Secure IT-/OT-devices according to IEC 62443-4-2: a worldwide solution

Stefan Loubichi

Kurzfassung

Aufgrund der sehr großen Zunahme von Cyberattacken auf Kritische Infrastrukturen und der Debatte darüber, inwieweit IT-/OT- Geräte eine unverhältnismäßig große Gefahr darstellen, wird weltweit nach einer Lösung gesucht, wie die Sicherheit von kritischen Komponenten nach gleichen Kriterien bewertet werden kann. Die -eher politisch denn wissenschaftlich- belegte These, dass die Geräte eines Herstellers nur deshalb unsicher sind, weil diese in einem bestimmten Land hergestellt werden, ist zu leicht als wirtschaftlicher Protektionismus zu durchschauen und wird auf lange Sicht keinen Bestand haben.

Die Normenreihe der IEC 62443 bietet hier einen Ausweg über eine Produktzertifizierung nach IEC 62443-4-2 in Verbindung mit der IEC 62443-4-1. Schließt man hieran noch einen Black-Box-Penetrationstest an, so hat man hierdurch den Nachweis der sicheren Funktionalität, ohne dass Hersteller ihren Quellcode offenlegen müssen.

Dieser Aufsatz zeigt, wie man die Herausforderungen der europäischen NIS Direktive 2.0 oder des deutschen IT-Sicherheits-gesetz 2.0 relativ einfach mit bewährten normativen Wegen lösen kann.

Das Gute an diesem Ansatz besteht im Übrigen auch darin, dass dieser Lösungsweg branchenunabhängig ist und man nicht -wie bei den Systemzertifizierungen- branchenbezogene Zusatznormen wie zum Beispiel die IEC 27019 in der (klassischen) Energiewelt benötigen würde.

Gegebenenfalls werden Lobbyisten gegen diese Lösung einwenden, dass diese zu viel Geld kosten würde und größere Hersteller von kritischen Komponenten begünstigen würde. Da ein Blackout deutlich teurer ist als die Kosten einer Produktzertifizierung nach IEC 62443-4-2 und die Europäische Union sicher einer Quersubvention der Kosten zustimmen würde, wenn hierdurch eine höhere Sicherheit erzeugt werden kann, laufen die Verhinderungsargumente gegen die IEC 62443-4-2 ins Leere.

Authors

Prof. h.c. PhDr. Dipl.-Kfm./Dipl.-Vw. Stefan Loubichi
international experienced lead auditor and consultant for information management systems (ISO 27001, § 8 BSI-Law and IT-security catalogue § 11 1a/b EnWG) and IT-OT senior security expert, more than ten years of international experience in implementing IT-/OT- security, key note speaker and author
Essen, Germany

Introduction

At the latest since the dispute about whether products from the company Huawei are classified as secure regarding the expansion of the 5G infrastructure in Europe, a broad public has become aware that even the best Intrusion Detection System or Intrusion Prevention System is useless if the device is not secure. In the new German IT Security Act 2.0 [01] (May 2021), for example, the legislator requires manufacturers of critical components to provide a warranty declaration on their devices. This warranty declaration must refer to the entire supply chain. In principle, you can first certify everything on a piece of paper. But ultimately it is a question of what the basis for this security is. Security in Industrial Automation and Control Systems (IACS) is a core issue worldwide. Therefore, this question should not be left to lawyers or politicians but to standards experts, computer scientists or engineers who prefer uniform standards worldwide. IACS represents all parts such as systems, components and processes that are necessary for the safe operation of a power plant. In addition to the components above, software components, applications and organisational parts are also included.

Many people believe that certification according to ISO/IEC 27001 [02] would already provide them with a standard for proving IT security and OT security. However, ISO/IEC 27001 [02] is a system standard and not a product-related standard. We also need a standard for energy generation or energy distribution that has an industrial background. A globally valid series of standards for the industrial environment is the IEC 62443 family of standards. Since we want to have a test basis with which we can test and confirm the safety of a critical component, the globally recognised IEC 62443-4-2 [03] is a suitable standard.

However, security already comes from development. For this reason, the development process must of course be integrated. The development process is normatively regulated in IEC 62443-4-1 [04]. Certification according to IEC 62443-4-2 [03] (based on IEC 62443-4-1 [04]), which should be carried out by an accredited certification body, would provide us with a globally recognised basis for a safe critical component.

One could still argue that it is not ensured that accredited certification bodies really deliver the same quality worldwide. For this reason, it is suggested that a penetration test would have to be carried out for the national deployment of such a critical component. Of course, the outcomes of penetration tests vary depending on the standards and methodologies used.

Standardising penetration tests worldwide will certainly not succeed. In the end, this is also not necessary, as you only test the quality of the security of the product. Many national authorities, such as the Federal Office for Information Security in Germany, do this for government units anyway, so that there would only be an expansion of testing activities.

Manufacturers of critical components invest a lot of money in the development of their products and are afraid that the source code of their products could be lost through industrial espionage during a penetration test. But there is a solution for this too: black box penetration tests, which do not have to reveal any source code.

With product certification according to IEC 62443-4-2 [3] by internationally accredited certification companies followed by a black-box penetration test by a national authority, it would be possible to achieve the security everyone wants, based solely on verifiable facts. Of course, all this costs money. But security cannot be purchased for free.

Product life cycle and the IEC 62443 family

For our understanding, a product supplier should develop products using process compliant to IEC 62443-4-1 [04]. Conformity to IEC 62443-4-2 [03] must be achieved for this product. Those products should be integrated later, usually by a system integrator, into an Automation Solution, using a process compliant to IEC 62443-2-4 [06]. Afterwards the Automation Solution is installed at a particular site and becomes part of an Industrial Automation and Control System (IACS). Of course, security measures according to IEC 62443-3-3 [07] must be considered as well as the IEC 62443-3-2 [05]. For the asset owner IEC 62443-2-1 [08] and IEC 62443-2-4 [06] are relevant.

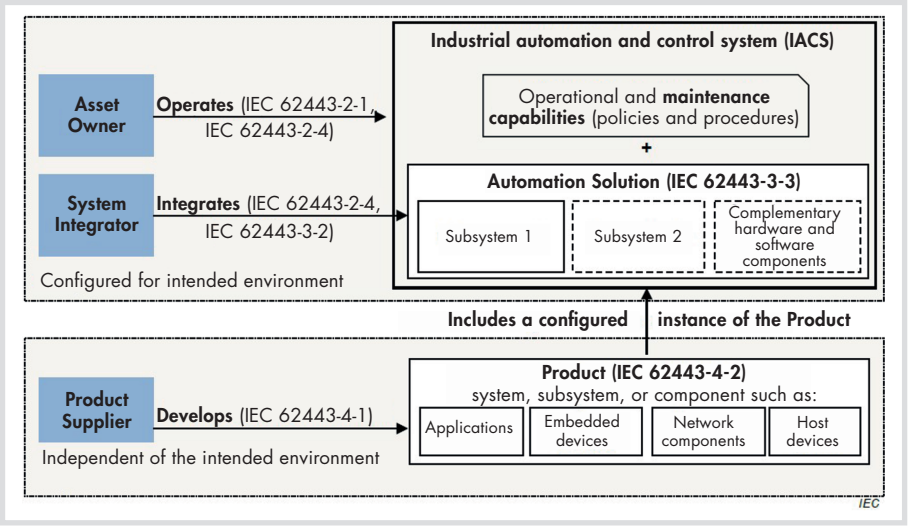

Fig. 1. Example of a product life cycle, Source: IEC 62443-4-1.

Of course, this can only be a very simplified representation of the interrelationships between the relevant IEC 62443 standards. Nevertheless, this representation and Figure 1 give an overview of the dependencies between the individual standards.

Structure of the test/audit according to IEC 62443-4.2

Industrial components according to IEC 62442-4-2 are divided into four device types:
- Embedded Devices
- Host Devices
- Network Devices
- Applications

The way chosen here for a test according to IEC 62443-4-2 is to select an SL level with associated requirements (CR) and resistance level. This allows the manufacturer's view of the Critical Component to be followed. The manufacturer can carry out an evaluation of the security properties of the various possible applications. For this purpose, the target level of the security properties must be defined via the SL level. This is preceded by an analysis of the component's operational environment.

The SL Capability (SL-C) is defined by:
- defined attacker type
- selection of requirements (CR)

IEC 62443-4-2 recognises the following types of attacks:
- SL-1: Protection against casual or coincidental violation.
- SL-2: Protection against intentional violation using simple means with low resources, generic skills, and low motivation.
- SL-3: Protection against intentional violation using sophisticated means with moderate resources, IACS specific skills and moderate motivation.
- SL-4: Protection against intentional violation using sophisticated means with extended resources, IACS specific skills and motivation.

When dealing with SL-4, one cannot avoid considering the specific safety concepts according to IEC 62443-3-2 [05]. For a better introduction to the subject, however, this essay will not refer to them.

And once again, it is pointed out here that the requirements for the secure development process for manufacturers of critical components in accordance with IEC 62443-4-1 [04] must be considered in any case. For our testing scheme, the following aspects from IEC 6244-4-1 should be used in any case in our testing scheme according to IEC 62443-4-2:
- SM-6 (File Integrity)
- SM-9 (Security requirements for externally provided components)
- SM-10 (Custom development components from third party suppliers)
- SR-1 (Product security context)
- SR-2 (Threat Model)
- SR-3 (Product security requirements)
- SR-4 (Product security requirements content)
- SR-5 (Security requirements review)
- SD-1 (Secure design principles)
- SD-2 (Defense-in-depth design)
- SD-3 (Security design review)
- SD-4 (Secure design best practices)
- SVV-1 (Security requirements testing)
- SVV-2 (Threat mitigation testing)
- SVV-3 (Vulnerability testing)
- SVV-4 (Penetration testing)
- SVV-5 (Independence of testers)
- SUM-2 (Security update communication)
- SUM-3 (Dependent component or operating system security update documentation)
- SG-1 (Product defense-in-depth)
- SG-2 (Defense-in-depth measures expected in the environment)
- SG-3 (Security hardening guidelines)
- SG-4 (Secure disposal guidelines)
- SG-5 (Secure operation guidelines)
- SG-6 (Account management guidelines)
- SG-7 (Documentation review)

Components of the test/audit according to IEC 62443-4.2

The following steps must be completed sequentially during a component test according to IEC 62443-4-2 [3]:
- Intended Use Verification
- Design Documentation
- User Documentation
- Conformity Assessment
- Vulnerability Analysis

For step 1 we must:
- define the component´s operational and security requirements (e.g., assumptions about the operational environment)
- establish and define a security concept / product security context (SR-1)
- establish and define a threat model (SR-2)

The component specification must include at least:
- Short component description
- Component identification
- Component label
- Component version
- Identification during operation, installation, and updates
- Proof of component integrity, primarily software (SM-6)
- Component category
- Excluded parts of the component.
- Component functionalities which are not considered
- Declaration of security requirements (by stating a security level: SL-x or by listing individual requirements)
- Specification of the assumed attacker type (resistance level) by stating a security level: SL-x or by describing the attacker.

In step 1 we need documented information of:
- Security concept / product security context (SR-1)
- Use cases
- Threat model (SR-2)
- Operational environment
- Product security requirements / Security functionality (SR-3 / SR-4)
- Implementation mechanism for security properties
- Information whether PKI techniques are supported or not
- For step 1 we must:

For step 2 we must:
- make a direct reference is made between the attack resistance and the absence of vulnerabilities.
- assigns the postulated design documentation to the before mentioned levels SL-x (resistance level).

The technical implementation has been adequate to the chosen security level (resistance), which is to be represented by the design documentation. This requirement results from the definitions of the seven Foundational Requirements (FR) given at the beginning of each chapter of the IEC 62443-4-2 [03]:

In this step 2 we reflect as well first "only" on the Component Requirements (CR) of the seven Foundational Requirements (FR), because these are the foundation for defining control system security capability levels:

FR-1: Identification and Authentication Control:
- CR 1.1 Human user identification and authentication
- CR 1.2 Software process and device identification and authentication
- CR 1.3 Account management
- CR 1.4 Identifier management
- CR 1.5 Authenticator management
- CR 1.6 Wireless access management
- CR 1.7 Strength of password-based authentication
- CR 1.8 Public key infrastructure certificates
- CR 1.9 Strength of public key authentication
- CR 1.10 Authenticator feedback
- CR 1.11 Unsuccessful login attempts
- CR 1.12 System use notification
- CR 1.13 Access via untrusted networks
- CR 1.14 Strength of symmetric key-based authentication

FR-2: Use Control:
- CR 2.1 Authorization enforcement
- CR 2.2 Wireless use control
- CR 2.3 Use control for portable and mobile devices
- CR 2.4 Mobile code
- CR 2.5 Session lock A
- CR 2.6 Remote session termination
- CR 2.7 Concurrent session control
- CR 2.8 Auditable events A
- CR 2.9 Audit storage capacity
- CR 2.10 Response to audit processing failures
- CR 2.11 Timestamps
- CR 2.12 Non-repudiation
- CR 2.13 Use of physical diagnostic and test interfaces

FR-3: System Integrity:
- CR 3.1 Communication integrity
- CR 3.2 Protection from malicious code
- CR 3.3 Security functionality verification
- CR 3.4 Software and information integrity
- CR 3.5 Input validation
- CR 3.6 Deterministic output
- CR 3.7 Error handling
- CR 3.8 Session integrity
- CR 3.9 Protection of audit information
- CR 3.10 Support for updates A
- CR 3.11 Physical tamper resistance and detection
- CR 3.12 Provisioning product supplier roots of trust
- CR 3.13 Provisioning asset owner roots of trust
- CR 3.14 Integrity of the boot process

FR-4: Data Confidentiality:
- CR 4.1 Information confidentiality
- CR 4.2 Information persistence
- CR 4.3 Use of cryptography

FR-5: Restricted Data Flow:
- CR 5.1 Network segmentation
- CR 5.2 Zone boundary protection
- CR 5.3 General purpose person-to-person communication restrictions

FR-6: Timely Response to Events
- CR 6.1 Audit log accessibility
- CR 6.2 Continuous monitoring

FR-7: Resource Availability:
- CR 7.1 Denial of service protection
- CR 7.2 Resource management
- CR 7.3 Control system backup
- CR 7.4 Control system recovery and reconstitution
- CR 7.5 Emergency power
- CR 7.6 Network and security configuration settings
- CR 7.7 Least functionality
- CR 7.8 Control system component inventory

For step 3 the following content of the user documentation is required:
- installing security updates for the component (SUM-2) additional independent components or underlying operating systems (SUM-3)
- rolling out security updates (SUM-4)
- describing the component's defense-in-depth strategy (SG-1)
- requirements of the defense-in-depth strategy on the operational environment
- performing security hardening via component configuration (SG-3)
- secure decommissioning/disposal (SG-4)
- secure operation (SG-5)
- account management (SG-6)

According to IEC 62443-4-1 [04], 3.1.15 defense-in-depth is an approach to defend the system against any attack using several independent methods:
- Security guidelines
- Specification of security requirements
- Security by design
- Secure implementation
- Security V&V testing

Step 4 refers to conformity assessment.

The IEC 62443-4-2 part of the standard specifies requirements (Component Requirements, CR). The requirements for the test case must be specified for each concrete component. Acceptance criteria are defined for this purpose, which are taken up as tester expectations during test case creation. In contrast to the standard, the acceptance criteria can be specified technologically. It is possible to name currently recommended technologies in concrete terms.

The procedure model for transferring the requirements looks as follows and would also have to be documented accordingly:
- requirements of the standard part
- definition of the acceptance criteria
- component-specific test cases

Let us look at how requirements are structured normatively in IEC 62443-4-2. Each component requirement consists of the following subtitles:
- Requirement
- Rationale and supplemental guidance
- Requirement enhancements
- Security levels

For example, let us examine CR 3-1 (Communication integrity):

For CR 3-1 we would have for SL-C1 the requirement: "Components shall provide the capability to protect integrity of transmitted information." For SL-C2 to SLC4 we would have additionally the requirement: "Components shall provide the capability to verify the authenticity of received information during communication".

In this case we would define the following acceptance criteria for SL-1 to SL-3:

SL-1:
- capability to protect integrity of transmitted information
- use of CRC (protection against casual or coincidental manipulation)
- use of standardized cryptographic protocol
- use of recommended protocols

SL-2:
- capability to authenticate information during communication

Not accepted would be in SL-2:
- use of error detection codes, weak hashing or weak signature functions
- authentication of information is not possible
- fallback to not recommended protocols

SL-3:

For SL-3 we would define no further requirements

In the test we would check connections for https and FTP under predefined test conditions. The following test steps would be:
- Establish connection
- Manipulate network packets
- Observe is data is still transmitted, received, and processed.

After these tests we have to our test results with the test expectations. Only if all cases are accepted, the result would be pass.

This process must be repeated for all requirements.

Step 5, vulnerability analysis, is the supreme discipline.

We begin with the vulnerability assessment methods. The following methods can be used:
- ISO/IEC 18045 [09]
- Common Methodology for Information Technology Security Evaluation [10]

Now the most used assessment model is the "Vulnerability Assessment (AVA) methodology from the Common Methodology for Information Technology Security Evaluation (CEM). In order to apply the AVA methodology to the IEC 62443 the security levels of the IEC 62443 must be adapted to the numerical values of CEM. For example:

Security Level	Sufficient Resistance Threshold
SL-1	> 0
SL-2	> 4
SL-3	> 14

The characteristics used as a basis for an attack are:
- Time needed for the design and for the execution of the attack
- Expertise
- Knowledge of the component
- Window of opportunity
- Attacker´s equipment

The following IEC 62443-4-1 practices can be used to identify vulnerabilities:
- Threat model (SR-2)
- Threat mitigation testing (SVV-2)
- Vulnerability testing (SVV-3)
- Penetration testing (SVV-4)

The goal must be achieved that all known and exploitable vulnerabilities are assessed.

According to SVV-5 of IEC 62443-4-1 [04], the auditor must have the necessary independence in the performance and evaluation of the results. The aim of the vulnerability analysis must be that at the end of the audit there are no vulnerabilities that could be successfully exploited with the attacker type defined via the SL levels of IEC 62443-4-2 [03].

In the context of the documentation of the vulnerability analysis, not only the vulnerability as such but also the entire path must be documented. For the evaluation of possible countermeasures, it is necessary to refer to the security architecture according to SD-2 of IEC 62443-4-1 [04]. For important vulnerabilities there should be an evaluation according to the Common Vulnerability Scoring System (CVSS).

CVSS is a free and open industry standard for assessing the severity of computer system security vulnerabilities. The Common Vulnerability Scoring System attempts to assign severity scores to vulnerabilities, allowing responders to prioritize responses and resources according to threat. Scores are calculated based on a formula that depends on different metrics that approximate ease of exploit and the impact of exploit. Scores range from 0 to 10, with 10 being the most severe.

Metrics are:
- Access Vector
- Access Complexity
- Authentication
- Confidentiality
- Integrity
- Availability

A Common Vulnerability Scoring System calculator can be found at:

https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator

Finally, let us look at how a conformity report of a component should look in relation to IEC 62443-4-2, so that comparability can be achieved and a government body has a basis for decision-making to define whether a component is secure or not.

Structure of a conformity report according to IEC 62443-4.2

1. Scope
1.1 Introduction
1.2 Intended Operational Environment
1.3 Basic and extended Security-Level
1.4 Abbreviations
1.5 Definitions
1.6 References
2. System Architecture
2.1 Architecture
2.2 Lifecycle Phases
3. Component Definition
3.1 Component Scope Definition
3.1.1 Short Component Description
3.1.2 Component Identification and Label
3.1.3 Component Version
3.1.4 Security Functions in terms of IEC 62443-x-x
3.1.5 Additional Security Functions
3.2 Component Type
3.3 Component Security Assumptions
3.3.1 Physical Assumptions
3.3.2 Logical Assumptions
3.3.3 Assumptions on Integrators
3.3.4 Assumptions on Supplier
3.4 Component Threats
4. Security Requirements
4.1 Use-Case Security-Level Capability
4.2 Component Requirements (CR) and Use-Case Security-Level Capability
4.2.1 Reasons for not selecting CR's
4.2.2 Modification of CRs
4.3 Additional Requirements
5. Evaluation
5.1 Required Test Environment
5.2 Required Test Interfaces
5.3 Acceptance Criteria
5.3.1 Acceptance Criteria for IEC 62443 Requirements
5.4 Acceptance Criteria for Additional Requirements
5.5 Binding Vulnerabilities
5.6 Countermeasure for Binding Vulnerabilities
5.7 CVSS classification

Evaluations would have to be carried out and documented at periodic intervals, but also when new vulnerabilities or a change in the state of the art become known.

Black box penetration test

Without a black box penetration test national authorities will (probably) not accept a conformity report according to IEC 62443-4-2.

Even if we do not want to propose a specific test, we would like to refer to the following criteria of the Cybersecurity & Infrastructure Agency [13], which are more than excellent in their quality and quantity to date:

Ease of use:
- Intuitive and easy to use for users new to automated testing tools
- Easy to install
- Tasks can be accomplished quickly, assuming basic user proficiency
- Easy to maintain automated tests, with a central repository that enables users to separate GUI object definitions from the script

Tool customisation:
- Fully customizable toolbars to reflect any commonly used tool capabilities
- Tool customisable
- Fully customized editor with formats and colours for better readability
- Tool support for required test procedure naming convention

Breadth of testing:
- Can be used with non-Microsoft platforms
- Tests for common website vulnerabilities
- Evaluates the test environment as well as the software
- Supports standard web protocols for fuzzing and domain testing

Test coverage and completeness:
- Coverage refers to the ability of the tools to test for all (known) categories of vulnerabilities relevant to the product that has been developed.

Accuracy/False-positive rate
- Is there a large number of false positives?
- Is there a large number of unidentified vulnerabilities?

Test language features:
- Allows add-ins and extensions compatible with third-party controls
- Does not involve additional cost for add-ins and extensions
- Has a test editor/debugger feature
- Test scripting language flexible yet robust; allows for modular script development
- Scripting language not too complex
- Scripting language allows for variable declaration and use and for parameter to be passed between functions
- A test script compiler or an interpreter used?
- Published APIs: Language Interface Capabilities
- Tool is not intrusive
- Allows data-driven testing
- Allows automatic data generation
- Allows adding timers for timing transaction start and end
- Allows adding comments during recording
- Allows automatic or specified synchronization between client and server
- Allows object data extraction and verification
- Allows database verification
- Allows text (alphanumeric) verification
- Allows wrappers (shells) whereby multiple procedures can be linked and called from one procedure
- Allows automatic data retrieval from any data source for data-driven testing
- Allows use of common spreadsheet for data-driven testing
- Ease of maintaining scripts when application changes

Test management:
- Supports test execution management
- Support for industry standards in testing processes
- Interoperability with tools being used to automate traditional testing
- Application requirements management support integrated with the test management tool
- Requirements management capability supports the trace of requirements to test plans to provide requirement coverage metrics
- Test plans can be imported automatically into test management repository from standard text files
- Can be customized to organization's test process
- Supports planning, managing, and analyzing testing efforts; can reference test plans, matrices, product specifications, in order to create traceability
- Supports manual testing
- Supports the migration from manual to automated scripts
- Can track the traceability of tests to test requirements
- Has built-in test requirements modules
- Can check for duplicate defects before logging newly found defects
- Allows measuring test progress
- Allows various reporting activities
- Allows tracking of manual and automated test cases
- Has interface to software architecture/modeling tool
- Is integrated with unit testing tools
- Has interface to test management tool
- Has interface to requirements management tool
- Has interface to defect tracking tool
- Has interface to configuration management tool
- Provides summary-level reporting
- Includes error filtering and review features
- Enables metric collection and metric analysis visualization

Interoperability:
- Major test automation suites provide functionality that is useful in any large-scale testing process.

Load and stress test features:
- All users can be queued to execute a specified action at the same time
- Automatic generation of summary load testing analysis reports
- Ability to change recording of different protocols in the middle of load-recording session
- Actions in a script can be iterated any specified number of times without programming or rerecording of the script
- Different connection speeds and browser types can be applied to a script without any rerecording
- Load runs and groups of users within load runs can be scheduled to execute at different times
- Automatic load scenario generation based on load testing goals: hits/second, number of concurrent users before specified performance degradation, and so on
- Cookies and session IDs automatically correlated during recording and playback for dynamically changing web environment
- Allows for variable access methods and ability to mix access methods in a single scenario: modem simulation or various line speed simulation
- Ability to have data-driven scripts that can use a stored pool of data
- Allows for throttle control for dynamic load generation
- Allows automatic service-level violation (boundary value) checks
- Allows variable recording levels (network, web, API, and so on)
- Allows transaction breakdown/drill-down capabilities for integrity verification at the per client, per session, and per instance level for virtual users
- Allows web application server integration
- Supports workload, resource, and/or performance modelling
- Can run tests on various hardware and software configurations
- Support headless virtual user testing feature
- Requires low overhead for virtual user feature (web, database, other?)
- Scales to how many virtual users?
- Simulated IP addresses for virtual users
- Thread-based virtual user simulation
- Process-based virtual user simulation
- Centralised load test controller
- Allows for reusing scripts from functional test suite
- Compatible with SSL recording
- Compatible with which network interaction technologies?
- Compatible with all relevant platforms?

Monitor test features:
- Monitors various tiers: web server, database server, and app server separately
- Supports monitoring for server frameworks?
- Supports monitoring of different platforms?
- Monitors network segments
- Supports resource monitoring
- Synchronization ability in order to determine locking, deadlock conditions, and concurrency control problems
- Ability to detect when events have completed in a reliable fashion
- Ability to provide client-to-server response times
- Ability to provide graphical results and export them to common formats

Conclusion

With the application of IEC 62443-4-2 (in conjunction with IEC 62443-4-1), we have a way of demonstrating the conformity of critical components with stringent safety requirements. A final black box pentesting then allows verification that the product certification is sufficient to confirm the security requirements.

References

[01] IT-Security Law 2.0, https://www.gesetze-im-internet.de/bsig_2009/BJNR282110009.html.

[02] ISO/IEC 27001 Information technology - Security techniques– Information security management systems– Requirements.

[03] ISO/IEC 62443-4-2 Security for Industrial Automation and Control Systems, Part 4-2: Technical requirements for IACS components.

[04] ISO/IEC 62443-4-1 Security for Industrial Automation and Control Systems, Part 4-1: Secure product development lifecycle requirements.

[05] ISO/IEC 62443-3-2 Security for Industrial Automation and Control Systems, Part 3-2: Security risk assessment for system design.

[06] ISO/IEC 62443-2-4 - Security for Industrial Automation and Control Systems Part 2-4: Security program requirements for IACS solution providers.

[07] ISO/IEC 62443-3-3 Security for Industrial Automation and Control Systems Part - Part 3-3: System security requirements and security levels.

[08] ISO/IEC 62443-2-1 Security for Industrial Automation and Control Systems Part - Part 2-1: Establishing an Industrial Automation and Control System Security Program.

[09] ISO/IEC 18045 Information security, cybersecurity and privacy protection – Evaluation criteria for IT-security – Methodology for IT-security evaluation.

[10] Common Methodology for Information Technology Security Evaluation, April 2017, Version 3.1, Revision 5.

[11] Common Criteria for Information Technology Security Evaluation, April 2017, Version 3.1 Revision 5, Part 1: Introduction and General Model.

[12] Common Criteria for Information Technology Security Evaluation, April 2017, Version 3.1 Revision 5, Part 2: Functional Security Components.

[13] Common Criteria for Information Technology Security Evaluation, April 2017, Version 3.1 Revision 5, Part 3: Assurance Security Components.

[13] https://us-cert.cisa.gov/bsi/ articles/ tools/black-box-testing/black -box-security-testing-tools.