the context of information security; will also be called 'ITU-T Recommendation X.1054' |
| **ISO/IEC TR 27015** | 2012 | Information security management guidelines for financial services | Applying ISO27k in the finance industry. |
| **ISO/IEC TR 27016** | 2014 | Information security management — Organisational economics | Economic theory applied to information security. |

| Standard | Published | Title | Notes |
|---|---|---|---|
| **ISO/IEC 27017** | 2015 | Code of practice for information security controls for cloud computing services based on ISO/IEC 27002 | Information security controls for cloud computing. |
| **ISO/IEC 27018** | 2019 | Code of practice for protection of personally identifiable information (PII) in public clouds acting as PII processors | Privacy controls for cloud computing. |
| **ISO/IEC TR 27019** | 2013 | Information security management guidelines based on ISO/IEC 27002 for process control systems specific to the energy industry | Information security for ICS/SCADA/ embedded systems (not just used in the energy industry). |
| **ISO/IEC 27021** | 2017 | Competence requirements for information security management professionals | Guidance on the skills and knowledge necessary to work in this field. |
| **ISO/IEC TR 27023** | 2015 | Mapping the Revised Editions of ISO/IEC 27001 and ISO/IEC 27002 | Belated advice for those updating their ISMSs from the 2005 to 2013 versions. |
| **ISO/IEC 27031** | 2011 | Guidelines for information and communications technology readiness for business continuity | Continuity (i.e. resilience, incident management and disaster recovery) for ICT, supporting general business continuity. |
| **ISO/IEC 27032** | 2012 | Guidelines for cybersecurity | Covers baseline security practices for stakeholders in Cyberspace. It addresses information security, network security, internet security by drawing out the unique aspects and by including dependencies on other security domains. |
| **ISO/IEC 27033** | -1 2015 | Network security overview and concepts | Various aspects of network security, updating and replacing ISO/IEC 18028. |
| | -2 2012 | Guidelines for the design and implementation of network security | |
| | -3 2010 | Reference networking scenarios — threats, design techniques and control issues | |
| | -4 2014 | Securing communications between networks using security gateways | |
| | -5 2013 | Securing communications across networks using Virtual Private Networks (VPNs) | |
| | -6 2016 | Securing wireless IP network access | |
| | -7 DRAFT | Network security — Guidelines for network virtualisation security | |

| Standard | Published | Title | Notes |
|---|---|---|---|
| ISO/IEC 27034 | -1 2011 | Application security — Overview and concepts | Multi-part application security standard. Promotes the concept of a re-usable library of information security control functions, formally specified, designed and tested. |
| | -2 2015 | Organisation normative framework | |
| | -3 2018 | Application security management process | |
| | -4 2021 DRAFT | Verification and validation | |
| | -5 2017 | Protocols and application security control data structure | |
| | -6 2016 | Case studies | |
| | -7 2018 | Application security assurance prediction framework | |
| ISO/IEC 27035 | 2016 | Information security incident management | Replaced ISO TR 18044; now being split into three parts. |
| ISO/IEC 27036 | -1 2014 | Information security for supplier relationships — Overview and concepts | Information security aspects of ICT outsourcing and services. |
| | -2 2014 | Common requirements | |
| | -3 2013 | Guidelines for ICT supply chain security | |
| | -4 2016 | Guidelines for security of cloud services | |
| ISO/IEC 27037 | 2012 | Guidelines for identification, collection, acquisition and preservation of digital evidence | First of several IT forensics standards. |
| ISO/IEC 27038 | 2014 | Specification for digital redaction | Redaction of digital documents. |
| ISO/IEC 27039 | 2015 | Selection, deployment and operations of intrusion detection and prevention systems (IDPS) | IDS/IPS |
| ISO/IEC 27040 | 2015 | Storage security | IT security for stored data. |
| ISO/IEC 27041 | 2015 | Guidelines on assuring suitability and adequacy of incident investigative methods | Assurance of the integrity of forensic evidence is absolutely vital. |
| ISO/IEC 27042 | 2015 | Guidelines for the analysis and interpretation of digital evidence | IT forensics analytical methods. |
| ISO/IEC 27043 | 2015 | Incident investigation principles and processes | The basic principles of eForensics. |
| ISO/IEC 27050 | -1 2016 | Electronic discovery — overview and concepts | More eForensics advice. |
| | -2 DRAFT | Guidance for governance and management of electronic discovery | Advice on treating risks relating to eForensics. |
| | -3 DRAFT | Code of practice for electronic discovery | A how-to-do-it guide. |
| ISO/IEC 27070 | 2021 DRAFT | Information technology — Security techniques — Requirements for establishing virtualised roots of trust | Requirements for establishing virtualised roots of trust, based on ISO/IEC 11889 multipart standards series. |

| Standard | Published | Title | Notes |
|---|---|---|---|
| **ISO/IEC 27400** | 2021 DRAFT | Cybersecurity — IoT security and privacy — Guidelines | Overall cybersecurity guidance for IoT devices addressed by ISO/IEC 2740x. |
| **ISO/IEC 27402** | 2021 DRAFT | Cybersecurity — IoT security and privacy — Device baseline requirements | Baseline cybersecurity requirements for IoT devices. |
| **ISO/IEC 27403** | 2021 DRAFT | Cybersecurity — IoT security and privacy — Guidelines for IoT-domotics | Cybersecurity guidelines for smart home and building automation. |
| **ISO 27799** | 2016 | Health informatics — Information security management in health using ISO/IEC 27002 | Information security advice for the healthcare industry. |
| **ISO/IEC/IEEE 29119** | -1 2013 | Software and systems engineering — Software testing — Part 1: Concepts and definitions | Introduces the concepts and vocabulary on which these test standards are built, as well as providing practical examples. |
| | -2 2013 | Software and systems engineering — Software testing — Part 2: Test processes | Defines software testing processes at the organisational level, test management level and dynamic test levels. |
| | -3 2013 | Software and systems engineering — Software testing — Part 3: Test documentation | Includes templates and examples of test documentation. |
| | -4 2015 | Software and systems engineering — Software testing — Part 4: Test techniques | Defines techniques that can be used during the test design and implementation process. |
| | -5 2016 | Software and systems engineering — Software testing — Part 5: Keyword-Driven Testing | Defines an efficient and consistent solution for testers that deploy test automation based on keywords. |
| **ISO/IEC 29190** | 2015 | Information technology — Security techniques — Privacy capability assessment model | Specifies a set of levels for privacy capability assessment. |
| **ISO/IEC 20008** | -1:2013 | Information technology — Security techniques — Anonymous digital signatures — Part 1: General | Signature mechanisms using a group public key and signature mechanisms using multiple public keys. |
| | -2:2013 | Information technology — Security techniques — Anonymous digital signatures — Part 2: Mechanisms using a group public key | General description of an anonymous digital signature mechanism using a group public key and mechanism that provide anonymous digital signatures. |

## 15.2 IEC Security and Testing-related Standards

The following table lists standards of IEC TC65 that are either published or being developed, together with further standards relating to software testing and security evaluation, especially IECEE.

| Standard | Published | Title | Notes |
|---|---|---|---|
| **IEC 62351** | -8 2020 | Power systems management and associated information exchange — Data and communications security — Part 8: Role-based access control for power system management | Role Based Access Control (RBAC) for the Smart Grid, in the context of the IEC 61850 and IEC 62351 standards series. |
| **IEC TR 62541** | -1 2015 | OPC unified architecture — Part 1: Overview and concepts | Concepts and overview of the OPC Unified Architecture (OPC UA). |
| **IEC TR 62541** | -2 2016 | OPC unified architecture — Part 2: Security Model | Overview of security features specified in other parts of the OPC UA specification, including references services, mappings and profiles. |
| **IEC 62443** | -2-4 2015 | Security for industrial automation and control systems — Part 2-4: Security program requirements for IACS service providers | Requirements for security capabilities for IACS service providers that they can offer to the asset owner during integration and maintenance activities of an automation solution. |
| | -3-2 2020 | Security for industrial automation and control systems — Part 3-2: Security risk assessment for system design | Initial and detailed security risk assessment in the context of IEC 62443-3-3. |
| | -4-1 2018 | Security for industrial automation and control systems — Part 4-1: Secure product development lifecycle requirements | Security requirements for developers of any automation and control products where security is a concern. |
| | -4-2 2019 | Security for industrial automation and control systems — Part 4-2: Technical security requirements for IACS components | Detailed technical control system component requirements associated with the 7 foundational requirements of IEC TS 62443-1-1. Requirements for control system capability security levels. |

| Standard | Published | Title | Notes |
|---|---|---|---|
| ISO/IEC 20009 | -1:2013 | Information technology — Security techniques — Anonymous entity authentication — Part 1: General | Model, requirements and constraints for anonymous entity authentication mechanisms that allow the legitimacy of an entity to be corroborated. |
| | -2:2013 | Information technology — Security techniques — Anonymous entity authentication — Part 2: Mechanisms based on signatures using a group public key | General description of an anonymous entity authentication mechanism, based on signatures using a group public key. Anonymous authentication mechanisms without an online Trusted Third Party (TTP). |
| | -4:2017 | Information technology — Security techniques — Anonymous entity authentication — Part 4: Mechanisms based on weak secrets | Anonymous entity authentication mechanisms based on weak secrets. Applicable to situations in which a server only verifies that a user belongs to a certain group without obtaining information that can be used to identify the user. |
| IEC 62859 | 2016 | Nuclear power plants — Instrumentation and control systems — Requirements for coordinating safety and cybersecurity | Requirements and recommendations for coordinating safety and cybersecurity. |
| IEC TR 63069 | 2019 | Industrial-process measurement, control and automation — Framework for functional safety and security | Provides guidance on the common application of functional safety (IEC 61508) and cybersecurity (IEC 62443) for industrial-process measurement, control and automation. |
| IEC 63096 | 2020 | Nuclear power plants — Instrumentation, control and electrical power systems — Security controls | Structured according to ISO/IEC 27002:2013 but with security grading (4 degrees) and consideration of automation platforms/products development, systems integration and plant operation as lifecycle phases. |

## 15.3 National Security Testing-related Standards

The following table lists Chinese standards relating to international security testing and evaluation standards.

| Standard | Published | Title | Notes |
|---|---|---|---|
| **BSI-Standard 200-1** | 2017 | Information Security Management Systems (ISMS) | Efficiently managing information security. |
| **BSI-Standard 200-2** | 2017 | IT-Grundschutz Methodology | Baseline security protection guideline. |
| **BSI-Standard 200-3** | 2017 | Risk Analysis based on IT-Grundschutz | Risk assessment based on the baseline security protection guideline. |
| **BSI-Standard 100-4** | 2009 | Business Continuity Management | Guidance on security aspects of business continuity management. |
| **BSI IT-Grundschutz-Kompendium** | 2020 | IT Baseline Protection Compendium, 3rd Edition | Practical security recommendations based on a set of 96 IT Baseline Protection modules. |
| **GB/T 20274** | -1 2006 | Information Security Technology — Evaluation Framework for Information Systems Security Assurance — Part 1: Introduction and General Model | Basic concept and model of information systems security assurance. |
| | -2 2008 | Information Security Technology — Evaluation Framework for Information Systems Security Assurance — Part 2: Technical Assurance | Assessment activities which the evaluator needs to complete when using the criteria defined by Part 1 for evaluation. |
| | -3 2008 | Information Security Technology — Evaluation Framework for Information Systems Security Assurance — Part 3: Management Assurance | Defines management assurance requirements. Basis for the evaluation of management assurance requirements of the TOE. |
| | -4 2008 | Information Security Technology — Evaluation Framework for Information Systems Security Assurance — Part 4: Engineering Assurance | Defines the security engineering assurance requirements for information systems. Basis for the evaluation of engineering assurance requirements of the TOE. |
| **GB/T 30270** | 2013 | Information technology — Security technology — Methodology for IT security evaluation | Supporting standard for ISO/IEC 15408:2008. Methodology for IT security evaluation limited to EAL1 to EAL4. |
| **GB/T 30271** | 2013 | Information security technology — Assessment criteria for information security service capability, GB/T 30271-2013 | Specifies the service process model and evaluation criteria for the service capabilities of information security service providers. |

| Standard | Published | Title | Notes |
|---|---|---|---|
| **GB/T 30273** | 2013 | Information security technology — Common methodology for information systems security assurance evaluation | Describes the evaluation activities which evaluators need to complete when using the criteria defined by GB/T 20274. |
| **GB/T 20275** | 2013 | Information security technology — Technical requirements and testing and evaluation approaches for network-based intrusion detection system | Specifies the technical requirements and test evaluation methods of network intrusion detection systems. |
| **GB/T 20277** | 2015 | Information security technology — Testing and evaluation approaches for network and terminal isolation products | Specifies the test and evaluation methods for network and terminal isolation products, based on the technical requirements of GB/T 20279-2015. |
| **GB/T 20279** | 2015 | Information Security Technology — Security Technical Requirements of Network and Terminal Separation Products | Specifies requirements for security functions, security assurance, environmental adaptation and performance of network and terminal separation products. |
| **GB/T 20280** | 2006 | Information security technology — Testing and evaluation approaches for network vulnerability scanners | Provides technical support and guidance for the development, production and certification of network vulnerability scanning products. |
| **GB/T 20281** | 2015 | Information security technology — Security technical requirements and testing and evaluation approaches for firewall | Specified security technical requirements and test evaluation methods for firewalls. |
| **GB/T 20945** | 2013 | Information security technology — Technical requirements, testing and evaluation approaches for information system security audit product | Specifies technical requirements and test evaluation methods for information system security audit products. |
| **GB/T 22239** | 2019 | Information Security Technology — Baseline for Classified Protection of Cybersecurity | Baseline for classified protection of cybersecurity. |
| **GB/T 28448** | 2019 | Information security technology — Evaluation requirement for classified protection of cybersecurity | Testing according to GB/T 22239-2019. |
| **GB/T 31509** | 2015 | Information security technology — Guide of implementation for information security risk assessment | Specifies the process and method for implementation of information security risk evaluation. |
| **GB/T 35274** | 2017 | Information security technology — Security capability requirements for big data services | Security capability requirements for big data services. |
| **GB/T 35295** | 2017 | Information technology — Big data — Terminology | Big data terminology. |
| **GB/T 35589** | 2017 | Information technology — Big data — Technical reference model | Big data technical reference model. |

| Standard | Published | Title | Notes |
|---|---|---|---|
| **GB/T 37953** | 2019 | Information security technology — Security requirements and evaluation approaches for industrial control network monitor | Specifies security requirements and evaluation approaches for industrial control network monitor. |
| **GB/T 37954** | 2019 | Information security technology — Technique requirements and testing and evaluation approaches for industrial control system vulnerability detection products | Specifies technique requirements and testing and evaluation approaches for industrial control system vulnerability detection products. |
| **GB/T 37962** | 2019 | Information security technology — Common criteria for industrial control system products security | Specifies common criteria for industrial control system products security. |
| **GB/T 37980** | 2019 | Information security technology — Guide for security inspection of industrial control systems | Guidance on security inspection of industrial control systems. |
| **GB/T 36323** | 2018 | Information security technology — Security management fundamental requirements for industrial control systems | Specifies security management fundamental requirements for industrial control systems. |
| **GB/T 36324** | 2018 | Information security technology — Information security classification specifications of industrial control systems | Specifies information security classification of industrial control systems. |
| **GB/T 36466** | 2018 | Information security technology — Implementation guide to risk assessment of industrial control systems | Implementation guidance on risk assessment of industrial control systems. |
| **GB/T 36470** | 2018 | Information security technology — Common security functional requirements for data acquisition and control field devices of industrial control systems | Specifies common security functional requirements for data acquisition and control field devices of industrial control systems. |
| **GB/T 34942** | 2017 | Information security technology — The assessment method for security capability of cloud computing service | Specifies the assessment method for security capability of cloud computing service. |
| **GB/T 31168** | 2014 | Information security technology — requirements of cloud computing services | Specifies security capability requirements of cloud computing services. |